A Privacy Self-Assessment Framework for Online Social Networks

A Privacy Self-Assessment Framework for Online Social Networks

Ruggero G. Pensa[a,*], Gianpiero Di Blasi[a]

[a] *University of Torino – Dept. of Computer Science*
C.So Svizzera, 185 – I-10149 Torino, Italy
Tel. +39.011.670.6798 – Fax. +39.011.75.16.03

Abstract

During our digital social life, we share terabytes of information that can potentially reveal private facts and personality traits to unexpected strangers. Despite the research efforts aiming at providing efficient solutions for the anonymization of huge databases (including networked data), in online social networks the most powerful privacy protection "weapons" are the users themselves. However, most users are not aware of the risks derived by the indiscriminate disclosure of their personal data. Moreover, even when social networking platforms allow their participants to control the privacy level of every published item, adopting a correct privacy policy is often an annoying and frustrating task and many users prefer to adopt simple but extreme strategies such as "visible-to-all" (exposing themselves to the highest risk), or "hidden-to-all" (wasting the positive social and economic potential of social networking websites). In this paper we propose a theoretical framework to i) measure the privacy risk of the users and alert them whenever their privacy is compromised and ii) help the users customize semi-automatically their privacy settings by limiting the number of manual operations. By investigating the relationship between the privacy measure and privacy preferences of real Facebook users, we show the effectiveness of our framework.

Keywords: privacy measures, online social networks, active learning

1. Introduction

Social networks are one of the main traffic sources in the Internet. At the end of 2014, they attracted more than 31% of the worldwide Internet traffic towards the Web. Facebook, the most famous social networking platform, drives alone 25% of the whole traffic. As a comparison, Google search engine represents just over 37% of the global traffic. More than two billions people are estimated to be registered in at least one of the most popular social media platforms (Facebook hits the goal of one billion users in 2012). Overall, the number of active "social" accounts are more than two billions. In view of these numbers, the risks

*Corresponding author
Email addresses: ruggero.pensa@unito.it (Ruggero G. Pensa), diblasi@di.unito.it (Gianpiero Di Blasi)

due to a more and more global and unaware diffusion of our sensitive and less sensitive personal data cannot be overlooked. If, on the one hand, many users are informed about the risks linked to the disclosure of personal facts (private life events, sexual preferences, diseases, political ideas, and so on), on the other hand the awareness of being exposed to privacy breaches each time we disclose facts that are apparently less sensitive is still insufficiently widespread. A GPS tag far from home or pictures taken during a journey, may alert potential thieves who may clean out the apartment. The disclosure of family relationships may expose our own or other family members' privacy to the risks of stalking, slander and cyberbullying. Moreover, the research project myPersonality (Kosinski et al., 2013), carried out at the University of Cambridge, has shown that, by leveraging Facebook user's activity (such as "Likes" to posts or fan pages) it is possible to "guess" some very private traits of the user's personality. According to another study, it is even possible to infer some user characteristics from the attributes of users who are part of the same communities (Mislove et al., 2010). As a consequence, privacy has become a primary concern among social network analysts and Web/data scientists. Also, in recent years, many companies are realizing the necessity to consider privacy at every stage of their business. In practice, they have been turning to the principle of *Privacy by Design* (Cavoukian, 2012) by integrating privacy requirements into their business model.

Despite the huge research efforts aiming at providing efficient solutions to the anonymization of huge databases (including networked data) (Zou et al., 2009; Xue et al., 2012; Zhou & Pei, 2011; Backstrom et al., 2011), in online social networks the most powerful privacy protection is in the hands of the users: they, and only they, decide what to publish and to whom. Even though social networking sites (such as Facebook), notify their users about the risks of disclosing private information, most people are not aware of the dangers due to the indiscriminate disclosure of their personal data. Moreover, despite the fact that all social media provide some advanced tools for controlling the privacy settings of the user's profile, such tools are not user-friendly and they are barely utilized, in practice. According to Facebook former CTO Bret Taylor, most people have modified their privacy settings, but in 2012, still "13 million users [in the United States] said they had never set, or didn't know about, Facebook's privacy tools". Often the choices of many users are limited to two: i) make their own profile completely public, being exposed to all the above mentioned risks, ii) make their own profile completely private, preventing all opportunities offered by the social network sites. Some studies try to foster risk perception and awareness by "measuring" users' profile privacy according to their privacy settings (Liu & Terzi, 2010; Wang et al., 2014). These metrics usually require a *separation-based* policy configuration: in other terms, the users decide "how distant" a published item may spread in the network. Typical separation-based privacy policies for profile item/post visibility include: visible to no one, visible to friends, visible to friends of friends, public. However, this policy fails when the number of user friends becomes large. According to a well-known anthropological theory, in fact, the maximum number of people with whom one can maintain stable social (and cybersocial) relationships (known as Dunbar's number) is around 150 (Dunbar, 2016; Roberts

et al., 2009), but the average number of user friends in Facebook is more than double[1]. This means that many social links are weak (offline and online interactions with them are sporadic), and a user who sets the privacy level of an item to "visible to friends" probably is not willing to make that item visible to *all* her friends. Other studies try to make the customization process of the privacy settings less frustrating (Fang & LeFevre, 2010). However, a consensus on how to identify a trade-off between privacy protection and exploitation of social network potentials is still far from being achieved.

With the final goal of enhancing users' privacy awareness in online social networks, in this paper we propose a theoretical framework to i) measure the privacy risk of the users and alert them whenever their privacy is compromised and ii) help the exposed users customize semi-automatically their privacy level by limiting the number of manual operations thanks to an active learning approach. Moreover, instead of using a *separation-based* policy for computing the privacy risk, in this paper we adopt a *circle-based* formulation of the privacy score proposed by Liu & Terzi (2010). We assume that a user may set the visibility of each action and profile item separately for each other user in her friend list. For instance, a user u may decide to allow the access to all photo albums to friends f_1 and f_2, but not to friend f_3. In our score, the sensitivity and visibility of profile item i published by user u are computed according to the set of u's friends that are allowed to access the information provided by i. We show experimentally that our circle-based definition of privacy score better capture the real privacy leakage risk. Moreover, by investigating the relationship between the privacy measure and the privacy preferences of real Facebook users, we show that our framework may effectively support a safer and more fruitful experience in social networking sites. Differently from other research works addressing the same problem, our framework takes into account both users' preferences and the real sensitive information leakage risk in deciding how much visibility should be given to each profile item.

Our contribution can be resumed as follows:

- we define a formal framework for privacy self-assessment in online social networks based on both sensitivity and visibility of user profile items;
- we use a new privacy score leveraging more accurate *circle-based* policies;
- we present a semi-supervised machine learning approach to support the configuration of the visibility level of user profile items;
- we report the results of several experiments on original data obtained from real Facebook users.

The remainder of the paper is organized as follows: we briefly review the related literature in Section 2; the overview and the theoretical details of our framework are presented in Section 3; Section 4 provides the report of our experimental validation; finally, we draw some conclusions, discuss some limitations and propose some future research directions in Section 5.

[1] `http://www.pewresearch.org/fact-tank/2014/02/03/6-new-facts-about-facebook/`

2. Related work

With the unrestrained success of online social networks, there has been increasing research interests about privacy protection methods for individuals that participate in them. Most research efforts are devoted to the identification and formalization of privacy breaches and to the anonymization of networked data. The goal is to modify data so that the probability of identifying an individual within the network is minimized. This objective is achieved by either anonymizing only the network structure or anonymizing both network structure and user attributes (Zheleva & Getoor, 2011).

Some of the most relevant contributions tackle the problem of graph anonymization by applying edge modification (Zou et al., 2009; Liu & Terzi, 2008; Zhou & Pei, 2011), randomization (Ying & Wu, 2011; Vuokko & Terzi, 2010), generalization (Hay et al., 2008; Cormode et al., 2009) or differentially private mechanisms (Hay et al., 2009; Task & Clifton, 2012). Among the approaches that anonymize also the user attributes, Zhou & Pei (2011) adopt a greedy edge modification and label generalization algorithm, Zheleva & Getoor (2008) anonymize nodes attribute first and then tries to preserve the network structure, Campan & Truta (2009) optimize an utility function using the attribute and structural information simultaneously.

All these works focus on how to share social networks owned by companies or organizations masking the identities or the sensitive connections of the individuals involved. However, less attention has been given to the privacy risk of users caused by their information-sharing activities (e.g., posts, likes, shares). In fact, since disclosing information on the web is a voluntary activity, a common opinion is that users should care about their privacy and control it during their interaction with other social network users. Although multiple complex factors are involved in user privacy protection on social media (Litt, 2013), privacy controls for online social networking sites are not fully socially aware (Misra & Such, 2016) and are barely utilized in practice. This statement is confirmed by a study of Liu et al. (2011) which shows that 36% of Facebook content is shared with the default privacy settings and exposed to more users than expected.

Thus, another branch of research has focused on investigating measures, strategies and tools to enhance the users' privacy awareness and help them act more safely during their day-to-day social network activity. Liu & Terzi (2010) propose a framework to compute a privacy score measuring the users' potential risk caused by their participation in the network. This score takes into account the sensitivity and the visibility of the disclosed information and leverages the item response theory as theoretical basis for the mathematical formulation of the score. Instead, Motahari et al. (2010) propose an information-theoretic estimation of the user anonymity level to help predict the identity inference risks according to both external knowledge and the correlation between user attributes. Cetto et al. (2014) present an online game, called Friend Inspector, that allows Facebook users to check their knowledge of the visibility of their shared personal items and provides recommendations on how to improve privacy settings. Instead, Fang & LeFevre (2010) propose a social networking privacy wizard to help users customize their privacy settings. Similarly, Wang et al. (2015) present an interactive visualization tool that helps users configure the privacy according to

their own personality traits derived from their social media data. Squicciarini et al. (2015) propose a framework which determines the best available privacy policy for user-uploaded images on content-sharing sites according to the user's available history on the site. Becker & Chen (2009) present a tool to detect unintended information loss in online social networks by quantifying the privacy risk attributed to friend relationships in Facebook. The authors show that a majority of users' personal attributes can be inferred from social circles. Talukder et al. (2010) present a privacy protection tool that measures the inference probability of sensitive attributes from friendship links. In addition, they suggest self-sanitization actions to regulate the amount of leakage. Squicciarini et al. (2014) propose an ontology-based privacy protection mechanism supporting semi-automated generation of access rules for users' profile information. Instead, Such & Rovatsos (2016) and Such & Criado (2016) suggest a computational mechanism that is able to negotiate conflicting privacy preferences of multiple users on any individual item and merge them into a single policy. Other approaches to privacy control in social networks investigate the problem of the risk perception. Akcora et al. (2012a,b), for instance, propose to provide users with a measure of how much it might be risky to have interactions with them, in terms of disclosure of private information. They use an active learning approach to estimate user risk from few required user interactions. Finally, the impact of user privacy policies on information diffusion processes has been studied as well (Bioglio & Pensa, 2017).

