

# On Privacy of Socially Contagious Attributes

Aria Rezaei, Jie Gao  
Stony Brook University  
{arezaei,jgao}@cs.stonybrook.edu

**Abstract**—A common approach to protect users privacy in data collection is to perform random perturbations on user’s sensitive data before collection in a way that aggregated statistics can still be inferred without endangering individual secrets. In this paper, we take a closer look at the validity of Differential Privacy guarantees, when sensitive attributes are subject to social contagion. We first show that in the absence of any knowledge about the contagion network, an adversary that tries to predict the real values from perturbed ones, cannot train a classifier that achieves an area under the ROC curve (AUC) above  $1 - (1 - \delta)/(1 + e^\epsilon)$ , if the dataset is perturbed using an  $(\epsilon, \delta)$ -differentially private mechanism. Then, we show that with the knowledge of the contagion network and model, one can do substantially better. We demonstrate that our method passes the performance limit imposed by differential privacy. Our experiments also reveal that nodes with high influence on others are at more risk of revealing their secrets than others. Our method’s superior performance is demonstrated through extensive experiments on synthetic and real-world networks.

## I. INTRODUCTION

The last decade has witnessed the exponential growth of data collection practices. While access to large-scale data has fueled the unprecedented power to solve problems previously thought impossible, it also imposes a great risk on the privacy of individuals in this new environment. A common policy is to consider individual data items to be sensitive, while knowledge of aggregated statistics on a population is not. For example, the fact that a person has a certain disease is considered sensitive, while it is safe to release the percentage of people with that disease within a population. This model has been the foundation of the popular differential privacy (DP) framework [1], in which individual entries are sensitive but queries on aggregated knowledge are answered with the guarantee that an adversary cannot use the answers to accurately infer individual data items.

In this paper we examine the interplay of personally sensitive data in a social environment. It has been widely recognized that social interactions shape the landscape of individual attributes – infectious diseases spread through social interactions and contacts; behavior changes such as obesity [2], exercising [3], or decision making processes such as voting [4] or charity donation [5] are contagious.

Due to the ubiquity of online social platforms in recent years, information about social ties and social interaction has become available. Such data can be available to the public with little effort (e.g.: professional affiliation on public web pages or friendship networks in public social networks such as Twitter), or can be mined through other means such as human mobility traces [6]. So the question we ask is: *how*

*safe are people’s sensitive attributes in a socially connected world?*

**Our Contribution.** In this paper we answer this question by proposing a novel attack to users’ sensitive attributes using information on social network connectivity, despite the fact that attributes are protected by DP mechanisms.

Suppose that the individuals participate in a survey in which they are asked about the sensitive attribute  $X$  with value 0 or 1. The goal of the survey is to learn the aggregated percentage of population who report “1”. Since the participants may not trust the data collector, they use a randomized perturbation mechanism  $\mathcal{M}$  to report data  $z_i = \mathcal{M}(x_i)$ . A simple scheme for  $\mathcal{M}$  is to flip a coin. If head, report 1 or 0 at random, otherwise report the true value. After aggregating the perturbed reports, one can approximate the true statistics by removing certain biases introduced by  $\mathcal{M}$ . For example, if there are  $p$  fraction in the population whose attributes are 1, the perturbation mechanism leads to a total fraction of  $1/4 + p/2$  reporting 1. From this, one can solve for  $p$ . Meanwhile, knowing  $z_i$  is not enough to accurately determine  $x_i$  – such protection can be formulated by differential privacy guarantee.

Now assume that the attacker knows the social connections between individuals in the survey as well as the contagion model (how this attribute spreads through the social ties). We propose an attack that exploits this information to infer the initial state  $\{x_i\}$ , with a performance bound exceeding that which is guaranteed by DP. More accurately, we show

- For *any* perturbation mechanism  $\mathcal{M}$  that guarantees  $(\epsilon, \delta)$ -differential privacy, i.e.,  $\forall z, x \neq x'$ ,

$$\Pr[\mathcal{M}(x) = z] \leq e^\epsilon \Pr[\mathcal{M}(x') = z] + \delta, \quad (1)$$

the best classifier from an attacker, *without* information of the social ties and the contagion models, has the Area Under the ROC Curve (AUC) at most  $1 - (1 - \delta)/(e^\epsilon + 1)$ .

- Next, we propose a method to infer the original sensitive values  $\{x_i\}$ , using the contagion model, the network structure, the perturbation mechanism  $\mathcal{M}$  and the noisy reported values  $\{z_i\}$ . This requires understanding how the real values correlate by accurately modeling the way they are produced by a contagion process. In prior work, contagions are ignored in modeling correlation between individual values, which results in models that are too simplistic to reflect real-world phenomena. In contrast, our model incorporates the network structure and contagion model directly into our calculations.

We proceed in two phases. First, we find the probabilistic *effectors* – a probability  $\alpha_v$  that each node  $v$  is an initiator

of a contagion that results in observed  $\{z_i\}$ . Next we run the contagion model forward from the probabilistic seeds to estimate  $\{x_i\}$ .

- Our experiments on both synthetic and real-world networks show that our method can achieve an AUC value higher than the limit imposed by DP, weakening its guarantee as a result. This also means that the social network information, while not sensitive itself, can indeed be exploited to infer sensitive knowledge. We also observe that nodes with high influence over others are more vulnerable to such attacks.

In what follows, we first present the background and related work on DP and its many variants along with prior work on social influence and contagions. We then report the theoretical upper bound on the performance of a binary classification with differential privacy protection, when there is no knowledge of the contagion network. This is followed by our attack leveraging social network structure and contagion model. We report the results of our experiments at the end.

## II. BACKGROUND AND RELATED WORK

**Contagion Models:** Many attributes are socially contagious. The way that these attributes spread in a social network is described via contagion models. A few models have received great attention in the literature. In the *Linear Threshold* model each edge has a weight that represents the influence between nodes, and nodes are activated when the sum of influence receiving from their neighbors exceeds a threshold randomly selected from  $[0, 1]$ . *Independent Cascade* model assumes that each node  $u$ , upon activation, has *one* chance to activate each of its neighbors, with different probabilities. This differs from prior work on virus contagion such as SI (susceptible-Infected) where activated node *continuously* try to activate their inactive neighbors in time-synchronous rounds. Recently there has been growing attention in *General Threshold* model [7] (first proposed by Granovetter [8]) and *Complex Contagions*, where infection requires a specific number of infected neighbors [9], [10]. In this work we mainly use the Linear Threshold model and discuss possible extensions in the last section.

**Data Privacy:** The most widely adopted privacy model is the model of differential privacy (DP) [1], which imposes constraints on publishing aggregate information about a database such that the privacy impact on individual entries is limited. Specifically, a randomized algorithm  $\mathcal{A}$  that takes a dataset as input is said to have  $(\epsilon, \delta)$ -differential privacy, if for all datasets  $D_1$  and  $D_2$  that differ on a single entry, and all subsets  $S$  of the image of  $\mathcal{A}$ :  $\Pr[\mathcal{A}(D_1) \in S] \leq e^\epsilon \Pr[\mathcal{A}(D_2) \in S] + \delta$ . The probability is taken over the randomness of the algorithm.

