Noname manuscript No.
(will be inserted by the editor)

Individual Privacy in Social Influence Networks

Sara Hajian · Tamir Tassa · Francesco Bonchi

Received: date / Accepted: date

Abstract On-line social-networking platforms have the possibility to collect an incredibly rich set of information about their users: the people they talk to, the people they follow and trust, the people they can influence, as well as their hobbies, interests, and topics in which they are authoritative. Analyzing this data creates fascinating opportunities for expanding our understanding about social structures and phenomena such as social influence, trust and their dynamics. At the same time, mining this type of rich information allows building novel on-line services, and it represents a great resource for advertisers and for building *viral marketing* campaigns. Sharing social-network graphs, however, raises important privacy concerns. To alleviate this problem, several anonymization methods have been proposed, that aim at reducing the risk of a privacy breach on the published data while still allowing to analyze them and draw relevant conclusions. The bulk of those proposals only considers publishing the network structure, that is a simple (often undirected) graph.

In this paper we study the problem of preserving users' individual privacy when publishing information-rich social networks. In particular, we consider the obfuscation of users' identities in a *topic-dependent social influence network*, i.e., a directed graph where each edge is enriched by a topic model that represents the strength of the social influence along the edge per topic. This information-rich graph is obviously much harder to anonymize than standard graphs. We propose here to obfuscate the identity of nodes in the network by randomly perturbing the network structure and the topic model. We then formalize our privacy notion, k-obfuscation, and show how to evaluate the level of obfuscation under a strong adversarial assumption. Experiments on two social networks confirm that

S. Hajian
Eurecat-Technology Centre of Catalonia, Barcelona, Spain
E-mail: hajian@eurecat.org

F. Bonchi
Eurecat / Algorithmic Data Analytics Lab, ISI Foundation, Turin, Italy
E-mail: francesco.bonchi@isi.it

T. Tassa
The Open University, Ra'anana, Israel
E-mail: tamirta@openu.ac.il

randomization can successfully protect the privacy of the users while maintaining high quality data for applications, such as *influence maximization* for viral marketing.

1 Introduction

Analyzing the structure of social networks is of interest in a wide range of scientific disciplines and applications. However, such practice is limited by the privacy concerns associated with publishing such sensitive information in raw form. The first naïve solution is to de-identify the social graph by removing the identifying information such as the user name, the user id or the email address. However, as shown by Backstrom *et al.* in [4], the mere structure of the released graph may reveal the identity of the individuals behind some of the nodes. Hence, one needs to apply a more substantial procedure of sanitization on the social graph before its release. Following this observation, several anonymization methods have been proposed in the literature, aiming at reducing the risk of a privacy breach on the published data, while still allowing to analyze them and draw relevant conclusions. Almost the totality of this literature (which we survey in Section 2) focuses on simple undirected and non-weighted graphs. Assuming such simple graph structures is overly restricting, as social network data contains much more information than the simple graph, and building real-world applications requires this richer setting.

It is a matter of fact that on-line social-networking platforms have the possibility to collect an incredibly rich set of information about their users: the people they talk to, the people they follow and trust, the people they can influence, as well as their hobbies, interests, and topics in which they are authoritative. Having the possibility to mine this richer set of data, one can get better understanding about social structures evolution, the dynamics of social influence and the interplay among the two [38]. At the same time, mining this type of rich information allows to build novel on-line services, and it represents a great resource for advertisers and for building viral marketing campaigns [26]. In this paper we study the problem of privacy-preserving publication of topic-dependent social influence networks. A topic-dependent social influence network is a directed graph where each edge is enriched by a topic model that represents the strength of the social influence along the edge per topic. Specifically, for each directed edge $e = (v, u)$, representing the fact that u *follows* v, or that v *can influence* u, we have an associated vector $\mathbf{w}_e = (w_e^1, w_e^2, ..., w_e^T)$ where T is the number of topics and w_e^τ represents the strength of influence of v on u for topic $1 \leq \tau \leq T$, or in other terms, the likelihood that information in topic τ will propagate from v to u. (For example, if $e = (v, u)$ and $w_e^\tau = 0.3$, it means that out of, say, 10 actions performed by v and which are related to the topic τ, u is expected to follow v and perform 3 of those actions.)

Consider for example the social network in Figure 1. This toy example describes the social influence relations between Alice, Bob, Carol and David (denoted as A,B,C, and D respectively in the figure), with regard to three topics — (music, sports, politics). Alice is a famous music critic that Bob, Carol and David follow. As can be seen, they are affected by her opinions on music, but her stands on other topics have almost no influence on them. Carol is a friend of both Bob and

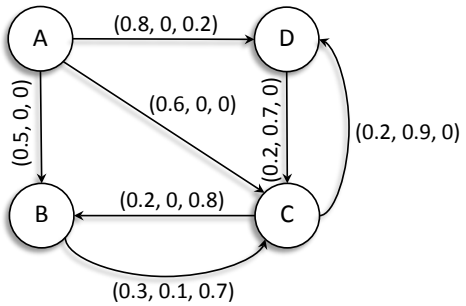

Fig. 1 A social influence network: each directed edge is labeled with a vector that represents the strength of the social influence along the edge for different topics [35,30,6,2,13,3,10].

David, but the latter two do not know each other. Carol is very much involved in sports and politics, and has a mutual high influence in politics on Bob and in sports on David. All three are interested also in music and have mutual influence with regard to that topic too, but it is less strong than Alice's influence.

This type of representation is a concise, yet meaningful summary of activity and behavioral information that is useful in various applications. For instance, it is used for topic-specific influence analysis in social networks in general [30,35] and in microblogs in particular [7,37]. Several authors have analyzed the interplay between topics, linguistic factors, and the dynamic of information and opinions diffusion in `Twitter` [16,34]. Beyond the mere analysis, such topic-enriched graphs can be used for better behavioral targeting and advertising in social networks [17], for influence maximization [2,6,10], or for influence-aware product design [5].

The enriching of edges with topic weights suggest a multi-layering of the network, where each topic induces a different layer of the network with its own set of weighted edges between the nodes. The need to generalize "traditional" network theory by developing and validating a framework and associated tools to study multi-layer systems in a comprehensive fashion arose already few decades above in several disciplines. Since then the study of multi-layer networks has become one of the most important directions in network science. We refer the reader to the review [27] where the authors discuss the history of multi-layer networks, and related concepts, and review the exploding body of work on such networks.

The semantic richness of the data that we want to publish has also a negative effect: the obfuscation of the individual identities becomes a task much harder than in the standard simple graph setting. Consider an adversary that wants to target a specific individual in the released network. For such target individual the adversary might easily know her in-degree (the number of people she follows) and out-degree (the number of her followers).[1] Moreover, the adversary might collect in the same way the target's social activity (e.g., her posts and reposts). As the topic model is published by the data owner (namely, the topics behind each component of the edge vector are known), the adversary can use his background knowledge to estimate the topic-wise influence strength along each incoming or outgoing edge. Such rich information on the target node usually distinguishes that node from all

[1] For instance, in `Twitter` in-degree and out-degree are public information.

other nodes in the network, making the task of preserving the individual privacy of the node by "blending in the crowd" much harder.

In this paper we propose a method for identity obfuscation in social influence networks, under a strong adversarial assumption as described above, by means of random perturbation of the network structure and the associated topic-dependent influence information.

Our contributions are summarized as follows:

- We study, for the first time, the problem of protecting the identity of users when publishing social network data enriched with topic-dependent social influence information, i.e., a topic-dependent social influence network. We propose a method based on random perturbation of the network structure and the associated topic model.
- We then formalize our privacy notion, k-obfuscation, and show how to evaluate the level of obfuscation achieved by the random perturbation method under a strong adversarial assumption.
- We experiment on two real-world datasets, where the topic-dependent influence associated with each social link is learned from real propagations data, using the *expectation-maximization* method of [6]. In our experiments we report the levels of identity obfuscation achieved and the utility preserved in the perturbed data for different levels of strength of the randomization.
- The utility preserved in the randomized data is shown in terms of structural properties of the social graph, and by running *topic-aware influence maximization* queries [6,2,3,10] on the original and on the perturbed graphs.

The rest of the paper is organized as follows. In Section 2 we survey the state of the art in social network anonymization. In Section 3 we describe the data model, introduce our privacy notion, and state our adversarial assumption. Then, in Section 4, we describe the random perturbation method that we use. Section 5 contains the computational derivations for assessing the level of obfuscation which is offered by the perturbed data. Finally, we report in Section 6 our experimental findings, and conclude in Section 7.