The positioning of our work is in this second branch of research, but differently from the above mentioned papers, our proposal considers all aspects usually involved in social network privacy issues. In fact, we take into account the real and perceived sensitiveness of profile items, the preferences of social network users regarding the disclosure level of their activity and the position of the user within the network. In addition, to support our claims, we performed a social experiment involving real Facebook users.

3. Keeping privacy under control

In this section we introduce our theoretical framework aiming at supporting the users participating in a social network in finding a balanced tradeoff between privacy protection and visibility of the profile. We assume that the social networking platform provides all required configuration tools to set the privacy of users' actions and profile items properly. In particular, our desired property is that a user may set the visibility of each action and profile item separately for each other user in her or his friend list. For instance, a user A may decide to allow the access to all photo albums to friends B and C, but not to friend D. Most social networking platforms (such as Facebook or Google+), provide an adequate flexibility in configuring privacy of profile items and user's actions. They offer some advanced facilities, such as the possibility of grouping friends into special lists or social circles. However, using them correctly is often an annoying and frustrating task and many users prefer to adopt simple but extreme strategies such as "visible-to-all" (exposing themselves to the highest risk), or "hidden-to-all" (wasting the positive social and economic potential of social networking websites).

Furthermore, privacy is not just a matter of users' preferences; it also relies on the context

in which an individual is immersed: the attitude of her or his friends towards privacy (some users likes or share friends' posts more often than the others, thus contributing to the rapid spread of information), the position within the network (very central users are more exposed than marginal users), her or his own attitude on disclosing very private facts, and so on.

The framework we propose in this paper takes into account both aspects: i) thanks to a semi-supervised learning approach that builds a model leveraging few user's preferences, it allows to extend privacy settings to all users' friends according to this model; ii) thanks to a score that quantify the privacy leakage of each user considering both individual and contextual parameters, it provides a constant feedback on the privacy protection level of each user. Moreover, our privacy score fits the real user expectations about the visibility of profile items. Before entering the technical details of framework, we briefly introduce some basic mathematical notation required to formalize the problem.

3.1. Preliminaries and notation

Here we introduce the mathematical notation we will adopt in the rest of our paper. We consider a set of n users $\mathcal{U} = \{u_1, \ldots, u_n\}$ corresponding to the individuals participating in a social network. Each user is characterized by a set of m properties or profile items $\mathcal{P} = \{p_1, \ldots, p_m\}$, corresponding, for instance, to personal information such as gender, age, political views, religion, workplace, birthplace and so on. Hence, each user u_i is described by a vector $\boldsymbol{p}^i =< p_{i1}, \ldots, p_{im} >$.

Users are part of a social network. Without loss of generality, we assume that the link between two users is always reciprocal (if there is a link from u_j to u_j then there is also a link from u_j to u_i). Hence, the social network here is represented as an undirected graph $G(V, E)$, where V is a set of n vertices $\{v_1, \ldots, v_n\}$ such that each vertex $v_i \in V$ is the counterpart of user $u_i \in \mathcal{U}$ and E is a set of edges $E = \{(v_i, v_k)\}$. Given a pair of users $(u_i, u_k) \in \mathcal{U}$, $(v_i, v_k) \in E$ iif users u_i and u_k are connected (e.g., by a friendship link).

For any given vertex $v_i \in V$ we define the neighborhood $\mathcal{N}(v_i)$ as the set of vertices v_k directly connected to vertex v_i, i.e., $\mathcal{N}(v_i) = \{v_k \in V \mid (v_i, v_k) \in E\}$. Conversationally speaking, $\mathcal{N}(v_i)$ is the set of friends (also known as *friend-list*) of user u_i, hence we use $\mathcal{N}(v_i)$ or $\mathcal{N}(u_i)$ interchangeably. Given a user u_i and its friend-list $\mathcal{N}(u_i)$, we also define the *ego network* centred on user u_i as the graph $G_i(V_i, E_i)$, where $V_i = \mathcal{N}(v_i) \cup \{v_i\}$ and $E_i = \{(v_k, v_l) \in E \mid v_k, v_l \in V_i\}$.

Finally, for any user u_i we introduce a *privacy policy matrix* $\boldsymbol{M}_i \in \{0, 1\}^{|\mathcal{N}(u_i)| \times m}$ defined as follows: for any element m^i_{kj} of $\boldsymbol{M}_i$, $m^i_{kj} = 1$ iif profile item $p_j \in \mathcal{P}$ is visible to user $u_k \in \mathcal{N}(u_i)$ (0 otherwise, i.e., iif user u_k is not allowed to access profile item p_j).

It is worth noting that our framework can be easily extended to the case of directed social networks (such as Twitter): in this case, the privacy policies are defined only on inbound links.

3.2. General framework

Let us now introduce the technical details of our framework that allows the users to actively control their own privacy leakage. The framework consists of two distinct core parts: i) a score $\phi_p(u_i)$ that measures the privacy leakage of each user u_i and ii) a set of models

$\{\mu_p(u_i)\}$ of privacy preferences, one for each user u_i. In a nutshell, the framework is based on a routine (see Algorithm 1) that: i) computes the privacy policy matrix $\boldsymbol{M}_i$ according to the privacy preference model $\mu_p(u_i)$ of each user u_i; ii) computes the privacy score $\phi_p(u_i)$ of all users; iii) notify each user u_i whose privacy score $\phi_p(u_i)$ exceeds a given threshold τ. Even if there hasn't been any change in the privacy policies, the routine should be executed periodically, since other types of changes may have occurred in the social network (e.g., creation or removal of vertices/links in G, voluntary changes in the privacy policy by any user, and so on).

Algorithm 1: *GenericPrivacyCheckRoutine*$(\{\mu_p(u_i)\}, \tau)$: $\{\mu_p(u_i)\}$ is the set of models of users' preferences and τ is a privacy leakage threshold

```
forall u_i ∈ 𝒰 do
    use the preference model μ_p(u_i) to compute the policy matrix M_i;
end
forall u_i ∈ 𝒰 do
    compute the privacy score φ_p(u_i);
    if φ_p(u_i) > τ then
        notify user u_i;
    end
end
```

In the following, we will provide more details on the key aspects of Algorithm 1: how to compute the privacy score $\phi_p(u_i)$ and the preference model $\mu_p(u_i)$. Before entering the computational details, we describe here the desired intuitive properties of $\phi_p(u_i)$ and $\mu_p(u_i)$.

- **Desired properties of** $\phi_p(u_i)$: The privacy score should satisfy the following properties: i) the higher the sensitivity of the disclosed information, the higher the value of the score; ii) the higher the visibility of the disclosed information within the network, the higher the value of the score.
- **Desired properties of** $\mu_p(u_i)$: The model describing users' privacy preferences should meet the following intuitive requirements: i) since deciding the access level of any profile item for any individual friend is a long and frustrating task, $\mu_p(u_i)$ should minimize the user's intervention; ii) despite this, the model should be as accurate as possible in predicting those privacy preferences not explicitly set by the users; iii) the model should by easily updatable when the user sets more privacy preferences or add new friends.

3.3. Privacy score

In our framework, the privacy score is inspired by the naive privacy score defined by Liu & Terzi (2010). It measures the user's potential risk caused by his or her participation in the network. A $n \times m$ response matrix $\boldsymbol{R}$ is associated to the set of n users $\mathcal{U}$ and the set of

m profile properties $\mathcal{P}$[2]. Each element r_{ij} of $\boldsymbol{R}$ contains a privacy level that determines the willingness of user u_i to disclose information associated with property p_j. In the binomial case $r_{ij} \in \{0, 1\}$: $r_{ij} = 1$ (resp. $r_{ij} = 0$) means that user u_i has made the information associated with profile item p_j publicly available (resp. private). Here we adopt the multinomial case, where entries in $\boldsymbol{R}$ take any non-negative integer values in $\{0, 1, \ldots, \ell\}$, where $r_{ij} = h$ (with $h \in \{0, 1, \ldots, \ell\}$) means that user u_i discloses information related to item p_j to users that are at most h links away in the social network G (e.g., if $r_{ij} = 0$ user u_i wants to keep p_j private, if $r_{ij} = 1$ user u_i is willing to make p_j available to all friends, if $r_{ij} = 2$ user u_i is willing to make p_j available to the friends of her or his friends, and so on). For this reason, we call this policy *separation-based*. However, in this work, we use a *circle-based* definition of privacy score, first introduced by Pensa & di Blasi (2016). A different meaning for the entries r_{ij} of $\boldsymbol{R}$ is adopted: in our framework r_{ij} is directly proportional to the number of friends to whom u_i is willing to disclose the information of profile property p_j. Hence, we can compute $\boldsymbol{R}$ according to the *circle-based* privacy policies defined by matrices $\boldsymbol{M}_i$'s using this formula:

$$r_{ij} = \left\lfloor \ell \cdot \frac{1}{|\mathcal{N}(u_i)|} \sum_{k=1}^{|\mathcal{N}(u_i)|} m_{kj}^i \right\rfloor \tag{1}$$

where $\mathcal{N}(u_i)$ is the set of friends of user u_i, m_{kj}^i denotes the visibility of user u_i's profile item p_j for friend u_k, and $\lfloor \cdot \rfloor$ is the floor function. As a consequence, $r_{ij} = \ell$ iif $\forall u_k \in \mathcal{N}(u_i)$, $m_{kj}^i = 1$. Our definition is conceptually different from the original one, since the latter does not take into account the possibility of disclosing personal items to just a part of friends.

In the following, we use $\boldsymbol{R}^S$ when we refer to the response matrix computed with the original separation-based policy approach defined by Liu & Terzi (2010). We use $\boldsymbol{R}^C$ when we refer to our circle-based definition of response matrix.

Using the response matrix, it is possible to compute the two main components of the privacy score: the sensitivity β_j of a profile item p_j, and the visibility V_{ij} of a profile item p_j due to u_i. The sensitivity of a profile item p_j depends on the item itself (attribute "sexual preferences" is usually considered more sensitive than "age"). The visibility, instead, captures to what extent information about profile item p_j of user u_i spreads in the network. Liu & Terzi (2010) use a mathematical model based on item response theory (a well known theory in psychometrics) to compute sensitivity and visibility. However, we adopt the naive but still effective formulation that, additionally, is more efficient from the computational point of view.