The original DP model does not explicitly specify the ramifications of the presence of correlation between data points, which could be accessible from outside. In fact, it is proved that when the data is assumed to be correlated, the privacy guarantees provided by DP becomes weaker [11]. This issue is acknowledged in a number of later definitions that try to address it. For example, inferential privacy [12] captures the

largest possible ratio between the posterior and prior beliefs about an individual’s data, after observing the results of a computation on a database. Here the data items may not be independent and the correlation is captured by a prior belief on the data items. In adversarial privacy [13], domain experts could plug in various data generating distributions and the goal is to protect the presence/absence of a tuple in the data set. The most general definition is PufferFish privacy [11], where one explicitly specifies the set of secrets to protect, how they shall be protected (by specifying indistinguishable pairs), how data evolve or are generated (e.g., are data items correlated), and what extra knowledge the potential attackers have. These definitions add to the complexity of DP and, as a result, we have yet to see any of them being as widely adopted as DP.

Our work could be considered as a motivation to devise mechanisms that explicitly incorporate contagion models into their privacy protection guarantees. In an age when activities are shared online and individuals interact with each other more as each day passes, it is not unimaginable that an adversary might have access to social information that can jeopardize individuals’ secrets. We propose a concrete attack that beats DP guarantees. Since many attributes shared by humans are socially correlated, this work reopens many of the problems studied under traditional DP setting (with no explicit assumption on data correlation and generation processes) and forces us to inspect new methods that can endure higher levels of scrutiny.

**Analyzing Social Contagions and Finding Effectors:** Our attack is closely related to analyzing social contagions, in particular the two problems of influence maximization and finding roots of contagion.

Influence maximization is initially studied in [14]: how to pick  $k$  initial seeds such that the number of nodes eventually infected is maximized. It is an NP-hard problem but could be approximated up to  $1 - 1/e$ , if one can have an oracle for computing the *influence* of a set of seeds  $S$  – the (expected) number of nodes infected with seed set  $S$ . Obviously one can run simulations to estimate the influence of a seed set. Computing the exact influence of a node can be done in linear time on a DAG but is #P-hard on a general graph [15]. A heuristic to speed-up the algorithm is to utilize local simple structures, such as local DAGs, to estimate the influence of a node [15]. Alternatively, Borgs *et al.* [16] proposed to use the reverse cascades to estimate influence, picking nodes that more frequently appear in cascades simulated in reverse direction.

Given the current activation state of a contagion in a social network, the *k-effector problem* is to find the most likely  $k$  effectors (initiators of contagions) You can see that influence maximization is a special case of this problem where all nodes are activated in the end. The *k-effector problem* is NP-hard for general graphs or even a DAG, but is solvable in polynomial time by dynamic programming on trees [17]. For general graphs, a heuristic algorithm [18] is to extract the most probable tree (which is NP-hard) and run the optimal algorithm on that tree. Finding effectors is also extensively studied for

the Susceptible-Infected (SI) propagation model [19], [20].

**Privacy of Social Networks and Attributes:** Our work is different from previous work on protecting social network privacy, which assumes that the social network graph itself is private data and network-wide statistics (e.g., degree distribution) is released [21]. We assume that the social network structure is publicly available and only the socially contagious attributes are sensitive.

Links between individuals in a social network can be telling. For instance, Kifer and Machanavajjhala [22] show that future social links can be predicted from the number of inter-community edges by assuming that network evolution follows some particular model. Somewhat similar to our work, Song *et al.* [23] considered flu infection – estimating how many people get flu while preventing the status of any particular individual being revealed. To avoid the intricate details of social contagion, they assume an overly simplistic model where all nodes in the same connected component are correlated and in each component, all pairs of nodes are equally correlated. In our work, we assume a contagion model that is aligned with established literature on contagion and social influence.

### III. PROBLEM DEFINITION

For a population of  $n$  individuals, let  $x_i$  be a sensitive binary attribute for individual  $i$ ,  $x_i \in \{0, 1\}$ , and denote by  $X$  the set of all values  $\langle x_1, \dots, x_n \rangle$ . We assume that this attribute is contagious and propagates over a directed network  $G(V, E)$  following the Linear Threshold cascade model. In this model, each edge has a weight  $w(u, v) \in (0, 1]$  which represents the influence that node  $u$  exerts on node  $v$ . Each node also has a threshold  $\lambda_v$  which is selected *uniformly at random* from  $(0, 1]$ . If the sum of influence from infected incoming neighbors goes beyond  $\lambda_v$ ,  $v$  becomes activated in the next round. Assuming that the set of activated nodes,  $A$ , is not empty at time 0, we can build it iteratively at every step via the following rule:

$$A \leftarrow A \cup \{v \in V \setminus A : \sum_{u \in N_{\text{in}}(v) \cap A} w(u, v) \geq \lambda_v\}. \quad (2)$$

Here  $N_{\text{in}}(v)$  is the set of neighbors with edges pointing to  $v$  (i.e., imposing influence on  $v$ ). The process proceeds until  $A$  stops growing.

Imagine that these individuals participate in a survey in which they are each asked about their sensitive attribute  $x_i$ . The goal of this survey is to calculate some aggregate statistic, e.g., the percentage of individuals having attribute 1. To avoid revealing their secrets, they could use a randomized perturbation mechanism,  $\mathcal{M}: \{0, 1\} \rightarrow \{0, 1\}$ , and use the resulting values to answer the survey. The observed answer of participants is the sequence  $\langle z_1, \dots, z_n \rangle$ , denoted by  $Z$ , where  $z_i = \mathcal{M}(x_i)$ . Assume that  $\mathcal{M}$  guarantees  $(\epsilon, \delta)$ -DP, i.e.,

$$\forall z, x \neq x': \Pr[\mathcal{M}(x) = z] \leq e^\epsilon \Pr[\mathcal{M}(x') = z] + \delta. \quad (3)$$

In this paper we want to examine two problems:

- What is the performance of the best classifier, using only information in  $Z$  and  $\mathcal{M}$ , to infer the true values  $X$ ?

- If we also know the contagion network  $G$  and the contagion model, can we perform better? In other words, how much more information is revealed by knowing the social structure and the way this sensitive attribute propagates in the network? The difference from the answer to the earlier question is the loss of privacy.

### IV. LIMITATIONS OF BINARY CLASSIFICATION WITH DIFFERENTIAL PRIVACY

To show that the presence of the underlying contagion network provides essential information that can pose a real threat to privacy, we study the limits of binary classification given *only* the reported values ( $Z$ ) and the randomization parameters of  $\mathcal{M}$ . This is a fair assumption since the real values,  $X$ , are never disclosed but  $Z$  is, and  $\mathcal{M}$  is known to all participants.

A classifier scores and subsequently ranks the participants based on their likelihood of having  $x = 1$ . We measure the success of such ranking by the probability that a randomly selected sample with  $x = 1$  (a positive sample) is ranked higher than a randomly selected sample with  $x = 0$  (a negative sample). This is known to be the area under the receiver operating characteristic curve (ROC curve) in an unsupervised classification problem, namely the AUC value [24].

**Theorem 1.** *Any classification attempt by an adversary, having access to only  $Z$  and  $\mathcal{M}$ , will have an Area Under the ROC Curve (AUC) at most  $1 - (1 - \delta)/(e^\epsilon + 1)$ .*

Recall that the ROC curve of a classifier is plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. AUC can be understood as the probability that the classifier ranks  $R_1$  higher than  $R_0$ , denoted by  $P(R_1 > R_0)$ , where  $R_1$  ( $R_0$ ) is a randomly chosen positive (negative) sample, with  $x = 1$  ( $x = 0$ ).