2 Related Work

Providing a comprehensive survey of the wide literature on identity obfuscation in social networks is beyond the scope of the current paper. Interested readers may consult the various available surveys [20,39]. The work in this area can be broadly classified in three categories: generalization by clustering nodes [14,21]; deterministic alteration of the graph by edge additions or deletions [12,29,43]; and randomized alteration of the graph by addition, deletion or switching of edges [22, 31,41].

In the last category of methods, which is more relevant to our paper, Hay *et al.* [22] study the effectiveness of random perturbations under degree-based re-identification attack. Given a node v in the real network, they quantify the level of anonymity that is provided for v by the perturbed graph as $(\max_u\{\Pr(v \mid u)\})^{-1}$, where the maximum is taken over all nodes u in the released graph and $\Pr(v \mid u)$ stands for the belief probability that u is the image of the target node v. Their experimentation indicated that the required perturbation for achieving meaningful levels of anonymity reduces utility significantly.

Bonchi *et al.* [9] take a different approach, by considering the entropy of the a-posteriori belief probability distributions as a measure of identity obfuscation.

The rationale is that while using the a-posteriori belief probabilities is a local measure, the entropy is a global measure that examines the entire distribution of these belief probabilities. Bonchi *et al.* show that the entropy measure is more accurate than the a-posteriori belief probability, in the sense that the former distinguishes between situations that the latter perceives as equivalent. Moreover, the obfuscation level quantified by means of the entropy is always greater than the one based on a-posteriori belief probabilities. By experimenting on three large datasets and comparing to Liu and Terzi [29], they demonstrate that random perturbation could be used to achieve meaningful levels of obfuscation while preserving a variety of graph characteristics.

Ying and Wu [42] study the problem of how to generate a synthetic graph that matches given features of a real social network, in addition to a given degree sequence. They propose a Markov Chain-based feature-preserving randomization.

Hanhijärvi *et al.* [18] study a similar problem, in the context of statistical hypothesis testing, albeit not from a privacy perspective. Their goal is to assess the significance of the graph mining results, and therefore they need to ensure that the final randomized samples are taken uniformly at random from the set of all graphs with the same statistics as the original data.

Recently, Boldi *et al.* [8] suggested a new approach for obfuscating graph data by means of randomization, based on injecting uncertainty to the edges of the social graph and publishing the resulting uncertain graph. As opposed to the methods which obfuscate graph data by adding or removing edges entirely, the method of Boldi *et al.* [8] uses finer-grained perturbations that add or remove edges partially, thus possibly achieving comparable levels of obfuscation with smaller changes in the data. While almost all of the above studies focus on simple graphs, Liu *et al.* [31] analyze the case in which the social network edges are weighted, considering two kinds of data characteristics to be preserved for data utility sake: how to keep the lengths of shortest paths between nodes within an error bound, and how to maintain the exact shortest paths between nodes in a selected subset.

Another line of work dealt with the design of differentially private algorithms for graph and network data. Here, the goal is to design probabilistic algorithms for answering queries on graphs so that differential privacy [15] is respected. Some of those algorithms enable to provide perturbed answers for specific (and known upfront) queries on the graph, such as number of triangles, MST cost, degree distribution or small subgraph counts, e.g. [19, 23, 24, 28, 33]. More recent studies [11, 25] provide accurate answers to a larger number of queries but only for a subclass of graphs, such as graphs with a sublinear degree bound, dense graphs, or graphs where the tail of the degree distribution is not too heavy.

Recently, Xiao *et al.* [40] proposed a data sanitization solution that infers a network's structure in a differentially private manner. They observe that, by estimating the connection probabilities between nodes instead of considering the observed edges directly, the noise scale enforced by differential privacy can be reduced.

Differently from the state of the art, in the present work we consider a directed network that is enriched with topic-dependent influence information, which translates into vectorial weights on the edges. Such graphs are by far harder to anonymize, because the data has to be altered significantly in order to achieve meaningful levels of privacy, due to the "curse of dimensionality" [1] (we elaborate on that in Section 3).

Finally, in the context of viral marketing applications, the only work about privacy we are aware of is a recent work by Tassa and Bonchi [36] focussing on *secure multiparty computation* of social influence strength: several players hold different pieces of data – activity logs of different users and the social network – and the goal is to jointly compute the social strength along edges of the network. That goal is very different from the goal of our work, which is publishing a social influence graph while protecting individual privacy.

3 Problem statement

3.1 The data model

Our data model is that of a topic-dependent social influence network. Given a set of T basic topics, such a network describes the influence relations between the users in the underlying population, where the influence depends on the topic.

Definition 1 A topic-dependent social influence network is a directed graph $G = (V, E, W)$ with nodes V, edges $E \subset V \times V$, and vectorial edge weights W. Each directed edge $e = (v, u) \in E$ has an associated vectorial weight $\mathbf{w}_e = (w_e^1, w_e^2, ..., w_e^T)$, where w_e^τ is the probability that user u will be influenced by user v on topic τ, $1 \leq \tau \leq T$.

3.2 The privacy model

Given a population V and the set of topics $\{1, \ldots, T\}$, we let $\Gamma = \Gamma_{V,T}$ denote the set of all social influence networks over V with respect to that set of topics. Furthermore, we let P_Γ denote the set of all probability distributions over Γ. Then the identity obfuscation methods that we consider here are of the form $R : \Gamma \to P_\Gamma$. Namely, given an input network $G \in \Gamma$, the method outputs a perturbed image of G in the form of a network $G' \in \Gamma$ by randomly selecting G' from Γ according to the distribution $R(G)$ on Γ. We assume that the function R is publicly known. In our discussion hereinafter we denote the set of nodes in G' by U, even though it equals the set of nodes in G, which is denoted by V. The introduction of a different notation is needed in order to distinguish between the set of nodes, as observed by the adversary in G', and the set of nodes in the original graph G.

The adversary knows the randomization method and the value of the parameters that control the underlying probability distributions. He also knows some property P of his target node. The goal of the adversary is then to locate the image in U of a specific node v in V, given his knowledge of $P(v)$ and the obfuscated graph G'. For example, let us assume, for the sake of simplicity of illustration, that all vector weights per topic are 1 (namely, for all $e \in E$, $\mathbf{w}_e = (1, \ldots, 1)$), and that the method of randomization just flips the direction of every edge $e \in E$ in some preset probability p. Then the adversary is assumed to know that this is the method of randomization and the value of p that was used in generating G' from G. Let us now consider the case in which the property of the target node v that the adversary knows is its in- and out-degrees, denoted $d^-(v)$ and $d^+(v)$, respectively. Then, as this particular method of randomization does not change the sum of

those degrees, the adversary will be able to limit his search in G' only to nodes u for which $d^-(u)+d^+(u) = d^-(v)+d^+(v)$, while all other nodes will be safely ruled out since they cannot be the image of v in G'. If the value of $d^-(v) + d^+(v)$ was unique in G, then the adversary will be able to locate the image u of v since there will be only one node u in G' with $d^-(u)+d^+(u) = d^-(v)+d^+(v)$. But if the value of $d^-(v)+d^+(v)$ was not unique in G, the adversary will have, say, k suspect nodes u in G', which are all of the nodes in G' for which $d^-(u)+d^+(u) = d^-(v)+d^+(v)$. The adversary may then use his knowledge of the randomization method and the underlying parameters (p in our example) in order to associate probabilities to each of those k suspects as being the image of v. Our basic privacy notion demands that this probability distribution over the suspect nodes will not reveal "too much information" in a sense that we proceed to define formally.