In this framework, for $h = \{1, \ldots, \ell - 1\}$ sensitivity is computed as follows:

$$\beta_{jh} = \frac{1}{2} \left(\frac{n - \sum_{i=1}^{n} \mathbb{1}_{(r_{ij} \geq h)}}{n} + \frac{n - \sum_{i=1}^{n} \mathbb{1}_{(r_{ij} \geq h+1)}}{n} \right) \tag{2}$$

where $\mathbb{1}_A$ is the indicator function that returns 1 when condition A is true (0 otherwise).

[2]In this work, we refer to $\mathcal{P}$ as a fixed set of profile properties or user actions. It is out of the scope of this paper to consider posted items individually. We address this point in the conclusions (see Section 5).

Table 1: Example of input dataset for the classification task

Friend ID	**Age**	**Gender**	**Hometown**	**Community**	**No. of friends**	C_{work}	C_{photos}	$C_{politics}$
102030	"21-30"	Male	Montreal	C10	"501-700"	allow	allow	deny
203040	"31-40"	Female	New York	C5	"201-300"	allow	deny	deny
304050	"15-19"	Female	Vancouver	C7	"101-200"	allow	deny	deny
405060	"41-50"	Female	Seattle	C5	"701-1000"	allow	deny	deny
506070	"51-60"	Male	Montreal	C10	"501-700"	allow	allow	deny
607080	"21-30"	Female	Montreal	C10	"301-500"	?	?	?
708090	"41-50"	Male	New York	C5	"301-500"	?	?	?

When $h = 0$ or $h = \ell$, the sensitivity values are respectively

$$\beta_{j0} = \frac{n - \sum_{i=1}^{n} \mathbb{1}_{(r_{ij} \geq 1)}}{n} \tag{3}$$

and

$$\beta_{j\ell} = \frac{n - \sum_{i=1}^{n} \mathbb{1}_{(r_{ij} \geq \ell)}}{n} \tag{4}$$

The meaning of Equations 2, 3 and 4 is the following: the more users adopt at least privacy level h for privacy item p_j, the less sensitive p_j is w.r.t. level h. Moreover, for intermediate values of h ($h = \{1, \ldots, \ell - 1\}$), the sensitivity values takes into account both level h and $h + 1$. This guarantees that $\beta_{j0} < \beta_{j1} < \ldots < \beta_{j\ell}$ (Liu & Terzi, 2010).

The visibility, for $h = \{0, \ldots, \ell\}$ is computed as follows:

$$V_{ijh} = Pr(r_{ij} = h) \cdot h \tag{5}$$

where $Pr(r_{ij} = h)$ is the probability that r_{ij} is equal to h. By assuming independence between profile properties and users, this probability can be computed as follows:

$$Pr(r_{ij} = h) = \frac{\sum_{i=1}^{n} \mathbb{1}_{(r_{ij}=h)}}{n} \cdot \frac{\sum_{j=1}^{m} \mathbb{1}_{(r_{ij}=h)}}{m} \tag{6}$$

Intuitively, visibility V_{ijh} is higher when the sensitivity of profile items is low and when users have the tendency to disclose lots of their profile items (Liu & Terzi, 2010). An alternative formulation of V_{ijh} is given by the following formula:

$$V_{ijh} = Pr(r_{ij} = h) \cdot f_j^i(h) \tag{7}$$

where $f_j^i(h)$ is the fraction of users in the network G that know the value of profile item p_j for user u_i, given that $r_{ij} = h$. It depends on the position of user u_i within the network and can be computed by exploiting any information propagation models (Kempe et al., 2003).

The normalized privacy score $\overline{\phi}_p(u_i, p_j)$ for any user u_i and profile property p_j is computed as follows:

$$\overline{\phi}_p(u_i, p_j) = \frac{\phi_p(u_i, p_j)}{\max\limits_{u_k \in \mathcal{U}} \phi_p(u_k, p_j)} \tag{8}$$

where

$$\phi_p(u_i, p_j) = \sum_{h=0}^{\ell} \beta_{jh} \cdot V_{ijh} \tag{9}$$

and $\max_{u_k \in \mathcal{U}} \phi_p(u_k, p_j)$ is the maximum value of Equation 9 among all users. Normalization is not strictly required, but it unifies the scale of the privacy scores and make the choice of a suitable threshold easier.

Finally, the overall privacy score $\phi_p(u_i)$ for any user u_i is given by

$$\phi_p(u_i) = \sum_{j=1}^{m} \overline{\phi}_p(u_i, p_j). \tag{10}$$

From Equation 8, 9 and 10 it is clear that users that have the tendency to disclose sensitive profile properties to a wide public are more prone to privacy leakage. Intuitively, $\phi_p(u_i) = 0$ means that, in each element of the summation, either $\beta_{jh} = 0$ (the profile item p_j is not sensitive at all), or $V_{ijh} = 0$ (the profile item p_j is kept private). On the contrary, the privacy score is maximum when a user discloses to all her or his friends ($V_{ijh} = 1$) all sensitive information ($\beta_{jh} = 1$).

In this paper, we use ϕ_p^S when we refer to the score computed using the original separation-based response matrix $\boldsymbol{R}^S$; we use ϕ_p^C when we refer to the privacy score leveraging our circle-based definition of response matrix $\boldsymbol{R}^C$.

Notice that our definition of privacy score requires the availability of visibility preferences for all user friends. It is worth noting that most social media platforms allow the users to define friends groups or circles and set privacy preferences for groups/circles instead of requiring them to set preferences for every individual friends. However, in the next section, we will see that our framework is designed to minimize the user's intervention while computing the circle-based privacy policy matrices $\boldsymbol{M}_i$.

3.4. User preference model

The second key part of our framework is the user preferences model $\mu_p(u_i)$. The classification model should be as accurate as possible in predicting those privacy preferences not explicitly set by the users. Moreover, the model should be easily updatable when the user sets more privacy preferences or adds new users. Our choice is to use a Naive Bayes classifier (Mitchell, 1997), since it has the desirable properties we enumerated in Section 3.2. In fact, Naive Bayes classifiers are simple and converge quickly even with few training data. Moreover, they can be easily embedded in an active learning framework using, for instance, uncertainty sampling (Dagan & Engelson, 1995) thus minimizing the intervention of the user in the model training phase.

Our privacy preference model for any given user $u_i \in \mathcal{U}$ and any given profile item $p_j \in \mathcal{P}$ is then a classification problem in which we have a set of $|\mathcal{N}(u_i)|$ instances $D = \{d_1, \ldots, d_{|\mathcal{N}(u_i)|}\}$ corresponding to all friends of u_i. Each instance d_k is characterized by a set of p attributes $\{A_1, \ldots, A_p\}$ with discrete values and m class variables

$\{C_1, \ldots, C_m\}$ that take values in the domain $\{allow, deny\}$: $C_j = allow$ (resp. $C_j = deny$) means that friend u_k is allowed (resp. is not allowed) to access the information of profile item p_j of user u_i. The values of attributes $\{A_1, \ldots, A_p\}$ are partly derived from the profile vector $\boldsymbol{p}^k =< p_{k1}, \ldots, p_{km} >$ of users u_k, partly from the ego network $G_i(V_i, E_i)$ of user u_i (see Section 3.1). For instance, they may contain information such as the workplace and home-town of u_k, or the communities in G_i u_k belong to. Table 1 is an example of possible small dataset for a generic user consisting of five training instances and two test instances with three profile-based attributes, two network-based attributes and three class variables.

The Naive Bayes classification task can be regarded as estimating the class posterior probabilities given a test example d_k, i.e., $Pr(C_j = allow|d_k)$ and $Pr(C_j = deny|d_k)$. The class with the highest probability is assigned to the example d_k. Given a test example d_k, the observed attribute values are given by the vector $\boldsymbol{d}^k = \{a_1^k, \ldots, \ldots, a_p^k\}$, where a_s^k is a possible value of A_s, $s = 1, \ldots, p$. The prediction is the class c ($c \in \{allow, deny\}$) such that $Pr(C_j = c|A_1 = a_1^k, \ldots, A_p = a_p^k)$ is maximal. By Bayes' theorem, the above quantity can be expressed as

$$\begin{aligned} & Pr(C_j = c|A_1 = a_1^k, \ldots, A_p = a_p^k) = \\ & = \frac{Pr(A_1 = a_1^k, \ldots, A_p = a_p^k|C_j = c)Pr(C_j = c)}{Pr(A_1 = a_1^k, \ldots, A_p = a_p^k)} = \\ & = \frac{Pr(A_1 = a_1^k, \ldots, A_p = a_p^k|C_j = c)Pr(C_j = c)}{\sum\limits_{c_x} Pr(A_1 = a_1^k, \ldots, A_p = a_p^k|C_j = c_x)Pr(C_j = c_x)} \end{aligned} \tag{11}$$

where, $Pr(C_j = c)$ is the class prior probability of c, which can be estimated from the training data. If we assume that conditional independence holds, i.e., all attributes are conditionally independent given the class $C_j = c$, then

$$\begin{aligned} & Pr(A_1 = a_1^k, \ldots, A_p = a_p^k|C_j = c) = \\ & = \prod_{s=1}^{p} Pr(A_s = a_s^k|C_j = c) \end{aligned} \tag{12}$$

and, finally

$$\begin{aligned} & Pr(C_j = c|A_1 = a_1^k, \ldots, A_p = a_p^k) = \\ & = \frac{Pr(C_j = c)\prod_{s=1}^{p} Pr(A_s = a_s^k|C_j = c)}{\sum\limits_{c_x} Pr(C_j = c_x)\prod_{s=1}^{p} Pr(A_s = a_s^k|C_j = c_x)} \end{aligned} \tag{13}$$

Thus, given a test instance d_k, its most probable class is given by:

$$c = \arg\max_{c_x} \left\{ Pr(C_j = c_x) \prod_{s=1}^{p} Pr(A_s = a_s^k|C_j = c_x) \right\} \tag{14}$$

where the prior probabilities $Pr(C_j = c_x)$ and the conditional probabilities $Pr(A_s = a_s^k | C_j = c_x)$ are estimated from the training data.

Hence, our preference model is given by $\mu_p(u_i) = (\Psi^p, \Psi^c)$, where Ψ^p is the set of all prior probabilities $Pr(C_j = c_x)$ and Ψ^c is the set of all conditional probabilities $Pr(A_s = a_s^k | C_j = c_x)$ computed on the set of training instances from D, i.e., on a set of users from $\mathcal{N}(u_i)$ for which u_i has given an allow/deny label explicitly. Now, the key question is: how to predict all C_j's accurately without requesting too much labeling work to u_i?