Suppose we take a positive (negative) sample  $x$  ( $x'$ ) and the perturbation mechanism  $\mathcal{M}$  produces a perturbed value  $z$  ( $z'$ ). Let's denote  $\Pr(\mathcal{M}(x) = z)$  by  $P(x \rightarrow z)$  for brevity. Let  $\mathcal{S}_1$  and  $\mathcal{S}_0$  be two distributions over  $(-\infty, +\infty)$  from which a score is drawn, if  $z = 1$  or  $z = 0$  respectively. For the perturbed value  $z, z'$ , the classifier chooses a score  $s, s'$  from  $\mathcal{S}_z, \mathcal{S}_{z'}$  respectively and the ranking is produced based on the scores. Denote by  $\gamma(z, z')$  the probability that  $s$  is higher than  $s'$ , i.e.,  $\Pr[s > s' | s \sim \mathcal{S}_z, s' \sim \mathcal{S}_{z'}]$ . Obviously,

$$\Pr[s = s' | s \sim \mathcal{S}_z, s' \sim \mathcal{S}_{z'}] = 1 - \gamma(z, z') - \gamma(z', z).$$

Then we can write  $P(R_1 > R_0)$  as (Section 2 of [25]):

$$\sum_{z, z' \in \{0, 1\}} P(1 \rightarrow z) P(0 \rightarrow z') (P(s > s') + \frac{1}{2} P(s = s')).$$

Continuing the above, we have:

$$\begin{aligned}
P(R_1 > R_0) &= \sum_{z, z'} P(1 \rightarrow z) P(0 \rightarrow z') \left( \frac{1 + \gamma(z, z') - \gamma(z', z)}{2} \right) \\
&= \frac{1}{2} \sum_{z, z'} P(1 \rightarrow z) P(0 \rightarrow z') (\gamma(z, z') - \gamma(z', z)) + \frac{1}{2} \\
&= \left( P(1 \rightarrow 1) P(0 \rightarrow 0) - P(0 \rightarrow 1) P(1 \rightarrow 0) \right) \cdot \\
&\quad \frac{\gamma(1, 0) - \gamma(0, 1)}{2} + \frac{1}{2} \tag{4}
\end{aligned}$$

**Observation 1.** Let  $\gamma^*$  be the one maximizing AUC, i.e.,  $\arg \max_{\gamma} P(R_1 > R_0)$ . Then,  $\gamma^*(1, 0) = 1$ , if

$$P(1 \rightarrow 1) P(0 \rightarrow 0) > P(1 \rightarrow 0) P(0 \rightarrow 1) \tag{5}$$

and 0 otherwise.

The above is clear from the right hand side of (4). This shows that an optimal AUC is achieved by a *deterministic* classification rule based solely on the condition in Observation 1.

**Corollary 1.** Bayesian inference achieves optimal AUC.

*Proof.* By Bayes' rule we have:

$$P(0 \rightarrow z) = \frac{P(z) - P(1 \rightarrow z) P(x = 1)}{P(x = 0)}.$$

Using the above, we can substitute  $P(0 \rightarrow 1)$  and  $P(0 \rightarrow 0)$  in (5). After canceling out phrases from both sides, we have:

$$\begin{aligned}
\frac{P(1 \rightarrow 1) P(z = 0)}{P(x = 0)} &> \frac{P(1 \rightarrow 0) P(z = 1)}{P(x = 0)} \\
\frac{P(x = 1, z = 1)}{P(x = 1) P(z = 1)} &> \frac{P(x = 1, z = 0)}{P(x = 1) P(z = 0)} \\
P(x = 1 | z = 1) &> P(x = 1 | z = 0).
\end{aligned}$$

The proof is symmetrical for the reverse inequality.  $\square$

We can now prove Theorem 1. Without loss of generality, we assume that the condition in Lemma 1 holds, we can then further simplify (4) as below:

$$\begin{aligned}
\Pr(R_1 > R_0) &= \frac{1}{2} + \frac{1}{2} \left( P(1 \rightarrow 1) + P(0 \rightarrow 0) - 1 \right) \\
&= \frac{P(1 \rightarrow 1) + P(0 \rightarrow 0)}{2} \tag{6}
\end{aligned}$$

By the  $\varepsilon$ -DP guarantees we have:

$$\begin{aligned}
P(1 \rightarrow 1) + e^\varepsilon P(0 \rightarrow 0) &\leq e^\varepsilon + \delta \\
P(0 \rightarrow 0) + e^\varepsilon P(1 \rightarrow 1) &\leq e^\varepsilon + \delta, \tag{7}
\end{aligned}$$

As a result, we have:

$$\Pr(R_1 > R_0) \leq 1 - \frac{1 - \delta}{1 + e^\varepsilon}. \tag{8}$$

Thus, Theorem 1 is proved. Note that the bound is realized if the inequalities in Equation (3) become equality. This theorem shows that if this bound is significantly surpassed, the guarantee of  $\varepsilon$ -DP no longer holds.

## V. ALGORITHM

### A. Objective Function

The goal is to infer  $P(x_v = 1)$  for all  $v$ . We denote this probability by  $x_v$  throughout this paper (note the difference between  $x$  and  $x$ ). To do this, we first find the initial seeds of contagion, then calculate the corresponding  $x_v$ . Our solution is hence an arrangement of probabilities of each node  $v$  being initially active, denoted by  $\alpha_v$ . Rather than a fixed number of most likely seeds, we seek to find a *distribution* of initial seeds that are likely to produce the observed  $Z$ . This is shown to significantly boost our performance. We now define the main objective for our problem.

**Definition 1** (Symmetric Difference). Given two instances of reports,  $Z_1$  and  $Z_2$ , we define their Symmetric Difference by:

$$D(Z_1, Z_2) = |Z_1 \setminus Z_2| + |Z_2 \setminus Z_1|. \tag{9}$$

Let  $\alpha$  be an assignment of the initial activation probabilities. Suppose that  $\mathcal{C}$  is the distribution of all possible cascades,  $\mathcal{C}$ , and  $\mathcal{R}$  is the distribution of all possible reports,  $\tilde{Z} = (\tilde{z}_1, \dots, \tilde{z}_n)$ . Then, the *expected* symmetric difference between  $\tilde{Z}$  and the originally observed values,  $Z$ , will be as below:

$$\begin{aligned}
&\mathbb{E} [D(\tilde{Z}, Z)] \\
&= \sum_{\tilde{Z} \sim \mathcal{R}} \Pr(\tilde{Z}) D(\tilde{Z}, Z) \\
&= \sum_{C \sim \mathcal{C}} \Pr(C) \sum_{v \in V} \Pr(\tilde{z}_v \neq z_v | C) \\
&= \sum_{C \sim \mathcal{C}} \sum_{v \in V} \sum_{x \in \{0,1\}} \Pr(\tilde{z}_v \neq z_v | x_v = x) \Pr(x_v = x | C) \\
&= \sum_{v \in V} \sum_{x \in \{0,1\}} \Pr(\tilde{z}_v \neq z_v | x_v = x) \Pr(x_v = x) \\
&= \sum_v (P(1 \rightarrow \sim z_v) - P(0 \rightarrow \sim z_v)) x_v + P(0 \rightarrow \sim z_v) \tag{10}
\end{aligned}$$

In the above  $\sim z_v = 1 - z_v$ , and the last line is due to  $\Pr(x_v = 0) = 1 - x_v$ . We define our objective function as  $f = \mathbb{E} [D(\hat{Z}, Z)]$  and find an  $\alpha$  that minimizes  $f$ . Since  $P(0 \rightarrow \sim z_v)$  is a constant, we can further simplify  $f$  as

$$f = \sum_{v \in V} c_v x_v \quad \text{s.t. } 0 \leq x_v \leq 1, \tag{11}$$

where  $c_v = P(1 \rightarrow \sim z_v) - P(0 \rightarrow \sim z_v)$ .