The above discussion implies that an adversary who knows a property P of his target node v, and knows the probabilistic method that was used to generate G' from G, may infer a probability distribution $X_v(\cdot)$ on the nodes in U, where, for every u in U, $X_v(u)$ is the probability that u is the image of v in G'. Specifically, if $v \in V$ is the target node, the adversary may compute for any $u \in U$ the value $f(v,u)$ which denotes the probability that a node with property $P(v)$ was converted, under the known randomization model, to a node with property $P(u)$. Consequently,

$$X_v(u) = \frac{f(v,u)}{\sum_{u' \in U} f(v,u')} \,. \tag{1}$$

For the sake of illustration, in the example above $f(v,u) = 0$ for all nodes $u \in U$ for which $d^-(u) + d^+(u) \neq d^-(v) + d^+(v)$, while for all other nodes $f(v,u)$ may be computed based on the knowledge of $d^{\pm}(v)$ (the adversary's background knowledge), $d^{\pm}(u)$ (the degrees as observed in G'), and p. In cases where $\Delta := d^+(v) - d^+(u) \geq 0$, that probability is given by

$$\sum_{0 \leq i \leq \min\{d^+(v)-\Delta, d^-(v)\}} p^{\Delta+i}(1-p)^{d^+(v)-\Delta-i} p^i (1-p)^{d^-(v)-i} \,.$$

Indeed, if the original in- and out-degrees were $(d^-(v), d^+(v))$ and the observed degrees of u are $(d^-(v)+\Delta, d^+(v)-\Delta)$, then such a transition could have occurred if $\Delta + i$ of the $d^+(v)$ out-going edges were flipped and i of the $d^-(v)$ in-coming edges were flipped, for any value of $i \geq 0$ such that $\Delta + i \leq d^+(v)$ and $i \leq d^-(v)$. A similar formula can be derived when $\Delta := d^+(v) - d^+(u) \leq 0$, since then $d^-(v) - d^-(u) \geq 0$.

We are now ready to define our notion of privacy.

Definition 2 ((k,ε)-Obfuscation) Let $G' = (U = V, E', W')$ be a perturbed image of $G = (V, E, W)$. A node $v \in V$ is said to be k-obfuscated in G' (with respect to adversarial background knowledge P) if the entropy of the random variable X_v over U is at least $\log k$. The graph G' is a (k,ε)-obfuscation of G if at least $(1-\varepsilon)|V|$ of the nodes in G are k-obfuscated in G'.

Consider for example the anonymization method that was proposed by Liu and Terzi [29] in the context of simple (undirected and non-weighted) graphs. They assumed that the adversary knows the degree of the target node and devised methods to convert G into a graph G' in which each degree that appears in the graph is shared by at least k nodes. Hence, even if the adversary is able to infer

correctly the degree of u, the image of v in G', he will not be able to narrow down the number of suspect nodes to less than k. Furthermore, as each of those suspect nodes has the same degree, whence they are all equally likely to be the image of v, the resulting probability distribution X_v will have an entropy of at least $\log k$. Therefore, as their method renders all nodes k-obfuscated, it issues a $(k, 0)$-obfuscation of the input graph (with respect to the adversarial background knowledge of the target node's degree).

Methods that issue output graphs G' in which, for some specific node property, every node property that appears in the graph is shared by at least k nodes are considered to be methods of *k-anonymization*. We adopt here the term *obfuscation* (that was suggested in [9]) instead of *anonymization* since randomized methods cannot guarantee a uniform k-obfuscation over all nodes (see e.g. also [9,22,41]). While this may be considered an advantage of deterministic methods over randomized methods in terms of the resulting privacy, deterministic methods will not alter nodes that are already k-anonymized (as opposed to randomized methods that alter all nodes) and hence such anonymized graphs may be vulnerable to attacks such as that in [4] (see more on that in [9]).

To illustrate this point, let us consider as a motivating example the work of Liu and Terzi [29]. In that paper, the authors devise algorithms for turning the input graph to a k-degree anonymous one, namely, one in which every node degree appears at least k times. If the adversary knows only the degree of his target node, he will not be able to trace it down to a subset of less than k nodes in the released anonymized graph. However, in typical real-world graphs, say ones where the degree distribution follows a power law, almost all nodes satisfy already the k-degree anonymity requirement, with the possible exception of a small number of hubs. Thus, an algorithm for k-degree anonymity has only to adjust the degrees of those hubs (by adding or deleting edges) while leaving the majority of the other nodes unaffected. Therefore, the proposed algorithms in Liu and Terzi [29] may leave the majority of the nodes unaffected. Hence, an adversary who has additional knowledge on his target node (say, connections between that node's neighbors), might be able to use such information in order to identify the target node in the released graph. Such attacks, as described in Backstrom et al. [4], cannot be launched if all nodes are affected in a randomized manner, as we do in obfuscation.

In addition, achieving k-anonymity in rich structures such as topic-dependent social influence networks, without stripping the data of its utility, seems to be a hefty challenge, because any attempt to achieve that will bring about the "curse of dimensionality" [1]. Namely, if the node property that needs to be anonymized is high-dimensional (as is the case with the property $\mathbf{I}(\cdot)$ that we describe in the next subsection, see Eq. (2)) then any attempt to "bucketize" them in buckets of size at least k and then forcing the property of all nodes in the same bucket to have the same value will require too many alterations to the input graph and that can obliterate the graph's utility.

3.3 Adversarial assumption

Our privacy model and obfuscation method are not limited to any particular adversarial assumption. However, once the data owner makes an assumption regarding the property that the adversary knows about his target individual, he needs to

derive the corresponding probability ensemble, $\{X_v : v \in V\}$, in order to assess the obtained level of obfuscation (namely, k and ε). Herein we assume that the property which the adversary knows is the 1-neighborhood of v. Specifically, the adversary knows the in- and out-degrees of the target node v; for example, in Twitter it would mean that the adversary knows how many followees and how many followers the target individual has. In addition, he knows the weights per topic along each of the edges in which v is involved.

Let $v \in V$ be a node in the network. Then the 1-neighborhood assumption means that the knowledge of the adversary about his target node v is

$$\mathbf{I}(v) := \left(d^-(v), d^+(v), \mathbf{w}_1^-(v), \ldots, \mathbf{w}_{d^-(v)}^-(v), \mathbf{w}_1^+(v), \ldots, \mathbf{w}_{d^+(v)}^+(v)\right), \qquad (2)$$

where:

- $d^-(v)$ and $d^+(v)$ denote v's in- and out-degrees;
- $\mathbf{w}_j^-(v)$ is the vector of probabilities along the jth edge that is incoming to v, $1 \le j \le d^-(v)$. Specifically, if that edge is $e = (u, v)$, then $\mathbf{w}_j^-(v)$ is a vector of the form $(w_e^1, \ldots, w_e^T)$, where w_e^τ is the probability that v is influenced by u on topic τ, $1 \le \tau \le T$.
- Similarly, $\mathbf{w}_j^+(v)$ is the vector of probabilities along the jth edge that is outgoing from v, $1 \le j \le d^+(v)$. If that edge is $e = (v, u)$, then $\mathbf{w}_j^+(v)$ is a vector of the form $(w_e^1, \ldots, w_e^T)$, where w_e^τ is the probability that v influences u on topic τ, $1 \le \tau \le T$.

We note that this is a strong adversarial assumption, since the adversary is assumed to have gained knowledge not only on the number of influence links in which v is involved, but also on the strength of each of those links per each of the topics. Similar analysis can be carried out also for other adversarial assumptions that are weaker. We chose to focus on the above strong assumption in order to demonstrate the capabilities of our proposed method of obfuscation with respect to both obfuscation level and data utility.

4 Obfuscation by randomization

The method of obfuscation which we propose, analyze and test here has two phases:

- **Edge sparsification.** The data owner selects a probability $p \in (0, 1)$. Then, for each edge e in E the data owner performs an independent Bernoulli trial, $B_e \sim B(1, p)$. He will remove the edge in the graph in case of success (i.e., $B_e = 1$) and will leave it otherwise ($B_e = 0$). We let $E' = \{e \in E \mid B_e = 0\}$ be the subset of edges that passed this selection process.
- **Random weight reduction.** Let q be some preset integer and let $\phi(\cdot)$ be a probability distribution over $\Omega_q := \{j/q : 0 \le j \le q\}$. Then for every edge $e \in E'$ and topic $1 \le \tau \le T$, the data owner randomly selects $r_{e,\tau} \in \Omega_q$ according to the probability distribution $\phi(\cdot)$ and then he replaces w_e^τ, the τ-weight of e, with the reduced weight $r_{e,\tau} w_e^\tau$. We let W' denote the resulting reduced vectorial weights over the edges of E'.