To solve this problem, we adopt an *active learning* approach named *uncertainty sampling* (Lewis & Gale, 1994) based on the *maximum entropy* principle (Dagan & Engelson, 1995). In an active learning settings the learning algorithm is able to interactively ask the user for the desired/correct labels of unlabeled data instances. A way to reduce the amount of labeling queries to the users is to sample only those data instances whose predicted class is the most uncertain. Different measures of uncertainty have been proposed in the literature, e.g., least confidence (Culotta & McCallum, 2005), smallest margin (Scheffer et al., 2001) and maximum entropy (Dagan & Engelson, 1995), but for binary classification tasks they are equivalent. Hence, we decided to adopt the maximum entropy principle. According to this principle, the most uncertain data instance d_u is given by:

$$d_u = \arg\max_{d_k} \left\{ -\sum_{c_x} Pr(C_j = c_x | d_k) \log Pr(C_j = c_x | d_k) \right\} \tag{15}$$

Since probabilities $Pr(C_j = c_x | d_k)$ are exactly those computed by the Naive Bayes classifier to take its decision, this principle can be easily adapted to our preference model.

Once all friends' labels are predicted, each entry of the policy matrix $\boldsymbol{M}_i$ can be updated as follows:

$$\forall u_k \in \mathcal{N}(u_i),\ m_{kj}^i = \begin{cases} 1, & \text{if } C_j = \textit{allow} \text{ for } u_k \\ 0, & \text{if } C_j = \textit{deny} \text{ for } u_k. \end{cases} \tag{16}$$

3.5. Privacy check routine

According to the choices that we detailed in the previous sections, we can now provide a more detailed instance of Algorithm 1. The final routine for privacy control is described by Algorithm 2. The first step is the construction of the dataset required by the Naive Bayes classifier, followed by the initialization of the privacy policy matrices $\boldsymbol{M}_i$. This initialization step can be performed in several ways: randomly, following a common criterium, using Naive Bayes on a first seed of labeled friends. Then, using matrices $\boldsymbol{M}_i$, the routine computes the response matrix $\boldsymbol{R}^C$ and the initial privacy scores $\phi_p(u_i)$.

The core part of the routine checks whether the privacy score of any user u_i exceeds a given threshold τ. If it is the case, the routine notifies user u_i. Once notified, user u_i has the possibility to enhance her privacy settings by both redefining their friends groups/circles or by trying to update her privacy settings with the active learning procedure described by Algorithm 3. This procedure selects the K most uncertain friends and asks u_i for their

Algorithm 2: $PrivacyCheckRoutine(\boldsymbol{P}, \{\mu_p(u_i)\}, G, \tau)$: $\boldsymbol{P}$ is the users' profile matrix, $\{\mu_p(u_i)\}$ is the set of models of users' preferences, G is the social graph and τ is a privacy leakage threshold.

forall $u_i \in \mathcal{U}$ **do**
  forall $u_k \in \mathcal{N}(u_i)$ **do**
    build $\boldsymbol{d}^k = \{a_1^k, \ldots, \ldots, a_p^k\}$ from $\boldsymbol{p}^k =< p_{k1}, \ldots, p_{km} >$ and $G_i(V_i, E_i)$;
  end
  compute the preference model $\mu_p(u_i)$ and the privacy policy matrix $\boldsymbol{M}_i$ using (14) and (16);
end
forall $u_i \in \mathcal{U}$ **do**
  forall $p_j \in \mathcal{P}$ **do**
    compute r_{ij} using (1);
  end
  compute the privacy score $\phi_p(u_i)$ using (10);
  if $\phi_p(u_i) > \tau$ **then**
    notify user u_i;
  end
end

Algorithm 3: $UpdatePreferences(u_i, \mu_p(u_i), K)$: u_i is the user, $\mu_p(u_i)$ is the model of user u_i's preferences and K is a positive integer.

ask user u_i for the labels of the K most uncertain friends according to (15);
update the preference model $\mu_p(u_i)$ and the privacy policy matrix $\boldsymbol{M}_i$ using (14) and (16);

labels. Afterwards, it launches the Naive Bayes classifier and reassign the new $\{allow, deny\}$ labels to all unlabeled friends. Matrix $\boldsymbol{M}_i$ is then updated accordingly.

3.5.1. Relabeling based on privacy score

So far, we have only considered users' relabeling as a result of the uncertain predictions based on the users' preference model. However, one may force the framework to be more protective w.r.t. users' privacy settings by leveraging the privacy score itself. Our assumption is that, if a user has an unsafe behavior w.r.t. his/her own privacy settings, then she/he is more prone to share posts and facts published by his/her friends. For this reason, when predicting the deny/allow labels for the unlabeled friends of a user u_i, we add a further control step in which we automatically set to *deny* all privacy settings related to profile properties p_j and friends u_k such as

$$\overline{\phi}_p(u_k, p_j) > \tau_\phi \tag{17}$$

where $\overline{\phi}_p(u_k, p_j)$ is the privacy score of friend u_k w.r.t. p_j and τ_ϕ is a user defined threshold. Of course, this control is performed only on predicted labels, i.e., those privacy settings for which the users have not expressed their preferences yet. In the remainder of the paper, we will refer to this particular setting as *strict framework*.

3.5.2. Theoretical complexity

We now analyze the theoretical computational complexity of our algorithm. Let n be the number of total users in the social network, f the average number of users' friends, p the number of attributes of the dataset D and m the number of profile items. The initialization step requires $O(n \cdot f \cdot p)$ operations for building the dataset D, and $O(n \cdot m)$ operation for computing the response matrix $\boldsymbol{R}^C$. Obtaining the privacy score for all users requires $O(n \cdot m \cdot \ell)$ operations for computing the sensitivity values β_{jh} (ℓ being the number of privacy levels in $\boldsymbol{R}^C$), the same cost for computing the visibility values V_{ijh} and for computing the final value of the score. Under the reasonable assumptions that $m << f$ and $\ell << p$, the overall complexity for computing all the privacy scores is then $O(n \cdot f \cdot p)$.

In the worst case (when all privacy scores are above the threshold) the core part of the routine needs to train a Naive Bayes classifier for all users and profile items (Algorithm 3). Since training a Naive Bayes classifier requires $O(f \cdot p)$ operations, the complexity of this part is $O(n \cdot m \cdot f \cdot p)$.

As a conclusion, the combination of Algorithm 2 and Algorithm 3 is linear in all terms. However, in standard applications, we can assume that $f << n$ (the number of users in a social network is much greater than the average number of users' friends). Also, it is straightforward to suppose that $\ell << m \sim p << f$. Following these reasonable assumptions, n prevails on all other terms and the overall complexity of a single execution of our routine is $O(n)$. Moreover, most operations (i.e., training the classifiers, computing individual privacy scores, selecting of the most uncertain friends) can be executed concurrently. A single check/update operation on all users is then highly scalable and the overhead for a system implementing our framework is reasonably low.

4. Experimental results

In this section we report and discuss the results of an online experiment that we conducted on real Facebook users. The main objectives of our online experiment were:

- to build an original and large enough dataset centered on privacy-related issues in social network data;
- to gather a significant number of correct privacy labels for a small set of relevant and differently sensitive items/user actions;
- to make people concern about their privacy in social networks.

As regards this specific work, the data we gathered should allow us to draw scientifically justified conclusions about:

- the relationship between the separation-based privacy policies and our circle-based policy definition;
- relationship between the separation-based privacy score ϕ_p^S defined by Liu & Terzi (2010) and our circle-based score ϕ_p^C;
- the relationship between users' attitude towards privacy self-protection in Facebook and the value of the privacy score;
- the trend of the privacy score value as a function of the amount of labeled friends;
- the impact of the threshold on the number of notified users;
- the reliability of the privacy score;
- the adoption of the additional criterion based on friends' privacy scores;
- the scalability of the application w.r.t. the number of users and CPU's.

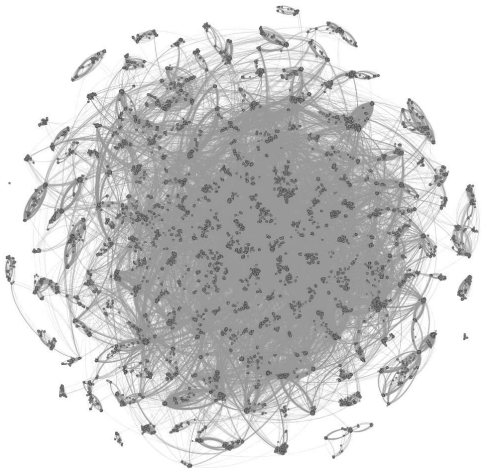
The section is organized as follows: first, we describe the data and how we gathered them; then we provide the details of our experimental settings; finally we report the results and discuss them.

4.1. Dataset

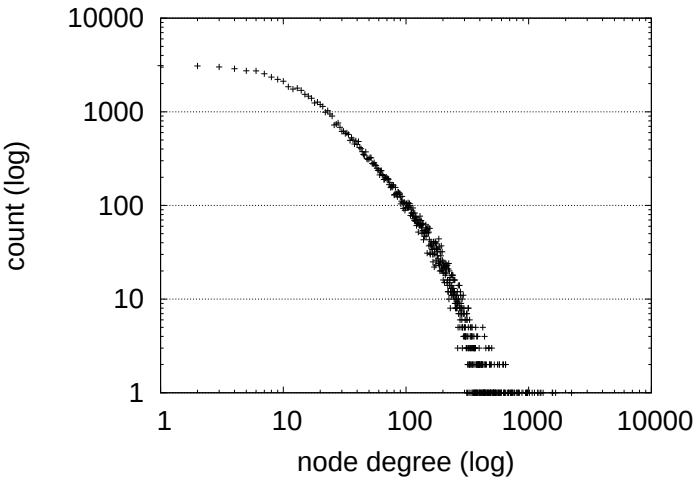
Our online experiments were conducted in two phases. In the first phase we promoted the web page of the experiment[3] where people could voluntarily grant us access to some data related to their own Facebook profile and friends' network. We were not able to access any other information rather than what we asked the permission for, i.e.: email (needed to contact the users for the second phase of our experiment), public profile, friend list, gender, age, work, education, hometown, current location and pagelikes. The participants were perfectly aware about the data we asked for and the purpose of our experiment. In this first phase, data were gathered through a Facebook application developed in Java JDK 8, using Version 1.0 of Facebook Graph API. From March to April 2015, we collected the data of 185 volunteers, principally from Europe, Asia and Americas. The social network consisting of all participants plus their friends is an undirected graph with 75,193 nodes and 1,377,672 edges. Although the overall social graph has been generated from participants' ego networks, the largest connected component consists of 73,050 nodes (97.15% of the overall network) and 1,333,276 edges (96.78% of the overall network). This goal was achieved by allowing the Facebook application to publish on the participant's timeline a special post inviting all her friend to join the experiment. Some statistics (number of nodes and edges, average degree, average clustering coefficient) about the dataset (as computed by Gephi[4]) are reported in Figure 1(c), while Figure 1 present a picture of the network and its degree distribution. All graphs are considered as undirected.