### B. Bounds on $X$

Suppose that  $X^* = (x_1^*, \dots, x_n^*)$  are the real attribute values.

**Theorem 2.** Let  $\tilde{P}(z)$  be the fraction of vertices reporting 1 and  $c = P(1 \rightarrow 1) - P(0 \rightarrow 1)$ ,

$$\tilde{P}(x) = \frac{\tilde{P}(z) - P(0 \rightarrow 1)}{c}.$$

Then, with high probability<sup>1</sup>:

$$\left| \tilde{P}(x) - \frac{1}{n} \sum x_v^* \right| \leq \sqrt{\frac{\log n}{2nc^2}}. \quad (12)$$

*Proof.* We can treat  $\tilde{P}(z)$  as the mean of  $n$  random variables, representing individual acts of reporting 0 or 1. The *expected* value of  $\tilde{P}(z)$  can be written as:

$$\mathbb{E} \left[ \tilde{P}(z) \right] = \frac{1}{n} \sum x_v^* P(1 \rightarrow 1) + (1 - x_v^*) P(0 \rightarrow 1). \quad (13)$$

Since  $0 \leq z_v \leq 1$  and each individual report is independent of others, we can apply Chernoff's bound. By using (13) we have:

$$\Pr \left[ \left| \frac{1}{n} \sum \left( x_v^* P(1 \rightarrow 1) + (1 - x_v^*) P(0 \rightarrow 1) \right) - \tilde{P}(z) \right| \geq \epsilon \right] \leq e^{-2n\epsilon^2}, \quad (14)$$

Using the definition of  $\tilde{P}(x)$  in (14), we have:

$$\Pr \left[ \left| \tilde{P}(x) - \frac{1}{n} \sum x_v^* \right| \geq \epsilon' \right] \leq e^{-2n\epsilon'^2 c^2}. \quad (15)$$

Recall that  $c = P(1 \rightarrow 1) - P(0 \rightarrow 1)$ . The probability above is asymptotically zero when:

$$\epsilon' = \sqrt{\frac{\log n}{2nc^2}} \quad (16)$$

□

The value of  $\tilde{P}(z)$  can be estimated from data by  $\{v \in V: z_v = 1\} / n$ . We can now update our objective function to accommodate this new constraint:

$$\begin{aligned} \hat{\alpha} &= \arg \min_{\alpha} \sum_{v \in V} c_v x_v \\ \text{s.t.}, 0 &\leq x_v \leq 1, \left| \frac{1}{n} \sum_{v \in V} x_v - \tilde{P}(x) \right| \leq \sqrt{\frac{\log n}{2nc^2}}, \end{aligned} \quad (17)$$

Although our solution finds soft probabilities ( $x_v \in [0, 1]$ ) instead of discrete values ( $x_v \in \{0, 1\}$ ), our experiments show that having this constraint can increase the accuracy of inferred values, especially when the amount of added noise is not extremely high (DP's  $\epsilon$  is not extremely low).

### C. Modelling Contagion

With  $\alpha$ , we want to derive a formula for  $x_v$ , the probability that node  $v$  is active in the end. Computing the influence of contagion given a fixed  $\alpha$  can be done in linear time for a DAG, using the following formula: (Lemma 3 [15]).

$$x_v = \alpha_v + (1 - \alpha_v) \sum_{u \in N_{\text{in}}(v)} w(u, v) x_u. \quad (18)$$

Since the original graph  $G$  is not necessarily a DAG, we find *local DAGs* containing nodes who impose high influence. In

<sup>1</sup>If  $\lim_{n \rightarrow \infty} P(a) = 1$ ,  $a$  happens with high probability.

this way, we try to benefit from the structural simplicity of DAGs, while losing minimal information. The approach of using local structures to approximate the influence in a general graph has been widely used in prior works in the context of influence maximization [15], [26], [27].

---

#### Algorithm 1 Local DAG with target $t$ . (Algorithm 3 [15])

---

**Require:**  $G(V, E)$ , Node  $t$ ,  $\eta$ : Threshold for Inf,  $N_{\text{max}}$ : Max allowed nodes.

**Ensure:**  $D_t(V_t, E_t)$ : The DAG around node  $v$ .

- 1: **Initialization:**  $V_t = \emptyset$ ,  $E_t = \emptyset$ ,  $\forall v \in V$ :  $\text{Inf}(v, t) = 0$ ,  $\text{Inf}(t, t) = 1$ .
  - 2: **while**  $\max_{v \in V \setminus V_t} \text{Inf}(v, t) \geq \eta$  and  $|V_t| \leq N_{\text{max}}$  **do**
  - 3:      $u \leftarrow \arg \max_{v \in V \setminus V_t} \text{Inf}(v, t)$
  - 4:      $E_t \leftarrow E_t \cup \{(u, v) \mid v \in N_{\text{out}}(u) \cap V_t\}$
  - 5:      $V_t \leftarrow V_t \cup u$
  - 6:     **for**  $v \in N_{\text{in}}(u)$  **do**      $\triangleright$  Neighbors' Inf is updated.
  - 7:          $\text{Inf}(v, t) += w(v, u) \text{Inf}(u, t)$
  - 8:     **end for**
  - 9: **end while**
  - 10: **return**  $D_t(V_t, E_t)$
- 

Algorithm 1 starts by the DAG  $D_t$  containing only  $t$ . We then calculate the influence of each node  $v$  on  $t$ , which is the activation probability of  $t$  if only  $v$  was initially active and influence would only spread through nodes already in  $D_t$ . This is denoted by  $\text{Inf}(v, t)$ . At each step, the node outside of  $D_t$  with highest  $\text{Inf}(\cdot)$  is added to  $D_t$  along with its outgoing edges that connect to nodes already in  $D_t$ . This is to ensure that the final  $D_t$  is a DAG. We then update the influence of incoming neighbors of  $v$  that are not yet in  $D_t$ . There can be two stopping criteria to the growing process: (1) When the influence of the most influential node falls below a threshold  $\eta$ , or (2) the number of nodes in  $D_t$  grows bigger than a maximum allowed number,  $N_{\text{max}}$ . If implemented using an efficient priority queue for  $\text{Inf}(v, t)$  values, Algorithm 1 runs in  $O(|E_t| \log |E_t|)$  time. Among nodes in  $D_t$ , the *local* activation probability,  $x_t(v)$ , is as below:

$$x_t(v) = \alpha_v + (1 - \alpha_v) \sum_{u \in N_{\text{in}}(v) \cap V_t} w(u, v) x_t(u). \quad (19)$$

Note that  $x_t(t)$  is the probability that  $t$  is activated only through nodes that are most influential on it and, as a result, can be considered to be a reasonable approximations of  $x_t$ . Our experiments show that this approach in selecting DAGs is essential to achieving high-quality results, and superior to alternative approaches.