The data owner then releases the topic-dependent social influence network $G' = (U = V, E', W')$. A good choice for $\phi(\cdot)$ would be a monotonically increasing

Algorithm 1 GenerateObfuscation

Input: $G = (V, E, W)$, p, and b
Output: $G' = (U = V, E', W')$

$E' \leftarrow E$
for all $e \in E$ **do**
  draw B_e at random from Bernoulli trial $B(1, p)$
  if $B_e = 1$
  then $E' \leftarrow E' \setminus \{e\}$
end for
Compute probability distribution $\phi(\cdot)$, using b
$W' \leftarrow W$
for all $e \in E'$ **do**
  for all $1 \leq \tau \leq T$ **do**
    draw $r_{e,\tau}$ at random from $\phi(\cdot)$
    $w_e^\tau \leftarrow r_{e,\tau} w_e^\tau$
  end for
end for
return $G' = (U = V, E', W')$

function, since such functions favor reduction factors closer to 1, and therefore they are expected to harm less the utility of the resulting graph G'. Namely, if $r_1 < r_2$, it is desirable that r_2 will have more chances to be selected than r_1, since using r_2 changes the weights less than using r_1. A simple family of such probability distribution functions ϕ consists of functions that are zero up to some threshold and then they are linearly increasing. Specifically, fix an integer $0 \leq b \leq q-1$ and then define $\phi(j/q) = \frac{2(j-b)_+}{(q-b)(q-b+1)}$ for all $0 \leq j \leq q$, where $(x)_+ := \max(x, 0)$. That function is zero for all $j = 0, \ldots, b$ and then linearly increases for $j = b+1, \ldots, q$. The scaling parameters were chosen so that ϕ is a probability distribution over Ω_q, in the sense that $\sum_{x \in \Omega_q} \phi(x) = 1$. Hence, the two parameters that control the strength of randomization are p, which controls the strength of edge sparsification, and b, which controls the strength of the random weight reduction. Algorithm 1 details our obfuscation method. Its computational cost depends linearly on the number of edges $|E|$ and the number of topics T, and is bounded by $O(|E| \cdot T)$.

We focus here on randomization methods that are based on edge sparsification. We do not consider edge perturbation, an operation that consists of edge deletions as well as edge additions, since edge additions requires also "inventing" fake weights for the added edges. Those weights should be selected in a manner that will not enable identifying the fake edges. While such a selection is not hard to make, the privacy-related probability analysis that follows becomes much more intricate. The study of methods that are based also on random addition of edges and the potential benefits from such an extended framework of random perturbations is left for future research. Moreover, we mention in this context the study of Mathioudakis *et al.* [32]; they apply (non-random) edge sparsification as a pre-processing step for the influence-maximization problem, showing that computations on sparsified models give up little accuracy, but yield significant improvements in terms of efficiency and scalability.

5 Quantifying the level of obfuscation

Here we describe how to compute the probability distributions $\{X_v : v \in V\}$ with respect to the 1-neighborhood property. Let $v \in V$ be a target node in V and assume that the adversary knows the information $\mathbf{I}_v := \mathbf{I}(v)$ about it (see Equation (2)). Let $u \in U$ be a candidate node in U with property $\mathbf{I}_u := \mathbf{I}(u)$. In Section 3.2 we defined $f(v,u)$ as the probability that a node with the known property $P(v)$ was converted under the known randomization method to a node with property $P(u)$. Then in our case, where $P(\cdot) = \mathbf{I}(\cdot)$, $f(v,u)$ equals the following conditional probability,

$$f(v;u) = \Pr(\mathbf{I}(u) = \mathbf{I}_u \mid \mathbf{I}(v) = \mathbf{I}_v, v \mapsto u)\,, \tag{3}$$

where $v \mapsto u$ means that u is the image of v in G'. Namely, given that v has property $\mathbf{I}(v) = \mathbf{I}_v$ and its image in G' is u, $f(v,u)$ is the probability that u's property is $\mathbf{I}(u) = \mathbf{I}_u$. (In order to avoid cumbersome notations we shall drop the notation $v \mapsto u$ from the conditional probabilities henceforth; it is always assumed that v and u are a preimage and image pair.) That probability is given by

$$f(v;u) = \prod_{s=\pm} \Big[\Pr(d^s(u) \mid d^s(v)) \cdot \Pr(\mathbf{w}_1^s(u), \ldots, \mathbf{w}_{d^s(u)}^s(u) \mid \mathbf{w}_1^s(v), \ldots, \mathbf{w}_{d^s(v)}^s(v)) \Big]\,, \tag{4}$$

where

- $\Pr(d^s(u) \mid d^s(v))$ is the probability that a node with an s-degree $d^s(v)$ was converted in the edge sparsification phase to a node with an s-degree $d^s(u)$, where $s = \pm$ (referring to either in-coming or out-going edges).
- Under the assumption that v, of an s-degree $d^s(v)$, was converted to u that has an s-degree $d^s(u)$, $\Pr(\mathbf{w}_1^s(u), \ldots, \mathbf{w}_{d^s(u)}^s(u) \mid \mathbf{w}_1^s(v), \ldots, \mathbf{w}_{d^s(v)}^s(v))$ is the probability that the random weight reduction phase changed the weights on the s-going edges adjacent to v (those are given by $\mathbf{w}_1^s(v), \ldots, \mathbf{w}_{d^s(v)}^s(v)$) to the weights on the s-going edges adjacent to u (those are given by $\mathbf{w}_1^s(u), \ldots, \mathbf{w}_{d^s(u)}^s(u)$).

For either $s = \pm$, $d^s(u) \sim B(d^s(v), 1-p)$, where $B(d^s(v), 1-p)$ is the Binomial distribution over $d^s(v)$ experiments and success probability $1-p$. Hence, the probability that the s-degree of u is $d^s(u)$, given that the s-degree of v is $d^s(v)$, is

$$\Pr(d^s(u) \mid d^s(v)) = \binom{d^s(v)}{d^s(u)} (1-p)^{d^s(u)} p^{d^s(v) - d^s(u)}\,. \tag{5}$$

Note that the probability in Equation (5) is zero when $d^s(u) > d^s(v)$.

Next, we compute the probability $\Pr(\mathbf{w}_1^s(u), \ldots, \mathbf{w}_{d^s(u)}^s(u) \mid \mathbf{w}_1^s(v), \ldots, \mathbf{w}_{d^s(v)}^s(v))$. We may assume that $d^s(u) \le d^s(v)$ since, otherwise, $f(v,u) = 0$ in view of Equations (4) and (5). The adversary does not know the mapping between the $d^s(v)$ s-going edges adjacent to v and the $d^s(u)$ s-going edges adjacent to u. When he considers the possibility that $v \mapsto u$, he needs to take into account all possible mappings. There are $A_{d^s(u)}^{d^s(v)} = d^s(v)!/(d^s(v) - d^s(u))!$ mappings from the $d^s(v)$ s-going edges adjacent to v to the $d^s(u)$ s-going edges adjacent to u, because such a mapping consists of first selecting $d^s(u)$ edges out of the $d^s(v)$ s-going edges adjacent to v and then ordering them (by choosing which of the $d^s(u)$ selected s-going edges adjacent to v corresponds to which of the $d^s(u)$ s-going edges adjacent to u).

Algorithm 2 Testing a node for being k-obfuscated

Input: $\mathbf{I}(v)$ for a $v \in V$, $\{\mathbf{I}(u) : u \in U\}$, k
Output: 1 if v is k-obfuscated and 0 otherwise

for all $u \in U$ **do**
  $f(v,u) \leftarrow \Pr(d^-(u) \mid d^-(v)) \cdot \Pr(d^+(u) \mid d^+(v))$
  if $f(v,u) = 0$ **then** continue
  for all $s = \pm$ **do**
    generate $\hat{\Psi}$, a random sample from Ψ
    for all $\psi \in \hat{\Psi}$ **do**
      set the vectors $\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{d^s(u)}$ as selected by ψ
      compute C_ψ by Equation (6)
    end for
    $\tilde{\mu} \leftarrow \frac{1}{|\hat{\Psi}|} \sum_{\psi \in \hat{\Psi}} C_\psi$
    $f(v,u) \leftarrow f(v,u) \cdot \tilde{\mu}$
  end for
end for
for all $u \in U$ **do**
  $X_v(u) = \frac{f(v,u)}{\sum_{u' \in U} f(v,u')}$
end for
compute the entropy of X_v
return $\{entropy(X_v) \geq \log k\}$