[3] `http://kdd.di.unito.it/privacyawareness/`

[4] `https://gephi.org/`



(a) Facebook network



(b) Degree distribution

#nodes	#edges	Avg. deg.	Avg. CC	#connected comp.	#communities
75,193	1,377,672	36.644	0.613	7	92

(c) Some dataset characteristics

Figure 1: Facebook network, its degree distribution and some characteristics

During the second phase, all the remaining participants were contacted for the interactive part of our experiment. First, the participants had to indicate to which level (0=no one, 1=close friends, 2=friends except acquaintances, 3=all friends, 4=friends of friends, 5=everyone on Facebook) they were willing to allow the access to five personal profile topics. The topics were proposed in form of direct questions (see Table 2) with different levels of sensitivity. We used the answers to fill the response matrix $\boldsymbol{R}^S$. Then, to each participant, we proposed a list of 60 randomly chosen friends and 6 randomly chosen friends of friends (when available). The participants had to indicate to which people they were willing to allow the access to the same five topics. For this phase, we developed a Java JDK 8 mobile-friendly web application leveraging Version 2.0 of Facebook Graph API. We used the answers on friends to fill the response matrix $\boldsymbol{R}^C$. From May 2015 to February 2016, 101 participants out of 185 replied to the first part of the survey, 111 to the second part and 74 out of 185 participants answered all questions of two surveys. Hence, we consider the network data provided by all 185 participants and the survey data related to the 74 participants who completed the two parts of the questionnaire. In Figure 2 we report some statistics describing the 74 participants. All the data have been anonymized to preserve volunteers' privacy[5]. The entries in the two resulting 74×5 matrices $\boldsymbol{R}^S$ and $\boldsymbol{R}^C$ take values in $\{0, \ldots, 5\}$.

[5]The data collection/storage and processing protocols have been approved by the Law Office of our institution.

Table 2: The five questions of our online survey

Q1	Which people would you like to tell that you have just changed job?
Q2	If your relationship status changed, which friends would you like to tell?
Q3	After a nice holiday, which friends would you share your photos with?
Q4	With whom would you like to share a comment on current affairs/politics?
Q5	With whom would you like to share your mood or something personal that happened to you?

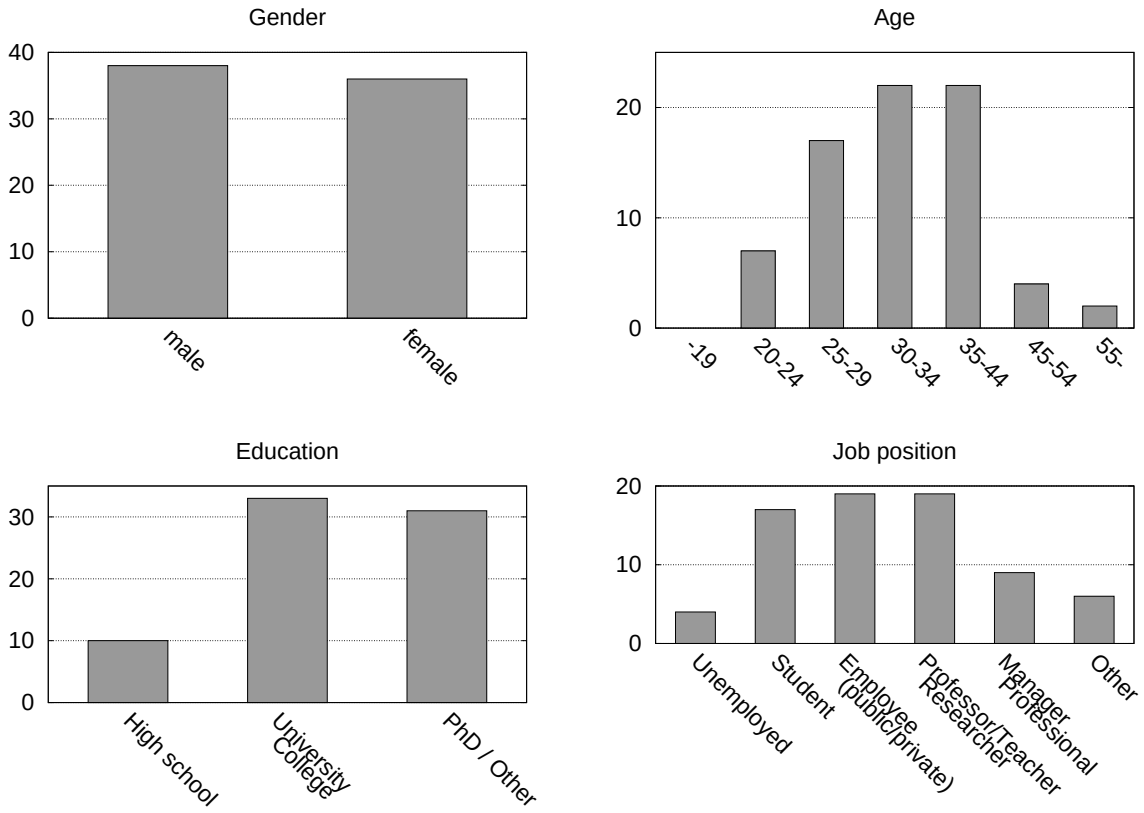


Figure 2: Personal data statistics of the individuals that participated in our online experiment

4.2. Separation-based vs. circle-based policies

As a preliminary analysis, we measure how the perception of topic sensitivity changes when the two policies (separation-based and circle-based) are presented to the participants. We compare the two response matrix $\boldsymbol{R}^S$ and $\boldsymbol{R}^C$ in several ways. First, we measure the

Pearson's correlation coefficient between the two matrices. Given two series of n values $X = x_1 \ldots, x_n$ and $Y = y_i, \ldots, y_n$, the Pearson's coefficient is computed as:

$$\rho(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{18}$$

where $\overline{x} = \sum_{i=1}^{n} x_i/n$ and $\overline{y} = \sum_{i=1}^{n} y_i/n$. It basically captures the correlation between the two series of values and ranges between -1 (for inversely correlated sets of values) and $+1$ (for the maximum positive correlation). In our experiment, $n = 74 \cdot 5$. We obtain a moderate positive correlation ($\rho(\boldsymbol{R}^S, \boldsymbol{R}^C) = 0.4632$), that indicates a substantial difference between the two policies. Then, for each question Q_j, we measure the average difference between each entry of the two matrices as $\sum_i (r_{ij}^S - r_{ij}^C)/n$. All the average differences are positive, i.e., the given separation-based policies are less restrictive than circle-based ones. In particular, we measure an average difference of 0.54 for Q_1, 0.43 for Q_2, 0.32 for Q_3, 0.35 for Q_4 and 0.15 for Q_5. Moreover, we measure the overall sensitivity of each topic as $\beta_j = \sum_h^{\ell} \beta_{jh}$ (see Section 3.3) in the two cases. As can be seen in Figure 3(a), all sensitivity values increase when the circle-based policy is adopted. The improved sensitivity perception is confirmed when we look at the users' policies more deeply. In particular, for each question Q_j, we count:

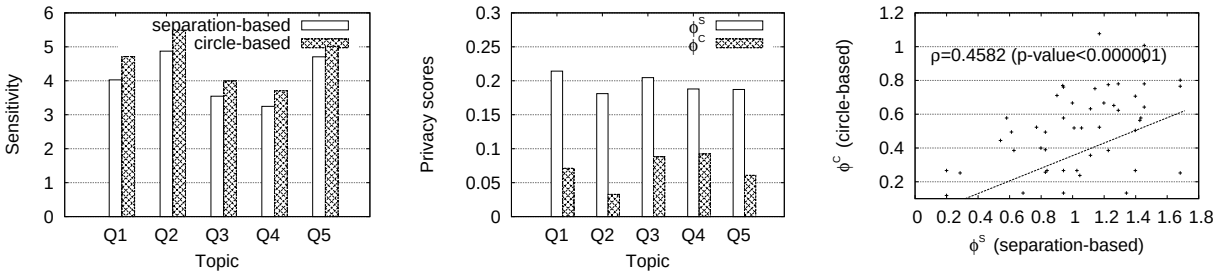
- the number **A** of participants that, in the separation-based test, have made Q_j at least visible to friends of their friends ($r_{ij}^S \geq 4$), but have denied the access to Q_j to some of the friends of their friends in the circle-based test;
- the number **B** of users that have granted the access to some of the friends of their friends in the circle-based test while $r_{ij}^S < 4$ in the separation-based test;
- the number **C** of participants that, in the separation-based test, have made Q_j visible at least to all friends ($r_{ij}^S \geq 4$), but have denied the access to Q_j to some of their friends in the circle-based test $r_{ij}^C < 5$;
- the number **D** of participants that, in the circle-based test, have made Q_j visible to all friends ($r_{ij}^C = 5$), but have denied the access to Q_j to some of their friends in the separation-based test $r_{ij}^S < 3$.

The results in Table 3 indicate that the major differences are on questions Q_3 and Q_4, that are the less sensitive according to Figure 3(a). However, then passing from a separation-based policy to a circle-based one, many users have reviewed their choices in a more restrictive way for question Q_1 and Q_2 as well.

Finally, we also compute the privacy scores $\phi_p^S(u_i, p_j)$ and $\phi_p^C(u_i, p_j)$ for each question Q_j and each participant u_i. The average score values are given in Figure 3(b). Interestingly, although the circle-based policy increases the perception of topic sensitivity, the related privacy scores are sensibly smaller than those computed within the separation-based hypothesis, i.e., the participants have a safer behavior w.r.t. the visibility of the topics. For

Table 3: Policy differences in visibility

Measure	Q1	Q2	Q3	Q4	Q5
A	2	2	4	9	1
B	0	0	4	9	1
C	20	5	19	21	4
D	0	0	4	9	1



(a) Sensitivity (b) Privacy score (c) Correlation analysis

Figure 3: Comparative results (separation-based approach vs. circle-based approach)

the sake of completeness, we perform a correlation analysis between the values of $\phi_p^S(u_i)$ and $\phi_p^C(u_i)$ in Figure 3(c). The value of the Pearson's ρ coefficient (0.4582) shows moderate positive correlation between the two series of scores.