Now we can move on to optimizing the objective function  $f$  in Equation (11). More specifically, we need to find  $\frac{\partial f}{\partial \alpha_v}$  for all  $v$ . Chen et al. have established that in a DAG, there is a linear relationship between  $x_t(t)$  and  $x_t(v)$  for all  $v \in V_t$  [15]. The linear factor, which is equal to  $\frac{\partial x_t(t)}{\partial x_t(v)}$  is computed as below:

$$\frac{\partial x_t(t)}{\partial x_t(v)} = \sum_{u \in N_{\text{out}}(v) \cap V_t} w(v, u) (1 - \alpha_u) \frac{\partial x_t(t)}{\partial x_t(u)}. \quad (20)$$

We can compute the above for all nodes  $v \in D_t$  by initially setting  $\frac{\partial x_t(t)}{\partial x_t(t)}$  as 1 and then going through nodes in reverse topological order. Finding this ordering and computing the partial gradients each takes  $O(|E_t|)$  time. Next, we find gradients of  $f$  based on each  $\alpha_v$ . Let  $I_v = \{t \in V : v \in V_t\}$ . Then, by taking the gradient of (11) and applying the chain rule we can write:

$$\begin{aligned} \frac{\partial f}{\partial \alpha_v} &= \sum_{t \in I_v} c_t \frac{\partial x_t(t)}{\partial \alpha_v} = \sum_{t \in I_v} c_t \frac{\partial x_t(t)}{\partial x_t(v)} \frac{\partial x_t(v)}{\partial \alpha_v} \\ &= \sum_{t \in I_v} c_t \frac{\partial x_t(t)}{\partial x_t(v)} \left( 1 - \sum_{u \in N_{in}(v) \cap V_t} x_t(u) w(u, v) \right). \end{aligned} \quad (21)$$

The last line is produced by taking a derivative of (19) by  $\alpha_v$ . Since  $\alpha_v$  does not have any effect on the activation probability of predecessors of  $v$  in any DAG, we can treat the summation on the right side of (19) as a constant with respect to  $\alpha_v$ . Calculating this summation is possible by dynamic programming when nodes are visited in their topological ordering. Computing values in (21) and (20) for all DAGs has a collective runtime of  $O(\sum |E_t|)$ .

## VI. EXPERIMENTS

In this section, we test our method on both synthetic and real-world networks. We demonstrate that our proposed constraints in Section V-B and greedily retrieved DAGs described in Section V-C play a key role in maintaining a high quality for our results. We also investigate attributes that can indicate how vulnerable nodes are to such attacks, namely in-degree, out-degree and PageRank.

### A. Methods

We tested the following methods in our experiments:

- 1) **CO-DAG**: Our main method, which optimizes our objective subject to the constraints in (17) using DAGs retrieved by Algorithm 1.
- 2) **O-DAG**: Similar to CO-DAG, but without enforcing the constraint on  $\sum x_v/n$ .
- 3) **CO-RND**: To show that our selected DAGs are essential to the high quality of our results, we repeat the experiments with a method similar to CO-DAG but with DAGs that grow by adding random neighbors of nodes already in the DAG, until the number of nodes reaches a threshold  $N_{\max}$ .
- 4) **O-RND**: Similar to CO-RND, but without enforcing the constraint on  $\sum x_v/n$ .
- 5) **Lappas+** [17]: The algorithm to find  $k$ -effectors, when  $k$  is known beforehand. Note that by knowing  $k$ , this method has access to more information compared to others.
- 6) **Bayesian**: Simple Bayesian inference as described in Corollary 1.

### B. Datasets

We now introduce the datasets used in our experiments.

**Synthetic Networks**: We use 4 types of randomly generated networks:

- 1) **Core-Periphery** [28]: A random network where nodes in a *periphery* are loosely connected to a dense center. These networks are generated as Kronecker graphs [29] with matrix parameter  $[0.9, 0.5; 0.5, 0.3]$ .
- 2) **Erdos-Renyi**: A random network in which all possible edges have equal probability  $p$  to appear. We set  $p$  such that the expected out-degree of every node will be 5.
- 3) **Power-law**: A network with a power-law degree distribution where  $f(x) \propto x^{-\gamma}$ . We set  $\gamma = 1$  to produce a degree sequence and ran configuration model [30] to obtain a network.
- 4) **Hierarchical** [31]: Random hierarchies generated as Kronecker graphs with matrix parameter set to  $[0.9, 0.1; 0.1, 0.9]$ .

We generate networks with 500 nodes, remove from it self-loops and nodes having both in and out-degrees less than 3 (except for Hierarchical networks). We assign random influence weights in the according to Section V.A of [15]: Random numbers between (0, 1] are assigned to edges, then incoming links to each node is normalized to sum to 1.

TABLE I: Real-world networks.

Network	#Nodes	#Edges
GrQc	2,422	21,842
HepTh	4,909	38,704
Amazon Videos	1,598	8,402
Amazon DVDs	9,488	55,146

**Real-world Networks**: We also test our method on real-world networks that have been extensively studied in the context of influence maximization before (Table I): two co-authorship networks GrQc and HepTh [32] and two co-purchase networks in Amazon, one containing videos and the other DVDs [33], [34].

### C. Experiment Setting

We simulate 10 cascades for each generated network. For each simulated cascade, we choose a specific number of initially active nodes (seed nodes) at random. These numbers are selected in a way that makes it easier to produce cascades of reasonable size in the networks (with at least 1/4 and at most 3/4 of all nodes). For Hierarchical networks, 50 and for the rest of synthetic networks 5 seed nodes were chosen. For real-world networks, we selected 5% of nodes at random since the networks' size greatly varied. We then simulate a cascade using the *triggering set* approach in Theorem 4.6 of [14]: Each edge  $(u, v)$  is kept with probability  $w(u, v)$  and tossed otherwise. Then, any node reachable from any of the seed nodes is deemed active.

To produce differentially private perturbations for the final  $x_v$  values, we choose the Randomized Response (RR) mecha-

TABLE II: AUC values of 7 methods for 3 values of  $\beta$  across all networks. Note that (1) our method outperforms others in almost all cases and (2) Lappas+ fails to pass the upper bound in almost all cases, despite having access to more information.