Let Ψ be the set of all $A_{d^s(u)}^{d^s(v)}$ possible mappings from the $d^s(v)$ s-going edges adjacent to v in G to the $d^s(u)$ s-going edges adjacent to u in G', and let ψ be any of the mappings in Ψ. Assume that the mapping ψ maps $d^s(u)$ s-going edges adjacent to v in G, with vectorial weights $(\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{d^s(u)})$, to the s-going edges adjacent to u in G' that have vectorial weights $(\mathbf{w}_1^s(u), \ldots, \mathbf{w}_{d^s(u)}^s(u))$. (Note that $(\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{d^s(u)})$ is any of the $A_{d^s(u)}^{d^s(v)}$ *ordered* selections of $d^s(u)$ vectors out of $(\mathbf{w}_1^s(v), \ldots, \mathbf{w}_{d^s(v)}^s(v))$.) Then the probability of reducing the ordered set of vectorial weights $(\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{d^s(u)})$ to $(\mathbf{w}_1^s(u), \ldots, \mathbf{w}_{d^s(u)}^s(u))$ is given by

$$C_\psi := \Pr(\mathbf{w}_1^s(u), \ldots, \mathbf{w}_{d^s(u)}^s(u) \mid \hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{d^s(u)}) = \prod_{i=1}^{d^s(u)} \prod_{\tau=1}^{T} \phi\left(\frac{w_i^\tau}{\hat{w}_i^\tau}\right), \qquad (6)$$

where, for any $1 \leq i \leq d^s(u)$, $\mathbf{w}_i^s(u) = (w_i^1, \ldots, w_i^T)$ and $\hat{\mathbf{w}}_i = (\hat{w}_i^1, \ldots, \hat{w}_i^T)$. The equality in Equation (6) stems from the fact that the weight reduction along any of the edges e in any of the topics τ is done by an independent selection of a reduction factor $r_{e,\tau}$, as explained in Section 4. Now, we conclude that

$$\Pr(\mathbf{w}_1^s(u), \ldots, \mathbf{w}_{d^s(u)}^s(u) \mid \mathbf{w}_1^s(v), \ldots, \mathbf{w}_{d^s(v)}^s(v)) = \frac{1}{A_{d^s(u)}^{d^s(v)}} \sum_{\psi \in \Psi} C_\psi := \mu\,. \qquad (7)$$

In summary, the probability $f(v,u)$ is given by Equations (4), (5), (6) and (7). Given the probabilities $f(v,u)$ for all $v \in V$ and $u \in U$, we can derive the probability distributions X_v on U by Equation (1).

The computational bottleneck is Equation (7), since the number of possible mappings is exponential in $d^s(u)$. Hence, we suggest replacing the exact computation of the probability in that equation with a corresponding estimate. Instead of averaging over all $A_{d^s(u)}^{d^s(v)}$ possible mappings, we randomly sample a multiset $\hat{\Psi}$ of mappings from Ψ and then average over the sampled mappings. Therefore,

Algorithm 3 Testing the obfuscation level of a graph

Input: $G = (V, E, W)$, $G' = (U = V, E', W')$, k
Output: The minimal ε such that G' is a (k, ε)-obfuscation of G

```
1: n ← 0
2: for all v ∈ V do
3:     x ←k-obfuscated(I(v),{I(u) : u ∈ U}, k)
4:     if x = 0 then n ← n + 1
5: end for
6: ε ← n/|V|
7: return ε
```

instead of the average μ that is given on the right hand side of Equation (7), we suggest computing the approximate average $\tilde{\mu} := \frac{1}{|\hat{\Psi}|} \sum_{\psi \in \hat{\Psi}} C_\psi$. In this approximate computation there is a tradeoff between the efficiency of the computation and its accuracy. Larger sample sets $\hat{\Psi}$ will result in better estimates $\tilde{\mu}$ at the cost of higher computational costs. Invoking the Chernoff bound we arrive at the conclusion that for every $\delta \geq 0$, $\Pr[\tilde{\mu} \geq (1+\delta)\mu] \leq \exp\left(-\frac{\delta^2 |\hat{\Psi}| \mu}{2+\delta}\right)$ and $\Pr[\tilde{\mu} \leq (1-\delta)\mu] \leq \exp\left(-\frac{\delta^2 |\hat{\Psi}| \mu}{2}\right)$. We see that for a given setting of the desired accuracy level δ, the effect of $|\hat{\Psi}|$ on accuracy is exponential. Using the above bounds it is possible to determine the required sample size $|\hat{\Psi}|$ such that the probability of $\tilde{\mu}$ deviating from μ by a fraction of at most δ would not be larger than some desired threshold.

Algorithm 2 tests a given target node $v \in V$ for being k-obfuscated in G'. (For simplicity, we assume that the randomization parameters p and b are known, hence we do not specify them as input parameters.) Algorithm 3 uses Algorithm 2 as a subroutine in order to test the level of obfuscation provided by G' for G. Specifically, it accepts an obfuscation level k and outputs the fraction ε of nodes in V that are not k-obfuscated in G'. The computational cost of Algorithm 3 is bounded by $O(|V|^2 \cdot T \cdot |\hat{\Psi}|)$. In large graphs such a cost may be prohibitive, in which case the testing of obfuscation could be restricted to a small subset of V, being either a random subset or a subset of nodes whose privacy is considered to be more sensitive. Given k and ε, it is impossible to determine analytically what are the best values of p and b that would yield a (k, ε)-obfuscation. However, the data owner can search for such values numerically. To that end, he may start with some setting of p and b and then compute the (k, ε) parameter curve of the resulting obfuscation. (Note that any given obfuscation G' is a $(k, \varepsilon(k))$-obfuscation of G, for any setting of $k \geq 1$, where $\varepsilon(k)$ is as computed by Algorithm 3 on the input parameters G, G', and k.) If the resulting obfuscation level is not satisfactory, then he may increase p or decrease b in order to achieve higher obfuscation levels. If, on the other hand, the obfuscation level is higher than needed, he may decrease p or increase b in order to preserve more utility.

A concluding remark. We assume above that the value of q, as introduced in Section 4, is known to the adversary. As a result, the adversary knows that the possible weight reduction factors are only of the form j/q for some integer $j \leq q$. The adversary may then rule out possible links between edges in the original and perturbed graphs, if the ratio between one of the observed weights and the corresponding known original weight is not of the form j/q for some integer $j \leq q$.

Table 1 Dataset characteristics.

Dataset	*# Nodes*	*# Edges*	*# Topics*
Flixster	6,353	84,607	10
Digg	11,141	99,845	10

(Namely, in Equation (6) $\phi\left(w_i^\tau/\hat{w}_i^\tau\right) = 0$ if the ratio $w_i^\tau/\hat{w}_i^\tau$ does not equal j/q for an integer j.) Using a large value of q reduces the significance of such a background knowledge. (We used in our experiments $q = 1000$, see Section 6.1.) However, we may also assume that q is a random and secret parameter in order to deprive the adversary from that background knowledge. By doing so we may increase the privacy guarantees of our method.

6 Experiments

In this section we report our experimental assessment of the effects of random sparsification and random weight reduction on the structure of the perturbed graph, as well as on the level of obfuscation achieved[2]. In all experiments we assume an adversary that uses $\mathbf{I}_v$ as the re-identification property (as defined in Section 3), knows the randomization method and the value of the randomization parameters p, q and b. We also assess the utility of the perturbed topic-dependent social influence network w.r.t. the influence maximization task compared to the original one. We next explain how our datasets (the social network and the topic model on the edges) are generated from real data.

One important problem in the area of social influence and the phenomenon of influence-driven propagations is to exploit the synergy of topic modeling and influence analysis to take into account the characteristics of the item propagating through the links of the social network. For example, some people are more likely to buy a product than others, *e.g.*, teenagers are more likely to buy videogames than seniors. Similarly, a user who is influential with respect to classic rock music is not very likely to be influential for what concerns techno music too. To that end, Barbieri *et al.* [6] introduce a topic-aware influence-driven propagation model, the *Topic-aware Independent Cascade* (TIC) model, that experimentally turns out to be more accurate in describing real-world cascades than the standard (topic-blind) propagation models studied in the literature (more details are given later in Section 6.3). In our work we adopt this model of social influence network. In the TIC model, given the social graph $G = (V, E)$ and a number of topics T, the topic-dependent influence probabilities $\mathbf{w}_e = (w_e^1, w_e^2, ..., w_e^T)$ for each directed edge $e \in E$ can be learned from a log of past propagations $\mathbb{D} = \{(User, Item, Time)\}$, where $User \in V$. Learning these parameters, in order to maximize the likelihood of $\mathbb{D}$ given the TIC model over the social graph G, is done by means of the *expectation-maximization method* developed in [6].