4.2.1. User's preferences vs. privacy score

To measure the performances of the active learning approach, we generate 74×5 datasets (one for each pair of users and questions) that we use to train and test the Naive Bayes classifier. These datasets contain, for each friend u_k of a user u_i, the following attributes: *gender* and *age* of u_k, *countryman* (true, if u_k and u_i were born in the same place, *fellow_citizen* (true, if u_k and u_i live in the same place), *coworker* (true, if u_k and u_i work or have worked in the same place), *schoolmate* (true, if u_k and u_i are or have studied in the same school/college/university), and the *Jaccard similarity of page likes* of u_i and u_k. All attribute values are derived from the information extracted by the Facebook profiles, when available. Additionally, we also consider the *list of communities* u_k is part of. To this purpose, we execute a community detection algorithm on the so called "ego-minus-ego" networks (the subgraph induced by the vertex set $\mathcal{N}(u_i) \setminus \{u_i\}$) of all 74 users. We use *DEMON* (Coscia et al., 2014), a local-first approach based on a label propagation algorithm that is able to discover overlapping communities. The algorithm requires two parameters as input: the minimum accepted size for a community ($minCommunitySize$) and a parameter ϵ that determines the minimum overlap two communities should have in order to be merged. In our experiments, we set $minCommunitySize = 3$ (to discard very small communities) and $\epsilon = 0.5$ (to admit an average overlap degree). Finally, each friend has a class variable that takes values in the set $\{allow, deny\}$.

In a first experiment, we study the relationship between the accuracy of the predicted

user's privacy settings and the resulting privacy score. By doing so, we are primarily interested in demonstrating empirically the effectiveness of our framework. Secondly, we aim at analyzing to what extent the preferences expressed by the users are in line with a careful and aware behavior w.r.t. their own privacy.

We conduct the experiment as follows. To simulate the active learning framework, for each user and question, i) we start with just five (randomly chosen) labeled friends with which we train the Naive Bayes classifier described in Section 3.4; ii) we test the classifier on the remaining 55 friends and iii) choose the friend whose prediction is the most uncertain, following the maximum entropy criterion (see Equation 15 in Section 3.4); iv) we assign to this friend the same label declared by the participant and v) we re-train the classifier on $5+1$ instances (friends); vi) finally, we test the new classifier on the remaining 54 instances. We repeat iteratively the last four steps until there are no test instances left.

At the end of each prediction step, we measure the following performance parameters:

- the *Accuracy* of the predictions, computed as

$$Accuracy = \frac{\text{number of correctly predicted labels}}{\text{number of test friends}};$$

- the *F-Measure* of the predictions, computed as

$$F\text{-}Measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

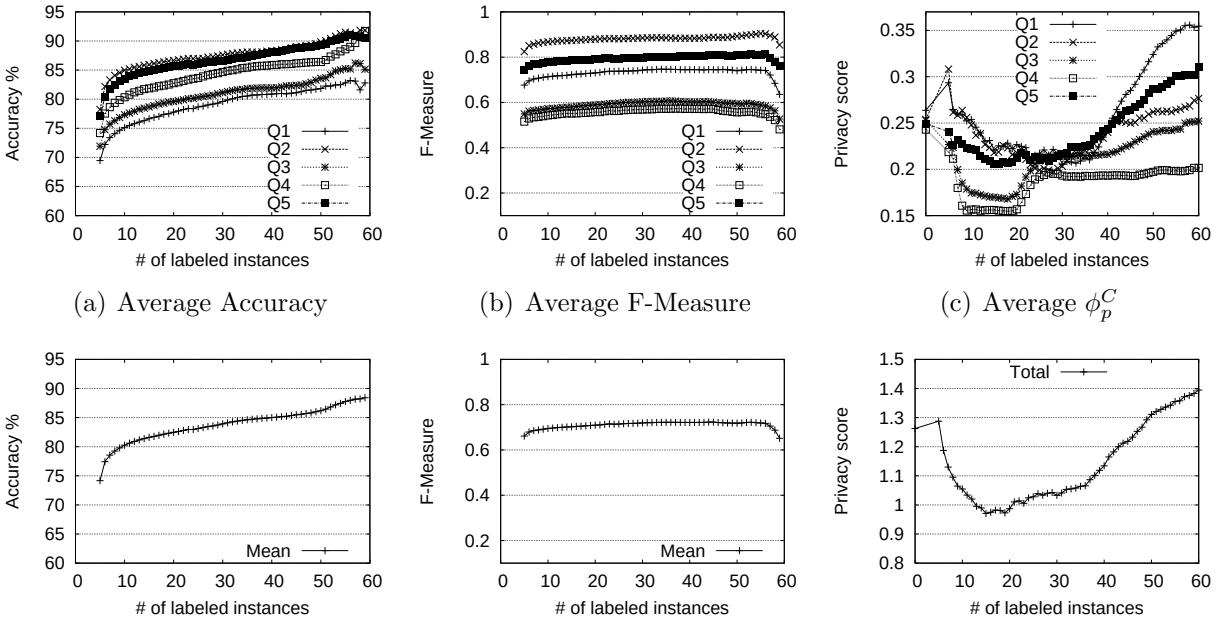
  where precision and recall are computed by considering the *deny* class as the positive one;

- the privacy score (Equation 10) computed by considering both given and predicted $\{allow, deny\}$ labels for all 74 users and applying Equation 16 to calculate matrices $\boldsymbol{M}_i$ and Equation 1 to compute the response matrix $\boldsymbol{R}^C$).

The values of the three parameters are averaged on all 74 users and 30 runs. In each run, the first five labeled friends are chosen randomly. The initial value of the privacy score (when no labels are given) is computed by assigning random labels to all 60 friends. All experiments are performed on a server equipped with 8 Intel Xeon E5-2643 dual core CPU's, 128GB RAM, running Linux (kernel release: 4.0.4).

4.2.2. Average results

The results are provided in Figure 4. The values of the three parameters are reported for each question separately and for all five questions together. As a general observation, the accuracy of the prediction increases significantly with the number of labeled friends (see Figures 4(a) and 4(d)). The growth of the F-Measure is less sharp, instead (Figures 4(b) and 4(e)). We recall that both measures are computed on the test instances only. The small drop of Accuracy and F-Measure in the last steps can be explained by the fact that misclassification errors of few test instances (less than 5 samples) are more likely to happen.



(a) Average Accuracy (b) Average F-Measure (c) Average ϕ_p^C

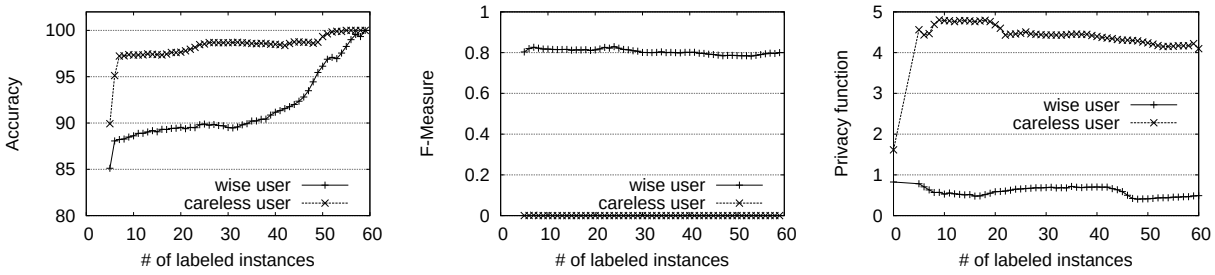
(d) Average Accuracy (mean) (e) Average F-Measure (mean) (f) Average Privacy score (total)

Figure 4: Prediction Accuracy vs. Privacy score: average results

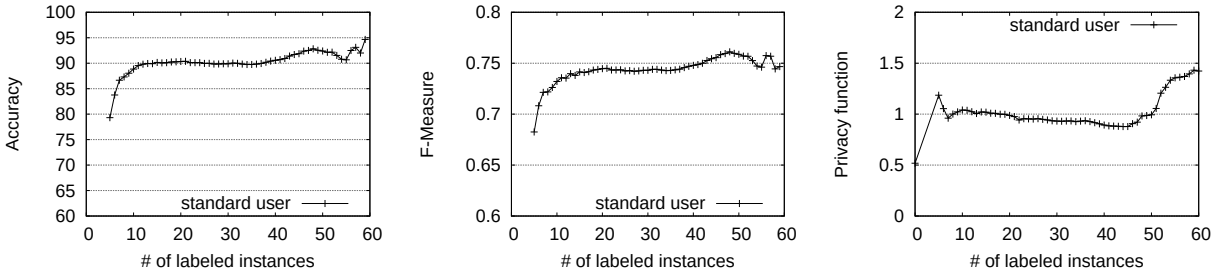
Most importantly, the overall privacy score (Figure 4(f)) starts to decrease when few friends (5 to 15) are labeled, then it starts to grow almost monotonically. This means that, on average, the users don't have a safe behavior w.r.t. their privacy in deciding whether their friends may access to their information or not. However, our framework may help to provide more effective privacy settings by demanding a very limited labeling effort to the users. Interestingly, predictions are more accurate for the two most sensitive questions (Q2 and Q5). In order to augment the readability of the plots, we do not report the standard deviations (error-bars) of the measures. However, they are reasonably low for all measures when the number of labeled friends is under 45. Then, the number of test friends decreases and the stability of the prediction is slightly affected. As an example, we obtain standard deviation values between 0.07 and 0.19 for the F-Measure and between 6.5 and 20 for the Accuracy. Instead, the variability of the privacy score is more pronounced (since it really depends on each users' attitude towards privacy).

4.2.3. Threshold assessment

According to our privacy check routine (see Algorithm 2 in Section 3.5) when a user exceeds a given alarm threshold τ, then she is notified and may possibly adjust her privacy settings. Hence, deciding a congruent value for threshold τ is not without consequences for the system. In fact, not only does it implicitly define the desired safety level of the social network, but it also has an impact on efficiency and usability. If the threshold is too low, many users are notified frequently and system performances may degrade. Furthermore, frequent notifications may annoy most users and compromise their experience. For this



(a) Careless and Wise Users' Accuracy (b) Careless and Wise Users' F-Measure (c) Careless and Wise Users' Privacy score



(d) Standard Users' Accuracy (e) Standard Users' F-Measure (f) Standard Users' Privacy score

Figure 5: Prediction Accuracy vs. Privacy score: results for three typical users

reasons, we also conduct an experiment to verify how many users could be potentially notified depending on increasing values of threshold τ (from 0 to 4) and increasing number of labeled instances (5 to 60). From the results shown in Figure 6, it can be observed that very low threshold values ($\tau < 1.0$) cause too many alarms and notifications. However, for intermediate values of the threshold ($1 < \tau < 2.5$), the number of users exceeding it, in percentage, is below 30%. This experiment also suggests that $\tau = 2.5$ is a reasonable alert threshold for Algorithm 2 which guarantees a reasonable safety level (we recall that the maximum value for the privacy score is 5) and a tolerable number of notifications. It is worth noting that, in our experiments, we do not study the impact of the threshold on users' decisions concerning their privacy settings. This analysis deserves further investigations, but since it requires the definition of a non trivial use study, we leave it for future work.