	Network	$\beta$	$\varepsilon$	Upper Bound	Bayesian	CO-DAG	O-DAG	CO-RND	O-RND	Lappas+
Synthetic	Core-Periphery	0.1	0.201	0.550	0.550	0.575	0.597	0.568	0.587	<b>0.614</b>
		0.3	0.619	0.650	0.648	<b>0.716</b>	0.708	0.689	0.699	0.657
		0.5	1.099	0.750	0.751	<b>0.833</b>	0.803	0.802	0.793	0.695
		0.7	1.735	0.850	0.847	<b>0.904</b>	0.870	0.887	0.868	0.706
		0.9	2.944	0.950	0.951	<b>0.967</b>	0.955	0.965	0.957	0.711
	Erdos Renyi	0.1	0.201	0.550	0.545	<b>0.571</b>	0.555	0.553	0.564	0.527
		0.3	0.619	0.650	0.659	<b>0.704</b>	0.699	0.678	0.685	0.571
		0.5	1.099	0.750	0.752	<b>0.806</b>	0.791	0.781	0.781	0.587
		0.7	1.735	0.850	0.851	<b>0.897</b>	0.874	0.876	0.870	0.611
		0.9	2.944	0.950	0.949	<b>0.967</b>	0.948	0.960	0.955	0.620
	Power-law Graph	0.1	0.201	0.550	0.545	0.590	<b>0.609</b>	0.580	0.587	0.586
		0.3	0.619	0.650	0.646	<b>0.715</b>	0.713	0.689	0.700	0.672
		0.5	1.099	0.750	0.745	<b>0.813</b>	0.805	0.801	0.795	0.701
		0.7	1.735	0.850	0.850	<b>0.890</b>	0.883	0.884	0.876	0.743
		0.9	2.944	0.950	0.949	<b>0.959</b>	0.948	0.939	0.953	0.753
Hierarchical	0.1	0.201	0.550	0.540	<b>0.602</b>	0.584	0.577	0.591	0.534	
	0.3	0.619	0.650	0.652	<b>0.730</b>	0.711	0.699	0.724	0.573	
	0.5	1.099	0.750	0.753	<b>0.821</b>	0.803	0.808	0.796	0.612	
	0.7	1.735	0.850	0.852	<b>0.893</b>	0.883	0.887	0.881	0.669	
	0.9	2.944	0.950	0.949	<b>0.964</b>	0.961	0.960	0.963	0.756	
Real-world	GrQc	0.1	0.201	0.550	0.554	<b>0.577</b>	0.576	0.564	0.561	N/A*
		0.3	0.619	0.650	0.644	<b>0.720</b>	0.718	0.683	0.685	N/A
		0.5	1.099	0.750	0.744	0.833	<b>0.834</b>	0.798	0.797	N/A
		0.7	1.735	0.850	0.849	<b>0.908</b>	0.892	0.892	0.878	N/A
		0.9	2.944	0.950	0.951	<b>0.973</b>	0.960	0.971	0.965	N/A
	HepTh	0.1	0.201	0.550	0.553	<b>0.584</b>	0.579	0.568	0.566	N/A
		0.3	0.619	0.650	0.658	<b>0.733</b>	0.716	0.698	0.703	N/A
		0.5	1.099	0.750	0.751	<b>0.831</b>	0.825	0.793	0.793	N/A
		0.7	1.735	0.850	0.849	<b>0.917</b>	0.893	0.884	0.876	N/A
		0.9	2.944	0.950	0.950	<b>0.972</b>	0.958	0.962	0.962	N/A
	Amazon Videos	0.1	0.201	0.550	0.554	0.598	<b>0.604</b>	0.584	0.585	N/A
		0.3	0.619	0.650	0.649	<b>0.748</b>	0.748	0.706	0.714	N/A
		0.5	1.099	0.750	0.747	<b>0.854</b>	0.836	0.828	0.812	N/A
		0.7	1.735	0.850	0.849	<b>0.919</b>	0.899	0.906	0.894	N/A
		0.9	2.944	0.950	0.950	<b>0.975</b>	0.962	0.974	0.967	N/A
Amazon DVDs	0.1	0.201	0.550	0.551	0.576	<b>0.584</b>	0.567	0.570	N/A	
	0.3	0.619	0.650	0.652	0.735	<b>0.739</b>	0.699	0.707	N/A	
	0.5	1.099	0.750	0.750	<b>0.830</b>	0.829	0.806	0.802	N/A	
	0.7	1.735	0.850	0.851	<b>0.906</b>	0.897	0.894	0.886	N/A	
	0.9	2.944	0.950	0.952	<b>0.963</b>	0.957	0.957	0.941	N/A	

\* Due to the large size of real-world networks, performing experiments with Lappas+ was not viable.

nism [35]. Given a value  $x \in \{0, 1\}$  an RR mechanism works as below:

$$\mathcal{RR}(x) = \begin{cases} x, & \text{with probability } \beta; \\ 1, & \text{with probability } (1 - \beta)/2; \\ 0, & \text{with probability } (1 - \beta)/2. \end{cases} \quad (22)$$

where  $\beta$  is a parameter that determines the rate by which respondents report the *true* value  $x$ . A RR mechanism with parameter  $\beta$  is a DP mechanism with  $\varepsilon = \log((1 + \beta)/(1 - \beta))$ .

To do the inference, we need to extract DAGs for each node. For random DAGs (used in O-RND and CO-RND), we use the average number of nodes in their corresponding greedy DAGs (used in O-DAG and CO-DAG) as node capacity. To optimize (17), we use ALGLIB<sup>2</sup> library.

<sup>2</sup>ALGLIB (www.alglib.net), Sergey Bochkunov

#### D. Going Above the Bound

In Section III we proved that, without extra information about the contagion, it is impossible to achieve an AUC higher than  $1 - (1 - \delta)/(1 + e^\varepsilon)$  on binary attributes perturbed by a  $(\varepsilon, \delta)$ -DP mechanism. Here, we test this guarantee after the contagion graph is known to an adversary. For 5 values of RR's  $\beta$  we perform inference using methods described in Section VI-A and report the resulting AUC scores, the corresponding  $\varepsilon$  and theoretical upper bounds in Table II.

As expected, we observe that the AUC scores for Bayesianis close to the theoretical upper bound. Furthermore, note that in almost all cases CO-DAG achieves the highest AUC score and in *all* cases the value is beyond the AUC upper bound with DP guarantees. For the medium range of  $\beta$  this difference becomes significant. We also observe that in cases where  $\beta$  is extremely low and the perturbed data

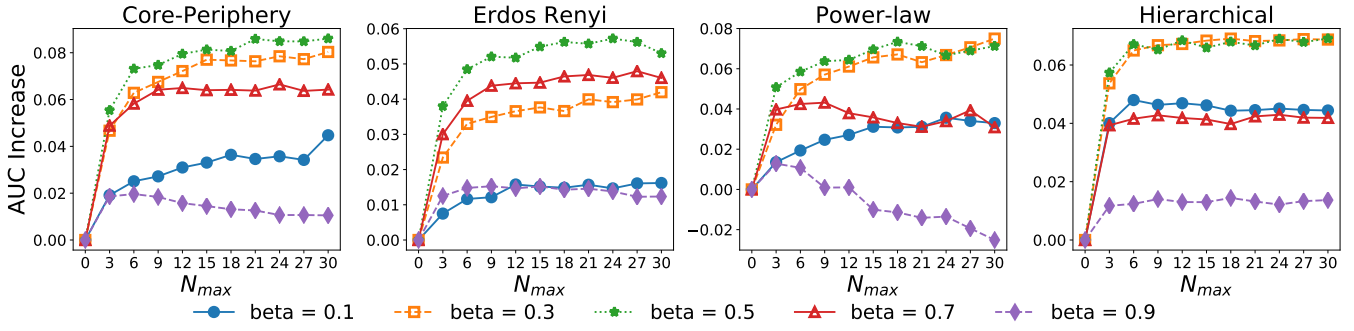


Fig. 1: Impact of maximum allowed nodes ( $N_{\max}$ ) in each DAG on increase in AUC scores relative to  $N_{\max} = 3$ .

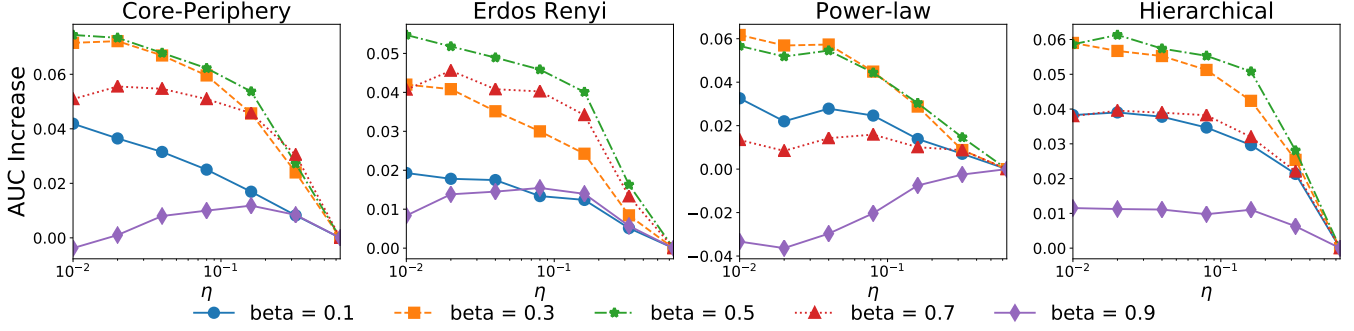


Fig. 2: Impact of threshold on Inf,  $\eta$  on increase in AUC scores relative to  $\eta = 0.01$ .