For our experiments, we use *two real-world publicly available datasets*, both containing a social graph G and a log of past propagations $\mathbb{D}$, and apply the above mentioned method to learn the topic-dependent influence probabilities.

[2] The datasets we use for experiments, as well as our random obfuscation software and the software for quantifying the level of obfuscation achieved, are available at `http://bit.ly/1pt9NJp`

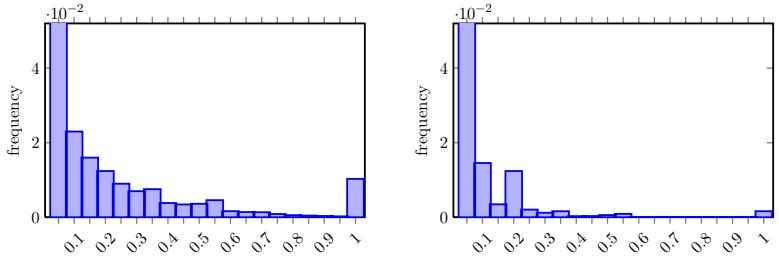

Fig. 2 Distribution of w_e^τ: Flixster (left), Digg (right). Note that the first bar of both histograms is cut: in Flixster 89% of the probabilities are ≤ 0.05, while in Digg this number grows to 96%.

The datasets come from Digg (`www.digg.com`) and Flixster (`www.flixster.com`). Digg is a social news website, where the users vote for stories. In this case $\mathbb{D}$ contains information about which user voted for which story (item) at which time. If we have user v vote for a story about the new iPhone, and shortly later v's friend u does the same, we consider the story as having propagated from v to u, and v as a potential influencer for u. Flixster is one of the main players in the mobile and social movie rating business. Here, an item is a movie, and the action of the user is rating the movie. The main characteristics of the datasets are provided in Table 1, while Figure 2 plots the distribution of the influence probabilities w_e^τ for all the $e \in E$ and all $1 \leq \tau \leq T$ for the two datasets.

6.1 Obfuscation level

The first objective of our experimental evaluation is to show the level of obfuscation achieved for the different values of p and b. We compute the obfuscation level of the perturbed graph obtained by random sparsification and weight reduction and averaged over fifty random runs, and compare with the original graph. Figure 3 reports the level of k-obfuscation on Flixster (left) and Digg (right) for various settings of the parameters b and p. Each curve corresponds to a given setting of p and b, where one of the curves corresponds to $p = 0$ and $b = q - 1$, namely, to the trivial perturbation that retains the original graph unchanged. (In all of our experiments we used $q = 1000$.) A point (k, n) on any of those curves means that n percent nodes did not reach the obfuscation level k, or, alternatively, that the corresponding perturbed graph respects (k, ε)-obfuscation with $\varepsilon = \frac{n}{|V|}$. From these plots we can observe that the number of non-obfuscated nodes in the original topic-dependent social influence networks is very high. This is the consequence of publishing also the topic model, which essentially makes every individual in the social network *unique*. This also motivates the substantial need for applying perturbation before publishing such rich graphs.

As shown in Figure 3, the level k of obfuscation that each of the nodes has in the perturbed data is significantly higher than the level of obfuscation that it has in the original topic-dependent social influence network. As expected, the levels of obfuscation increase with larger p and smaller b. In the Digg dataset, more nodes are easily obfuscated thanks to the many more topic-dependent probabilities which are close to 0 (as reported in the caption of Figure 2). Note that Figure 3 reports the level k of obfuscation as computed using a sample size of $|\hat{\Psi}| = 100$.

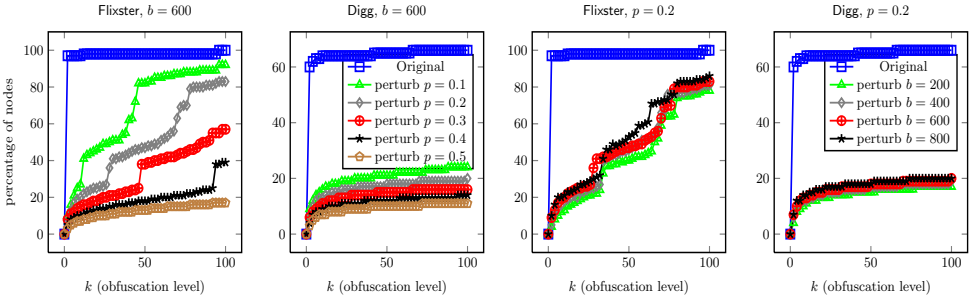

Fig. 3 The level of k-obfuscation on Flixster and Digg, for $b = 600$ and varying values of p (left), and for $p = 0.2$ and varying values of b (right). The x-axis shows the obfuscation level and the y-axis shows the percentage of nodes that did not reach such obfuscation level. The minimum and maximum standard deviations of the reported values over fifty random runs are $(0.0002, 0.083)$, $(0.0001, 0.0038)$, $(0.0018, 0.0780)$ and $(0.0002, 0.0037)$, from the left to right plots, respectively.

We performed the same experiments for sample sizes 200, 300 and 400 and we obtained similar results. That similarity of results shows that a small sample size is sufficient to capture the probability μ (see Equation (7)).

6.2 Graph structure statistics

The second objective of our experimental evaluation is to show that our randomization technique can preserve to a large extent the structure properties of the original social graph. Thus, we first compare some graph statistics when measured on the original graph and on its perturbed versions, for varying values of p. We measure the following statistics:

- Average in(out)-degree: $\frac{1}{|V|}\sum_{v\in V} d^-(v) = \frac{1}{|V|}\sum_{v\in V} d^+(v)$.
- Clustering coefficient: the fraction of closed triplets of nodes among all connected triplets (when considering the graph as undirected).
- Average distance: the average distance among all pairs of nodes that are directed path-connected.
- Diameter: the maximum distance among all directed path-connected pairs of nodes.

In Figure 4 we report all of the above statistics, for different values of p, on the Flixster and Digg datasets, averaged over fifty random runs. It is worth noting that in this experiment the value of the parameter b is not relevant as we focus only on the graph structure and not on the associated topic model (edge weights).

Note that changing p has an effect on the resulting parameters (k, ε) of the obfuscation. Therefore, for the sake of illustration, the legend of the x-axis in the plots in the first column has three rows: the value of p (top), the corresponding values of $(k = 20, \varepsilon)$ (middle), and $(k, \varepsilon = 0.2)$ (bottom). As expected, the average in(out)-degree and clustering coefficient decrease with larger p (as we sparsify the graph), while average distance and diameter tend to increase with larger p. Only diameter on Digg has a non-monotone behavior, but this is due to the fact that diameter is a very sensitive measure, as the removal of just one edge can change

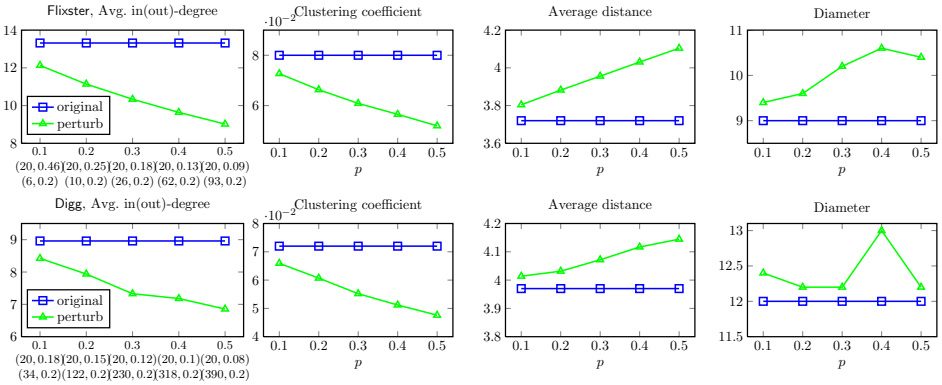

Fig. 4 Graph structure statistics for varying values of p: Flixster first row and Digg second row. For Flixster, the minimum and maximum standard deviations of the reported values over fifty random runs are $(0.0058, 0.0141)$, $(0.00038, 0.0012)$, $(0.0017, 0.0039)$ and $(0.44, 0.54)$, from left to right plots, respectively. For Digg, the minimum and maximum standard deviations of the reported values over fifty random runs are $(0.0094, 0.028)$, $(0.0002, 0.001)$, $(0.006, 0.020)$ and $(0.447, 0.836)$, from left to right plots, respectively.

substantially the maximum distance. From these plots we conclude that random perturbation (especially for smaller value of p) can preserve to a large extent the structure properties of the original graph. These findings are in line with those of [9] (who dealt with simple undirected graphs).