4.2.4. Typical users' results

Sice the results presented in Section 4.2.2 are on average, we also investigate the behaviour of the three performance parameters on three typical users: a wise user (the one with the lowest non-zero privacy score, computed on the correct labels), a careless user (the one with the highest privacy score) and a standard user (the one exhibiting the privacy score closest to the mean). The results reported in Figure 5(c), show clearly that for a wise user and a careless user, our framework is not useful. However, for a standard user (Figure 5(f)), the active learning algorithm allows its privacy score to decrease and go below 1, confirming the reliability of this threshold for this specific dataset. Notice that the overall accuracy and

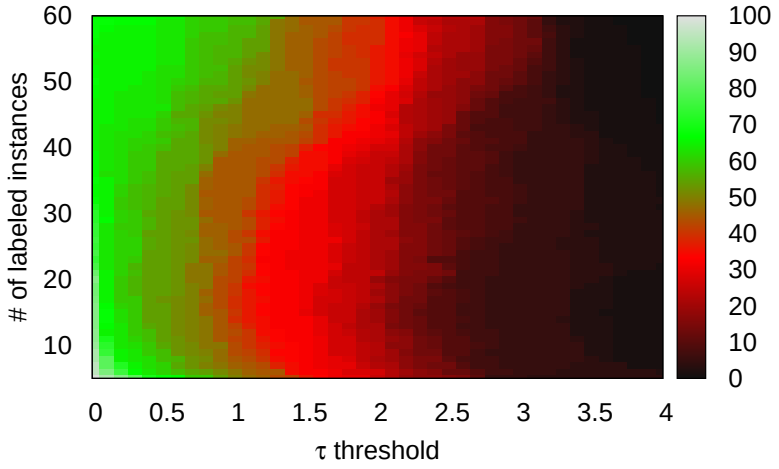


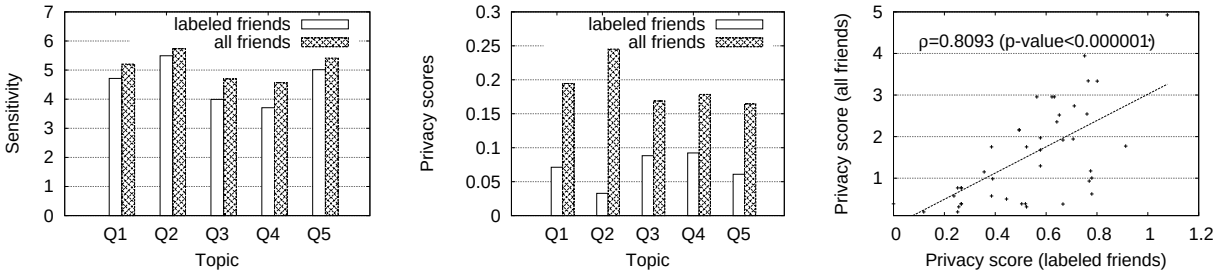
Figure 6: Percentage of notified users for increasing values of the alarm threshold and increasing number of labeled friends

F-Measure of the standard user show that the classifier correctly predicts the *allow*/*deny* classes (See Figures 5(e) and 5(d)). These results also show that with a limited effort (just 20 labeled friends) this user may enhance her privacy protection using settings that follow her preference model. Instead, for the careless user, the F-Measure is 0 since there are no true positives (this user has almost always labeled as *allow* her friends); consequently, precision and recall are both equal to zero. Notice also that now the values of the privacy score are stable: the standard deviations are between 0.01 and 0.21 for the wise user, between 0.06 and 0.27 for the careless user and between 0.01 and 0.20 for the standard user.

4.3. Reliability of the framework

We also study the reliability of the framework by extending the prediction to all participants' friends. Since we do not have the correct labels for friends who do not belong to the list proposed to the participants, we can only measure the privacy score computed on the basis of the predicted set of labels. We compare these measures with the privacy score computed by just considering the labeled friends.

To do that, we first compare the sensitivity values in the two cases (see Figure 7(a)). All questions are subject to an increase of their sensitivity, but when looking at the average privacy scores (Figure 7(b)) we note that all scores are higher than those computed when considering only labeled friends. This means that the visibility of the topics is high. Hence, we perform a correlation analysis in order to check whether the behavior of scores is coherent in the two cases and measure the Pearson's ρ coefficient on the two series of privacy score values. Given the privacy scores $\phi_l(u_i)$ for the labeled case and the privacy scores $\phi_p(u_i)$ for



(a) Sensitivity (b) Privacy scores (c) Correlation analysis

Figure 7: Privacy scores computed with labeled friends only Vs. privacy scores computed on all friends

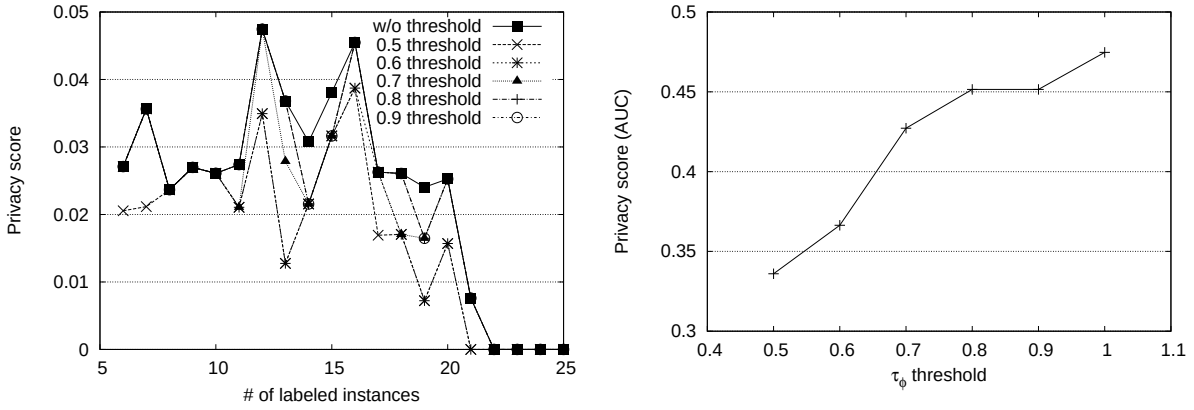
the predicted case, the Pearson's coefficient is computed as:

$$\rho = \frac{\sum_{i=1}^{n} \left(\phi_l(u_i) - \overline{\phi_l}\right)\left(\phi_p(u_i) - \overline{\phi_p}\right)}{\sqrt{\sum_{i=1}^{n} \left(\phi_l(u_i) - \overline{\phi_l}\right)^2}\sqrt{\sum_{i=1}^{n} \left(\phi_p(u_i) - \overline{\phi_p}\right)^2}} \tag{19}$$

where $\overline{\phi_l} = \sum_{i=1}^{n} \phi_l(u_i)/n$ and $\overline{\phi_p} = \sum_{i=1}^{n} \phi_p(u_i)/n$ are the average privacy scores in the two cases ($n = 74$). We obtained a Pearson's coefficient of $\rho = 0.8093$ (see Figure 7(c)) denoting high positive correlation. To assess the significance of this result, we should verify whether the null hypothesis that ρ is not significantly different from zero can be rejected. This can be verified with a two-tailed t-test by observing that the quantity $t = \rho\sqrt{(n-2)/(1-\rho^2)}$ is distributed approximately as the Student t-distribution. In our test, $t = 11.69$, thus the null hypothesis that ρ is not significantly different from zero is rejected with a p-value $p < 0.00001$. These results confirm that: i) the experiments on the limited set of 60 friends per user are significant enough and that, ii) the framework is reliable even for users with a realistic number of friends. Notice that the overall number of friends of the participants spans between 120 and 1558 (with an average of 435).

4.4. Results on the strict framework

To test the strict framework setting presented in Section 3.5.1, a required property is that a privacy score is associated to all user's friend. In a realistic scenario, privacy scores are available for all users in the social network. In our experiments, however, since we asked to label only 60 of each participants' friends, it turns out that the size of the maximal subnetwork of users having the required property is 5. With these numbers it is not possible to compute reliable privacy scores and preference models. Hence, we identify the user u_x who has the largest number of friends among the participants to our online survey and asked her to provide privacy settings labels for them (in fact, her initial set of 60 friends not necessarily include some participants to the survey). Then, we execute the same experimental protocol described in Section 4.2.1. The only difference is that, when predicting the privacy settings of user u_x, we take into account the rule given by Equation 17 for $\tau_\phi \in [0.5, 1.0]$ (with 0.1 step), where the threshold of 1.0 corresponds to the standard framework setting. As before, the results are averaged on 30 runs. The results are reported in Figure 8.



(a) Privacy scores

(b) Privacy scores (AUC)

Figure 8: Results for the strict framework setting

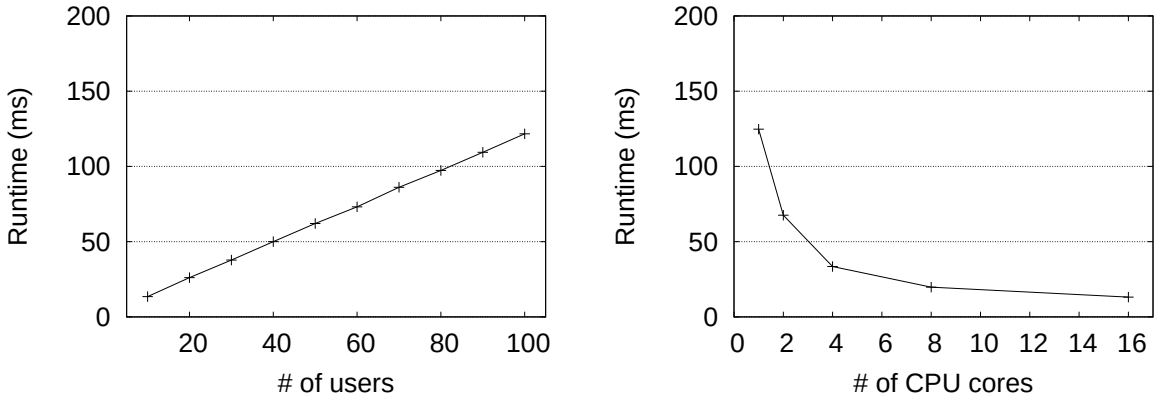
In particular, in Figure 8(a) we observe that, when we introduce the strict settings rule, the privacy scores are always behind those computed in the standard framework. This is also confirmed by Figure 8(b) where we plotted the values of the area under the six curves of Figure 8(a). Furthermore, the figure shows that the overall privacy score increase monotonically with the value of τ_ϕ, i.e., as expected, lower thresholds correspond to safer settings and viceversa. The overall gain, in terms of privacy, is between 5% (for $\tau_\phi = 0.9$) and 30% (for $\tau_\phi = 0.5$).