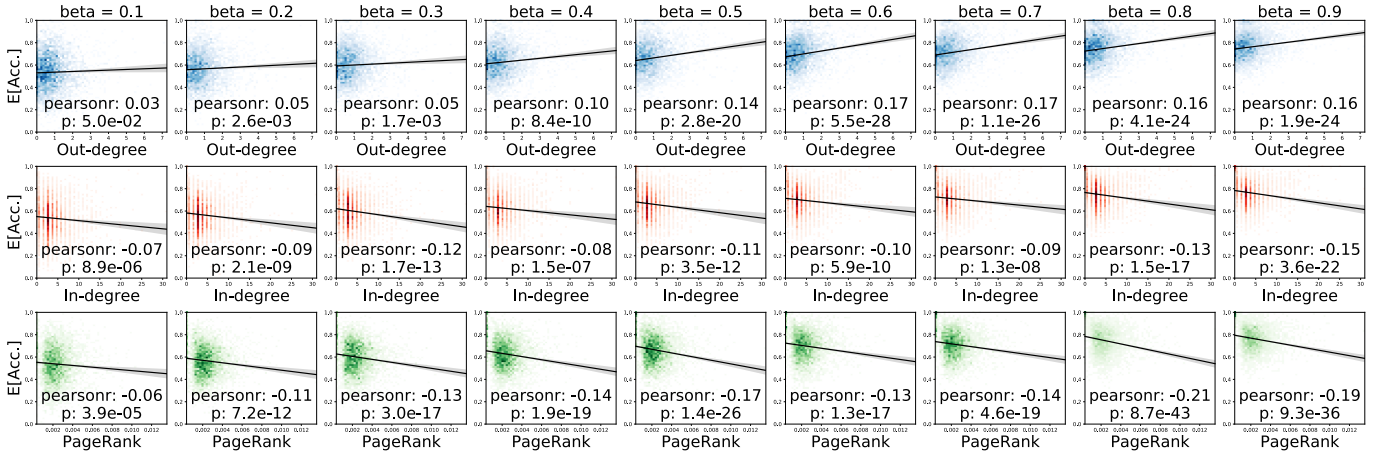


Fig. 3: The correlation between  $\mathbb{E}[\text{Acc}]$  and from top to bottom (1) out-degree, (2) in-degree, and (3) PageRank.

is extremely noisy, O-DAG tends to outperform CO-DAG. We believe that this is due to the deteriorating quality of the bound on  $\sum x_v/n$  when the added noise becomes too big. It is also worth noting that our method's capability to go beyond the bound does not come trivially, since Lappas+ fails to do so in all but one cases, even as it receives *more* information as input (number of seeds,  $k$ ). Finally, notice that in virtually all cases, dropping the greedy DAGs or the constraint on  $\sum x_v/n$  hurts performance.

### E. Impacts of DAG's size on Inference Quality

We now test the impact of changing  $\eta$  (the threshold on nodes' influence) or  $N_{\max}$  (maximum allowed nodes) of DAGs on the quality of inferred values.

We iterate over an increasing  $N_{\max}$  with a fixed RR's  $\beta$  across 4 synthetic networks. Since we are interested in the change in resulting AUC scores and not their absolute values, we move each resulting curve so that their starting points would land on 0.00. Similarly for  $\eta$ , we start from  $\eta = 0.01$  and exponentially grow the threshold to 0.64 while  $\beta$  is fixed. To make it easier to observe the change in AUC scores, we



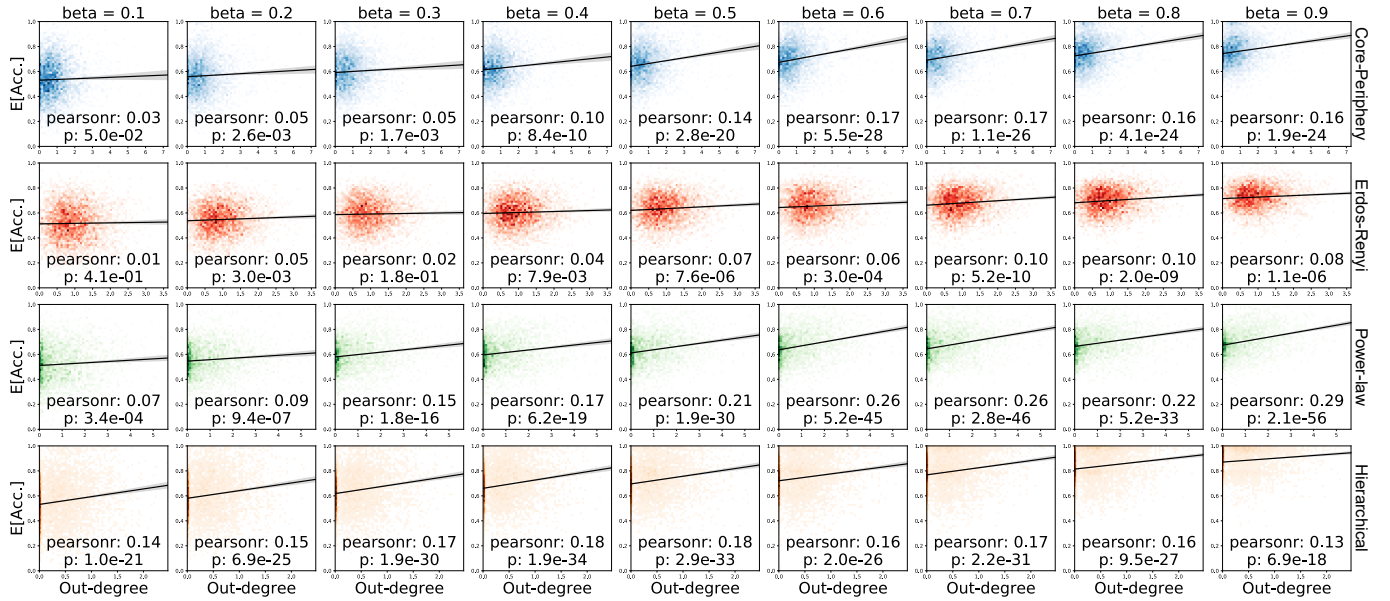


Fig. 4: The correlation between out-degree ( $x$ -axis) and  $\mathbb{E}[\text{Acc}]$ . Each column represents experiments with the same  $\beta$ .

collocate the ending point of all curves on 0.00.

The results for  $N_{\max}$  and  $\eta$  are depicted in Figures 1 and 2 respectively. Intuitively, one might expect that larger DAGs will always lead to more accurate results, albeit with some additional computational cost. This is *almost* true in both experiments. Interestingly, in some of the networks when the perturbation is minimal ( $\beta = 0.9, \varepsilon = 2.94$ ), the quality actually drops as DAGs grow larger. This means that having a more *selective* DAG, with more restrictive criteria, can sometimes yield better results.

#### F. Who is More Vulnerable?

The AUC value describes the effectiveness of our inference algorithm but does not provide details on individual vertex level. Here we look into attributes that can indicate a node's vulnerability to such an inference attack. Without access to ground truth values of  $x_v$ , finding the best cutoff threshold of  $x_v$  for classification is not possible. As an alternative, we calculate the *expected accuracy* of a single node's inferred value if the threshold  $\theta$  is selected at random from  $[0, 1]$ :

$$\mathbb{E}[\text{Acc}] = \mathbb{E}_{\theta \sim [0,1]} [I(x_v = 1)x_v + I(x_v = 0)(1 - x_v)], \quad (23)$$

where  $I(\cdot)$  is an indicator function. Note that we are not interested in the absolute values of  $\mathbb{E}[\text{Acc}]$ , but the relative difference between different nodes, in order to compare nodes' vulnerability to our attack.