These statistics consider only the graph structure and ignore the topic model. Next, we evaluate the impact of random sparsification and random weight reduction on the topic-dependent influence probabilities w_e^τ. Let $\mathbf{w}_e = (w_e^1, w_e^2, ..., w_e^T)$ be the topic-dependent influence probabilities associated with thte edge e of the original network, and $\hat{\mathbf{w}}_e = (\hat{w}_e^1, \hat{w}_e^2, ..., \hat{w}_e^T)$ be the same probabilities after the perturbation. We measure the weight reduction error, which is the average distance between all the original vectorial weights and the perturbed ones, $\frac{1}{|E|} \sum_{e \in E} ||\mathbf{w}_e - \hat{\mathbf{w}}_e||_2$. If one edge e is removed during the random sparsification, then it is counted as having $\hat{w}_e^\tau = 0$ for each $1 \leq \tau \leq T$. In Figure 5 we report that weight reduction error for different values of p and $b = 600$ (left plot) and for different values of b and $p = 0.2$ (right plot), on both datasets, averaged over fifty random runs. As expected, the cost of obfuscation increases with larger p and smaller b. We can observe that the distances are generally small, especially on Digg that, as discussed before, has smaller input probabilities: in this dataset the average distance is below 0.025 even under the strongest settings of the perturbation parameters p and b.

6.3 Influence maximization queries

The last objective of our experimental evaluation is to measure the impact of random perturbation on the utility of the data for a concrete application: the *influence maximization* problem. In particular, we want to compare the difference in the result obtained for the same *Topic-aware Influence Maximization* (TIM) query, when run over the original data and on the perturbed data. We first intro-

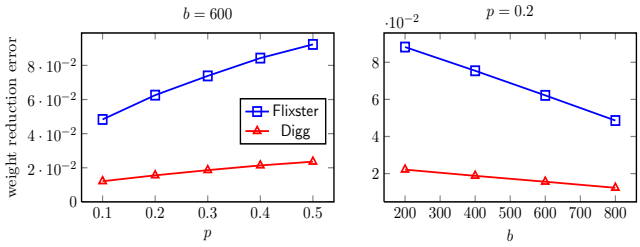

Fig. 5 weight reduction error for different values of p (left) and for different values of b (right), on both datasets. For Flixster, the minimum and maximum standard deviations of the reported values over fifty random runs is $(0.00023, 0.0006)$, respectively. For Digg, the minimum and maximum standard deviations of the reported values over fifty random runs is $(0.00015, 0.00028)$, respectively.

duce the needed background about the influence maximization problem and TIM queries, and then we present the results.

Kempe *et al.* [26] formalized the *influence maximization* problem based on the concept of a propagation model: i.e., a stochastic model that governs how users influence each other and thus how propagations happen. Given a propagation model and a set of nodes $S \subseteq V$, the expected number of nodes which are "infected" in the viral cascade that starts with S is called the *(expected) spread* of S and is denoted by $\sigma(S)$. The influence maximization problem asks for a set $S \subseteq V$, $|S| = k$, such that $\sigma(S)$ is maximal, where k is an input parameter.

The most studied propagation model is the so called *Independent Cascade* (IC) model. We are given a directed social graph $G = (V, E)$ with edges $(u, v) \in E$ labeled by influence probabilities $w_{u,v} \in (0, 1]$, representing the strength of the influence of u over v. If $(u, v) \notin E$, we define $w_{u,v} = 0$. At a given time step, each node is either active (an adopter of a product) or inactive. At time 0, a set S of seeds is activated. Time unfolds deterministically in discrete steps. As time unfolds, more and more neighbors of an inactive node v become active, until, eventually, v too becomes active. This event, in turn, may trigger further activations of nodes that are connected to v. When a node u first becomes active, say at time t, it has one chance to influence each inactive neighbor v with probability $w_{u,v}$, independently of the history thus far. If the attempt succeeds, v becomes active at time $t + 1$.

Now consider an advertisement platform allowing to implement "viral ads" over social networks, where users, by clicking on engaging ads, make them propagate to their friends. Advertisers come to the platform with a description of the ad (e.g., a set of keywords) to be promoted and they compete for the attention of the users which are considered influential w.r.t. the given description. Here we assume the *Topic-aware Independent Cascade* (TIC) propagation model introduced in [6], the parameters of which are learned from a log of past propagation traces. In particular for each edge $(u, v) \in E$ and for each topic $1 \leq \tau \leq T$ we have a probability $w^{\tau}_{u,v}$ representing the strength of influence that user u exerts over user v for topic τ.

An item x is described by a distribution $\boldsymbol{\gamma}_x = (\gamma^1_x, \ldots, \gamma^T_x)$ over the T topics, where γ^{τ}_x is the correlation of the item x with topic τ, $1 \leq \tau \leq T$, and $\sum_{\tau=1}^{T} \gamma^{\tau}_x = 1$. A propagation in the TIC model happens like in the IC model: when a node u first becomes active on item x, it has one chance of influencing each inactive neighbor v, independently of the history thus far. The tentative succeeds with a probability

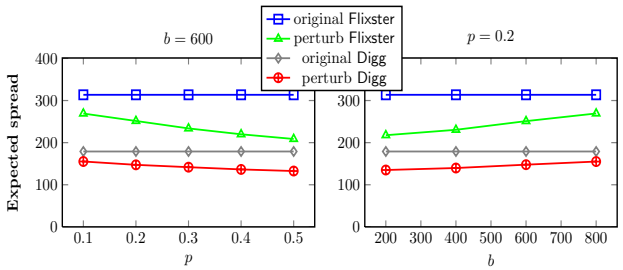

Fig. 6 Average expected spread achieved by TIM queries for varying values of p and $b = 600$ (left) and for $p = 0.2$ and varying values of b (right). For Flixster, the minimum and maximum standard deviations of the reported values over fifty random runs is $(29.23, 43.29)$, respectively. For Digg, the minimum and maximum standard deviations of the reported values over fifty random runs is $(9.68, 12.86)$, respectively.

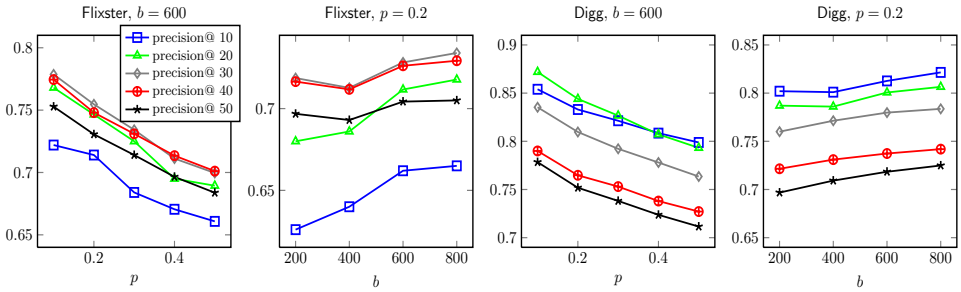

Fig. 7 Average precision@ in the seed sets extracted by TIM queries: on Flixster and Digg for varying values of p and $b = 600$ and for $p = 0.2$ and varying values of b.

that is the weighted average of the link probabilities w.r.t. the topic distribution of the item x, namely $w^x_{u,v} = \sum_{\tau=1}^{T} \gamma^\tau_x w^\tau_{u,v}$. In this context a Topic-aware Influence Maximization (TIM) query $Q(\boldsymbol{\gamma}_x, k)$, takes as input an item description $\boldsymbol{\gamma}_x$ and an integer k and it requires to find the seed set $S \subseteq V$, $|S| = k$, such that the expected number of nodes adopting item x, denoted by $\sigma(S, \boldsymbol{\gamma}_x)$, is maximum: $Q(\boldsymbol{\gamma}_x, k) = \arg\max_{S \subseteq V, |S|=k} \sigma(S, \boldsymbol{\gamma}_x)$. It is important to observe that a TIM query can always be processed by a standard influence maximization computation in the IC model. In fact, given the query item description, we can derive a directed probabilistic graph where the probability for each edge is defined as $w^x_{u,v}$ above.