4.5. Scalability analysis

In Section 3.5.2 we claimed that the overall complexity of a single execution of our routine is linear in the number of users. Here we provide also the empirical evidence of this statement. We let the number of users vary between 10 and 100 and plot the measured runtime averaged on 30 executions $\times$ 56 prediction steps (from 5 to 60 labeled instances). In a realistic scenario, this would correspond to an iteration of the *while* loop of Algorithm 2, when all users are asked to label new friends and all privacy scores are recomputed. Figure 9(a) confirms the linearity of the algorithm w.r.t. the number of users. It also shows that an execution on 100 users requires less than 150 milliseconds. On a network of one million users, the same algorithm would require about 20 minutes. However, the computational time can be reduced further, since our algorithm scales well on multiprocessor systems, as shown in Figure 9(b). To obtain this curve, we have simply implemented the algorithm using the *Callable* multithreading interface of Java, and executed it on all 111 users who answered the second part of our survey. With just 16 cores, our algorithm would take about two minutes to perform a complete execution on one millions users.

5. Conclusions

With the final goal of supporting users' privacy awareness in the Web, we have proposed a framework to keep the privacy risks under control in online social networks. Our



(a) Runtime Vs. Users

(b) Runtime Vs. CPUs

Figure 9: Runtime (in milliseconds) for increasing number of users and CPU cores

framework consists of two main core parts: the computation of a privacy score that can be monitored to alert all users exposed to privacy breaches; ii) an active learning approach to help the exposed users customize their privacy settings by limiting the number of manual operations. We have validated experimentally our framework on an original dataset obtained through a large scale online survey on real Facebook users. The experiments have shown the effectiveness, the reliability and the computational efficiency of our approach. We have also shown that state-of-the-art metrics are based on a distorted perception of sensitivity of published items. Based on these results, we believe that our framework can be easily plugged into any domain-specific or general-purpose social networking platforms without affecting their responsiveness. Furthermore, it may inspire the design of privacy-preserving social networking components for *Privacy by Design* compliant software (Cavoukian, 2012).

In this paper we have investigated the problem from a simplified perspective. In fact, we have considered the problem of sharing a well-defined set of attributes (e.g. work status, relationship status, holidays picture album). To be able to infer the sensitivity of the attributes and to be able to model them from the privacy perspective, they need to belong to classes common in a large part of the population so the behavior of the users with respect to them can be modeled. As a further refinement of this work, we will address the inference of such classes (or topics) for posted items by leveraging NLP, sentiment analysis, topic modeling and text categorization techniques. Moreover, since users, supported by social media tools, often provide additional information on their posted items (e.g., tags, geolocation, user IDs from face recognition in images, hashtags), we will investigate a more complex framework to further define the context of each privacy policy for individual items.

Acknowledgments

This work was supported by Fondazione CRT (grant number 2015-1638). The author wish to thank all the volunteers who participated in the survey.

Akcora, C. G., Carminati, B., & Ferrari, E. (2012a). Privacy in social networks: How risky is your social graph? In *Proceedings of IEEE ICDE 2012* (pp. 9–19). IEEE Computer Society.

Akcora, C. G., Carminati, B., & Ferrari, E. (2012b). Risks of friendships on social networks. In *Proceedings of IEEE ICDM 2012* (pp. 810–815). IEEE Computer Society.

Backstrom, L., Dwork, C., & Kleinberg, J. M. (2011). Wherefore art thou R3579X?: anonymized social networks, hidden patterns, and structural steganography. *Commun. ACM*, *54*, 133–141.

Becker, J., & Chen, H. (2009). Measuring privacy risk in online social networks. In *Proceedings of Web 2.0 Security and Privacy (W2SP) 2009*.

Bioglio, L., & Pensa, R. G. (2017). Modeling the impact of privacy on information diffusion in social networks. In *Proceedings of the 8th Conference on Complex Networks CompleNet 2017* (pp. 95–107). Springer.

Campan, A., & Truta, T. M. (2009). Data and structural k-anonymity in social networks. In *Proceedings of PinKDD 2008* (pp. 33–54). Springer volume 5456 of *LNCS*.

Cavoukian, A. (2012). Privacy by design [leading edge]. *IEEE Technol. Soc. Mag.*, *31*, 18–19.

Cetto, A., Netter, M., Pernul, G., Richthammer, C., Riesner, M., Roth, C., & Sänger, J. (2014). Friend inspector: A serious game to enhance privacy awareness in social networks. In *Proceedings of IDGEI 2014*.

Cormode, G., Srivastava, D., Bhagat, S., & Krishnamurthy, B. (2009). Class-based graph anonymization for social network data. *PVLDB*, *2*, 766–777.

Coscia, M., Rossetti, G., Giannotti, F., & Pedreschi, D. (2014). Uncovering hierarchical and overlapping communities with a local-first approach. *TKDD*, *9*, 6:1–6:27.

Culotta, A., & McCallum, A. (2005). Reducing labeling effort for structured prediction tasks. In *Proceedings of AAAI 2005* (pp. 746–751). AAAI Press / The MIT Press.

Dagan, I., & Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. In *Proceedings of ICML 1995* (pp. 150–157). Morgan Kaufmann.

Dunbar, R. I. M. (2016). Do online social media cut through the constraints that limit the size of offline social networks? *Royal Society Open Science*, *3*.

Fang, L., & LeFevre, K. (2010). Privacy wizards for social networking sites. In *Proceedings of WWW 2010* (pp. 351–360). ACM.

Hay, M., Li, C., Miklau, G., & Jensen, D. (2009). Accurate estimation of the degree distribution of private networks. In *Proceedings of ICDM 2009* (pp. 169–178). IEEE.

Hay, M., Miklau, G., Jensen, D., Towsley, D. F., & Weis, P. (2008). Resisting structural re-identification in anonymized social networks. *PVLDB*, *1*, 102–114.

Kempe, D., Kleinberg, J. M., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of ACM SIGKDD 2003* (pp. 137–146). ACM.

Kosinski, M., Stillwell, D., & Graepel, T. (2013). Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, *110*, 5802–5805.

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of ACM-SIGIR 1994* (pp. 3–12). ACM/Springer.

Litt, E. (2013). Understanding social network site users' privacy tool use. *Computers in Human Behavior*, *29*, 1649–1656.

Liu, K., & Terzi, E. (2008). Towards identity anonymization on graphs. In *Proceedings of ACM SIGMOD 2008* (pp. 93–106). ACM.

Liu, K., & Terzi, E. (2010). A framework for computing the privacy scores of users in online social networks. *TKDD*, *5*, 6:1–6:30.

Liu, Y., Gummadi, P. K., Krishnamurthy, B., & Mislove, A. (2011). Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of ACM SIGCOMM IMC '11* (pp. 61–70). ACM.

Mislove, A., Viswanath, B., Gummadi, P. K., & Druschel, P. (2010). You are who you know: inferring user profiles in online social networks. In *Proceedings of WSDM 2010* (pp. 251–260). ACM.

Misra, G., & Such, J. M. (2016). How socially aware are social media privacy controls? *IEEE Computer*, *49*, 96–99.

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.

Motahari, S., Ziavras, S. G., & Jones, Q. (2010). Online anonymity protection in computer-mediated communication. *IEEE Trans. Information Forensics and Security*, *5*, 570–580.

Pensa, R. G., & di Blasi, G. (2016). A semi-supervised approach to measuring user privacy in online social networks. In *Proceedings of DS 2016* (pp. 392–407). Springer volume 9956 of *LNCS*.

Roberts, S. G. B., Dunbar, R. I. M., Pollet, T. V., & Kuppens, T. (2009). Exploring variation in active network size: Constraints and ego characteristics. *Social Networks*, *31*, 138–146.

Scheffer, T., Decomain, C., & Wrobel, S. (2001). Active hidden markov models for information extraction. In *Proceedings of IDA 2001* (pp. 309–318). Springer volume 2189 of *LNCS*.

Squicciarini, A. C., Lin, D., Sundareswaran, S., & Wede, J. (2015). Privacy policy inference of user-uploaded images on content sharing sites. *IEEE Trans. Knowl. Data Eng.*, *27*, 193–206.

Squicciarini, A. C., Paci, F., & Sundareswaran, S. (2014). Prima: a comprehensive approach to privacy protection in social network sites. *Annales des Télécommunications*, *69*, 21–36.

Such, J. M., & Criado, N. (2016). Resolving multi-party privacy conflicts in social media. *IEEE Trans. Knowl. Data Eng.*, *28*, 1851–1863.

Such, J. M., & Rovatsos, M. (2016). Privacy policy negotiation in social media. *TAAS*, *11*, 4:1–4:29.

Talukder, N., Ouzzani, M., Elmagarmid, A. K., Elmeleegy, H., & Yakout, M. (2010). Privometer: Privacy protection in social networks. In *Workshops Proceedings of ICDE 2010* (pp. 266–269). IEEE.

Task, C., & Clifton, C. (2012). A guide to differential privacy theory in social network analysis. In *Proceedings of ASONAM 2012* (pp. 411–417). IEEE.

Vuokko, N., & Terzi, E. (2010). Reconstructing randomized social networks. In *Proceedings of SIAM SDM 2010* (pp. 49–59). SIAM.

Wang, Y., Gou, L., Xu, A., Zhou, M. X., Yang, H., & Badenes, H. (2015). Veilme: An interactive visualization tool for privacy configuration of using personality traits. In *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI 2015* (pp. 817–826). ACM.

Wang, Y., Nepali, R. K., & Nikolai, J. (2014). Social network privacy measurement and simulation. In *Proceedings of ICNC 2014* (pp. 802–806). IEEE.

Xue, M., Karras, P., Raïssi, C., Kalnis, P., & Pung, H. K. (2012). Delineating social network data anonymization via random edge perturbation. In *Proceedings of CIKM 2012* (pp. 475–484).

Ying, X., & Wu, X. (2011). On link privacy in randomizing social networks. *Knowl. Inf. Syst.*, *28*, 645–663.

Zheleva, E., & Getoor, L. (2008). Preserving the privacy of sensitive relationships in graph data. In *Proceedings of PinKDD 2007* (pp. 153–171). Springer volume 4890 of *LNCS*.

Zheleva, E., & Getoor, L. (2011). Privacy in social networks: A survey. In C. C. Aggarwal (Ed.), *Social Network Data Analytics* (pp. 277–306). Springer US.

Zhou, B., & Pei, J. (2011). The *k*-anonymity and *l*-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowl. Inf. Syst.*, *28*, 47–77.

Zou, L., Chen, L., & Özsu, M. T. (2009). K-automorphism: A general framework for privacy preserving network publication. *PVLDB*, *2*, 946–957.