We tested 3 node attributes related to the centrality and importance of a node in a complex network: (1) (Weighted) Out-degree, (2) (Weighted) In-degree, and (3) PageRank [36]. We limit our visualizations to Core-Periphery networks due to limited space. Results are similar for the other 3 networks. We test 9 values of  $\beta = \{0.1, 0.2, \dots, 0.9\}$  and infer the real values using CO-DAG. In Figure 3 the resulting  $\mathbb{E}[\text{Acc}]$

for each node  $v$  is plotted against the 3 metrics mentioned above along with correlation analysis printed on each plot. Each column contains plots with a similar  $\beta$  value, printed on the top. Among the 3 metrics, two of them (in-degree and PageRank) show a negative correlation, while a strong correlation is observed between  $\mathbb{E}[\text{Acc}]$  and out-degree which grows even stronger as  $\beta$  is increased and the added noise becomes minimal. A reasonable explanation is that when  $v$  has a high impact on many nodes in the network, it essentially sends signals of its true value to those nodes during the spread of the contagion. These signals, each insignificant on its own, can collectively reveal the true value of  $x_v$  to a great extent. Of course, these signals only grow stronger when the amount of added noise to each report is reduced as  $\beta$  grows bigger. In contrast, there might be many signals on the influence received by a node with a high in-degree, but without that influence spreading around, we only have the report by that node as a single indicator of its true value which is masked well by conventional privacy protection techniques.

We repeat this experiment on other synthetic networks to see if we observe similar results. In Figure 4,  $\mathbb{E}[\text{Acc}]$  is plotted against out-degree of all nodes for  $\beta$  between 0.1 and 0.9. A similar trend is visible across all 4 types of network, with different intensity. In the 3 networks where the distribution of influence is skewed, namely Core-Periphery, Power-law and Hierarchical, the correlation is strong. Moreover, as  $\beta$  increases and the added noise become smaller, the correlation grows stronger. In Erdos-Renyi networks, where degrees are distributed more evenly, the difference in  $\mathbb{E}[\text{Acc}]$  among nodes is minor and as a result, the correlation is weaker. Notice that in Hierarchical network, this *unequal* distribution of influence reaches its extreme point, where the majority of the nodes are leaves, with *no* impact on others, while a handful of nodes at

the top influence many others. This radical difference leads to a constant unbalance between  $\mathbb{E}[\text{Acc}]$  of the two groups across all values of  $\beta$ .

## VII. CONCLUSION AND FUTURE WORK

In this work, we provide further evidence that a privacy-protecting measure that is oblivious to socially contagious properties of attributes is unlikely to provide guarantees in practice as advertised. There are two obvious directions for future research: 1) design a privacy protection mechanism that is aware of the contagious property of some attributes and employs models of contagions in its design; Our method can be used to test if that method succeeds. 2) extend the results to other contagion models. Notice that the extension to any progressive contagion model (where active nodes stay active) with a differentiable formula (e.g, Independent Cascade model) is easily possible. However, ideas specific to those models are required for efficient implementation and model design.

## ACKNOWLEDGEMENT

Aria Rezaei and Jie Gao acknowledge support through NSF DMS-1737812, NSF CNS-1618391 and NSF CCF-1535900.

## REFERENCES

- [1] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, 2014.
- [2] E. Cohen-Cole and J. M. Fletcher, "Is obesity contagious? social networks vs. environmental factors in the obesity epidemic," *J. Health Econ.*, vol. 27, no. 5, pp. 1382–1387, Sep. 2008.
- [3] J. Hamari and J. Koivisto, "'working out for likes': An empirical study on social influence in exercise gamification," *Comput. Human Behav.*, vol. 50, pp. 333–347, Sep. 2015.
- [4] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. I. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler, "A 61-million-person experiment in social influence and political mobilization," *Nature*, vol. 489, no. 7415, pp. 295–298, Sep. 2012.
- [5] J. Meer, "Brother, can you spare a dime? peer pressure in charitable solicitation," *J. Public Econ.*, vol. 95, no. 7, pp. 926–941, Aug. 2011.
- [6] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug. 2011, pp. 1100–1108.
- [7] J. Gao, G. Ghasemiefteh, G. Schoenebeck, and F.-Y. Yu, "General threshold model for social cascades: Analysis and simulations," in *Proceedings of the 2016 ACM Conference on Economics and Computation*, Jul. 2016, pp. 617–634.
- [8] M. S. Granovetter, "The strength of weak ties," *Am. J. Sociol.*, vol. 78, no. 6, pp. 1360–1380, 1973.
- [9] D. Centola and M. Macy, "Complex contagions and the weakness of long ties," *Am. J. Sociol.*, vol. 113, no. 3, pp. 702–734, Nov. 2007.
- [10] G. Ghasemiefteh, R. Ebrahimi, and J. Gao, "Complex contagion and the weakness of long ties in social networks: Revisited," in *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, 2013, pp. 507–524.
- [11] D. Kifer and A. Machanavajjhala, "Pufferfish: A framework for mathematical privacy definitions," *ACM Trans. Database Syst.*, vol. 39, no. 1, pp. 3:1–3:36, Jan. 2014.
- [12] A. Ghosh and R. Kleinberg, "Inferential privacy guarantees for differentially private mechanisms," Mar. 2016.
- [13] V. Rastogi, M. Hay, G. Miklau, and D. Suciu, "Relationship privacy: Output perturbation for queries with joins," in *Proceedings of the Twenty-eighth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2009, pp. 107–116.
- [14] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 137–146.
- [15] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *2010 IEEE International Conference on Data Mining*, Dec. 2010, pp. 88–97.
- [16] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, Dec. 2013, pp. 946–957.
- [17] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, "Finding effectors in social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 1059–1068.
- [18] W. Luo, W. P. Tay, and M. Leng, "Identifying infection sources and regions in large networks," *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2850–2865, Jun. 2013.
- [19] H. T. Nguyen, P. Ghosh, M. L. Mayo, and T. N. Dinh, "Multiple infection sources identification with provable guarantees," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Oct. 2016, pp. 1663–1672.
- [20] B. A. Prakash, J. Vreeken, and C. Faloutsos, "Spotting culprits in epidemics: How many and which ones?" in *2012 IEEE 12th International Conference on Data Mining*, Dec. 2012, pp. 11–20.
- [21] C. Task and C. Clifton, "A guide to differential privacy theory in social network analysis," in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Aug. 2012, pp. 411–417.
- [22] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, 2011, pp. 193–204.
- [23] S. Song, Y. Wang, and K. Chaudhuri, "Pufferfish privacy mechanisms for correlated data," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1291–1306.
- [24] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [25] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
- [26] C. Wang, W. Chen, and Y. Wang, "Scalable influence maximization for independent cascade model in large-scale social networks," *Data Mining and Knowledge Discovery*, vol. 25, no. 3, pp. 545–576, 2012.
- [27] A. Goyal, W. Lu, and L. V. Lakshmanan, "Simpath: An efficient algorithm for influence maximization under the linear threshold model," in *2011 IEEE 11th international conference on data mining*, 2011, pp. 211–220.
- [28] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 4, p. 21, 2012.
- [29] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 985–1042, 2010.
- [30] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.
- [31] A. Clauset, C. Moore, and M. E. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, p. 98, 2008.
- [32] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 2, 2007.
- [33] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 1, p. 5, 2007.
- [34] A. Rezaei, B. Perozzi, and L. Akoglu, "Ties that bind: Characterizing classes by attributes and social ties," in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017, pp. 973–981.
- [35] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.
- [36] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Tech. Rep., 1999.