Results. As discussed before, we generate the topic-dependent influence probabilities for each edge from real data. The same learning process also generates the description $\boldsymbol{\gamma}_x$ for each item x contained in the propagation log $\mathbb{D}$. From all the items we have in the database, we select 10 items at random to run the experiments. Then we run the influence maximization greedy algorithm of [26] to identify seed-sets of size $k = 50$. As a result, we measure the *expected spread* of the identified seed set, *i.e.*, the number of active nodes at the end of the influence maximization process and, the *precision*, the percentage of the overlap between the identified seed-sets in the original graph and the perturbed ones. We compute the precision for the first 10, 20, 30, 40 and 50 seeds in the seed sets extracted from the perturbed graph w.r.t. the ground-truth seed set extracted from the original

graph[3]. Due to the random nature of the perturbation, each experiment is run fifty times, and the average is reported.

In Figure 6, we report the expected spread, for different values of p and $b = 600$ (left plot) and for different values of b and $p = 0.2$ (right plot) on the two datasets. We can observe that, especially for Digg dataset, the quality of the solution produced on the perturbed graph is maintained high, very close to the quality on the original graph even under strong perturbations. In Figure 7 we report the precision of the identified seed sets. Here the quality which is maintained in the perturbed data is even more striking. For instance in Flixster, under the standard setting $p = 0.2$, $b = 600$, the solution extracted on the perturbed graph contains in average 36 of the first 50 seed nodes (influential users) that would be selected by the algorithm on the original graph. In Digg, in average 8 of the top-10 most influential users computed in the original network are extracted also from the obfuscated network, even under the strongest settings of the perturbation parameters.

7 Conclusions and future work

We studied the problem of privacy-preserving publication of topic-dependent social influence networks. Ours is the first study that considers privacy preservation in social networks of such semantic richness of data. That semantic richness poses a prominent challenge for individual privacy, because the data has to be altered significantly in order to achieve meaningful levels of privacy. We formulated the privacy notion of k-obfuscation and showed how to achieve meaningful levels of identity obfuscation while maintaining high quality data. In particular, we showed that in the specific application of topic-aware influence maximization, the random perturbation maintains very high quality in the solution.

Following this work we intend to look into the following research directions: (a) Examining the applicability of our analysis and method also for other applications in which there is a social network enriched by auxiliary data, either of the type which we considered here (directed edges and vectorial edge weights) or of other types (e.g., vectorial node weights, non-numerical labels, or hypergraphs). (b) Extending our framework to methods that are based on random deletions as well as additions of edges. (c) Exploring the possibility of achieving k-anonymity in such networks; as it is hard to achieve uniform k-anonymity without stripping the data of its utility, one may consider relaxations of that notion and assume weaker adversarial assumptions. (d) Considering a non-uniform obfuscation model in which the level of obfuscation applied depends on node properties (say, degrees, betweenness centrality, or sum of outgoing weights) or edges properties (say, weights, or characteristic of its two end points).

Acknowledgments and disclaimer

The research leading to these results has received funding from the European Unions Horizon 2020 innovation action programme under grant agreement No 653449 TYPES project.

[3] It is worth noting that, although called "set", a seed set as output by a TIM query is an *ordered list* of nodes

References

1. C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB*, pages 901–909, 2005.
2. C. Aslay, N. Barbieri, F. Bonchi, and R. Baeza-Yates. Online topic-aware influence maximization queries. In *EDBT*, 2014.
3. Ç. Aslay, W. Lu, F. Bonchi, A. Goyal, and L. V. S. Lakshmanan. Viral marketing meets social advertising: Ad allocation with minimum regret. *PVLDB*, 8(7):822–833, 2015.
4. L. Backstrom, C. Dwork, and J. M. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW*, 2007.
5. N. Barbieri and F. Bonchi. Influence maximization with viral product design. In *SDM*, 2014.
6. N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. In *ICDM*, 2012.
7. B. Bi, Y. Tian, Y. Sismanis, A. Balmin, and J. Cho. Scalable topic-specific influence analysis on microblogs. In *WSDM*, 2014.
8. P. Boldi, F. Bonchi, A. Gionis, and T. Tassa. Injecting uncertainty in graphs for identity obfuscation. In *VLDB*, 2012.
9. F. Bonchi, A. Gionis, and T. Tassa. Identity obfuscation in graphs through the information theoretic lens. In *ICDE*, 2011.
10. S. Chen, J. Fan, G. Li, J. Feng, K.-l. Tan, and J. Tang. Online topic-aware influence maximization. *Proceedings of the VLDB Endowment*, 8(6):666–677, 2015.
11. S. Chen and S. Zhou. Recursive mechanism: towards node differential privacy and unrestricted joins. In *SIGMOD*, 2013.
12. J. Cheng, A. W.-C. Fu, and J. Liu. K-isomorphism: privacy preserving network publication against structural attacks. In *SIGMOD*, 2010.
13. Y. Chu, X. Zhao, S. Liu, J. Feng, J. Yi, and H. Liu. An efficient method for topic-aware influence maximization. In *APWeb*, 2014.
14. G. Cormode, D. Srivastava, S. Bhagat, and B. Krishnamurthy. Class-based graph anonymization for social network data. *PVLDB*, 2:766–777, 2009.
15. C. Dwork. Differential privacy. In *ICALP*, 2006.
16. K. Dyagilev and E. Yom-Tov. Linguistic factors associated with propagation of political opinions in twitter. *Social Science Computer Review*, 32:195–204, 2014.
17. I. Gur. *Broker-based ad allocation in social networks*. PhD thesis, Bilkent University, 2013.
18. S. Hanhijärvi, G. Garriga, and K. Puolamaki. Randomization techniques for graphs. In *SDM*, 2009.
19. M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. In *ICDM*, 2009.
20. M. Hay, G. Miklau, and D. Jensen. Analyzing private network data. In F. Bonchi and E. Ferrari, editors, *Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques*, pages 459 – 498. Chapman & Hall/CRC, 2010.
21. M. Hay, G. Miklau, D. Jensen, D. F. Towsley, and C. Li. Resisting structural re-identification in anonymized social networks. *VLDB J.*, 19:797–823, 2010.
22. M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. *University of Massachusetts Technical Report*, 07, 2007.
23. M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *PVLDB*, 3:1021–1032, 2010.
24. V. Karwa, S. Raskhodnikova, A. D. Smith, and G. Yaroslavtsev. Private analysis of graph structure. *ACM Trans. Database Syst.*, 39:22, 2014.
25. S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith. Analyzing graphs with node differential privacy. In *TCC*, 2013.
26. D. Kempe, J. M. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.
27. M. Kivel, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of Complex Networks*, 2:203–271, 2014.
28. B. Lin and D. Kifer. Information preservation in statistical privacy and bayesian estimation of unattributed histograms. In *SIGMOD*, 2013.
29. K. Liu and E. Terzi. Towards identity anonymization on graphs. In *SIGMOD*, 2008.
30. L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *CIKM*, 2010.

31. L. Liu, J. Wang, J. Liu, and J. Zhang. Privacy preservation in social networks with sensitive edge weights. In *SDM*, 2009.
32. M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, and A. Ukkonen. Sparsification of influence networks. In *KDD*, 2011.
33. K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, 2007.
34. D. M. Romero, B. Meeder, and J. M. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *WWW*, 2011.
35. J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, 2009.
36. T. Tassa and F. Bonchi. Privacy preserving estimation of social influence. In *EDBT*, 2014.
37. J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, 2010.
38. L. Weng, J. Ratkiewicz, N. Perra, B. Gonçalves, C. Castillo, F. Bonchi, R. Schifanella, F. Menczer, and A. Flammini. The role of information diffusion in the evolution of social networks. In *KDD*, 2013.
39. X. Wu, X. Ying, K. Liu, and L. Chen. A survey of privacy-preservation of graphs and social networks. In C. C. Aggarwal and H. Wang, editors, *Managing and Mining Graph Data*, pages 421–453. Springer, 2010.
40. Q. Xiao, R. Chen, and K.-L. Tan. Differentially private network data release via structural inference. In *KDD*, 2014.
41. X. Ying and X. Wu. Randomizing social networks: A spectrum preserving approach. In *SDM*, 2008.
42. X. Ying and X. Wu. Graph generation with prescribed feature constraints. In *SDM*, 2009.
43. L. Zou, L. Chen, and M. T. Özsu. K-automorphism: A general framework for privacy preserving network publication. *PVLDB*, 2:946–957, 2009.