Heterogeneous Differential Privacy

Mohammad Alaggan*
HCI Lab, CS Department
Helwan University
Cairo, Egypt
malaggan@fci.helwan.edu.eg

Sébastien Gambs
INRIA/IRISA
Université de Rennes 1
Rennes, France
sebastien.gambs@irisa.fr

Anne-Marie Kermarrec
INRIA Rennes Bretagne-Atlantique
Rennes, France
anne-marie.kermarrec@inria.fr

January 24, 2015

Abstract

The massive collection of personal data by personalization systems has rendered the preservation of privacy of individuals more and more difficult. Most of the proposed approaches to preserve privacy in personalization systems usually address this issue uniformly across users, thus ignoring the fact that users have different privacy attitudes and expectations (even among their own personal data). In this paper, we propose to account for this non-uniformity of privacy expectations by introducing the concept of heterogeneous differential privacy. This notion captures both the variation of privacy expectations among users as well as across different pieces of information related to the same user. We also describe an explicit mechanism achieving heterogeneous differential privacy, which is a modification of the Laplacian mechanism by Dwork, McSherry, Nissim, and Smith. In a nutshell, this mechanism achieves heterogeneous differential privacy by manipulating the sensitivity of the function using a linear transformation on the input domain. Finally, we evaluate on real datasets the impact of the proposed mechanism with respect to a semantic clustering task. The results of our experiments demonstrate that heterogeneous differential privacy can account for different privacy attitudes while sustaining a good level of utility as measured by the recall for the semantic clustering task.

1 Introduction

The amount of personal information about individuals exposed on the Internet is increasing by the second. While such data may be used for recommendation and personalization purposes [1, 2, 3, 4, 5], this also raises serious privacy concerns. In recent years, several approaches have been proposed to rely on Privacy-Enhancing Technologies (PETs), whose aim is to preserve privacy while maintaining a good level of utility for the proposed personalization service [6, 7, 8, 9, 10]. One popular approach is the concept of differential privacy [11, 12, 13, 9, 14, 15, 16].

Most of these approaches implicitly assume homogeneity by considering that users have uniform privacy requirements. However, in an environment composed of a myriad of communities, such as the Internet, it is highly plausible that users have heterogeneous privacy attitudes and expectations. For instance, consider a collaborative social platform in which each user is associated to a profile (*e.g.*, a set of URLs that a user has tagged in a system such as Delicious[1]). It is natural to expect that for a particular user some items in his profile are considered more sensitive by him than others, thus calling for a system that can deal with different privacy requirements across items. Similarly, Alice might be more conservative about her privacy than Bob, requiring different privacy requirements across users.

This non-uniformity of privacy attitudes has been acknowledged by major social networking sites [17, 18]. For instance in Facebook, a user can now set individual privacy settings for each item in his profile. However in this particular example, privacy is mainly addressed by restricting, through an access-control mechanism. Our approach can be considered to be orthogonal but complementary to access-control. More precisely, we consider a personalized service, such as a recommendation algorithm, and we enforce the privacy requirements of the user on its output[2].

Furthermore, as highlighted by Zwick and Dholakia in 1999 [19] and as evidenced by anthropological research, privacy attitudes are highly dependent on social and cultural norms. A similar point was raised in 2007 by Zhang and Zhao in a paper on privacy-preserving data mining [20] in which they mentioned that in practice it is unrealistic to assume homogeneous privacy requirements across a whole population. In particular, their thesis is that enforcing the same privacy level across all users and for all types of personal data could lead to an unnecessary degradation of the performance of such systems as measured in terms of accuracy. More specifically, enforcing the same privacy requirements upon all users (even those who do not require it) might degrade the performance in comparison to a system in which strict privacy requirements are only taken into account for those who ask for it. The same type of argument can also be made for different items of the same user. Hence, designing a system supporting heterogeneous privacy requirements could lead to a global improvement of the performance of this system as compared to a homogeneous version. Therefore, the main challenge is to be able to account for the variety of privacy requirements when leveraging personal data for recommendation and personalization.

In this paper, we address this challenge through the introduction of the concept of *heterogeneous differential privacy*, which considers that the privacy requirements are not homogeneous across users and items from the same user (thus providing item-grained privacy). This notion can be seen as an extension of the concept of differential privacy [11] introduced originally by Dwork in the context of databases. We also describe an explicit mechanism achieving heterogeneous differential privacy, which we coin as the "stretching mechanism". We derive a bound on the distortion introduced by our mechanism, which corresponds to a distance between the expected output of the mechanism and the original value of the function to be computed. Finally, we conduct an experimental evaluation of our mechanism on a semantic clustering task using real datasets. The results obtained show that the proposed approach can still sustain a high utility level (as measured in terms of recall) while guaranteeing heterogeneous differential privacy.

The outline of the paper is as follows. First, in Section 2, we describe the background of differential privacy as well as some preliminaries on matrices and sets necessary to understand our work. Afterwards in Section 3, we introduce the novel concept of heterogeneous differential privacy along with the description of an explicit mechanism achieving it before concluding with a discussion on the actual limitations of the approach as well as possible extensions in Section 5. .

[*] Work done while the author was at IRISA, Université Rennes 1.

[1] http://del.icio.us/

[2] Note that systems supporting item-grained privacy can also provide user-grained privacy (*i.e.*, for instance by setting the privacy level of all items in some user's profile to the same value in the privacy setting of this user). However, this assumes that the privacy weights have a global meaning across the entire system, and are not defined only relative to a user.

2 Background

In this section, we briefly introduce the background on differential privacy (Section 2.1) as well as some notions on matrices and sets that are necessary to understand the concept of heterogeneous differential privacy (Section 2.2).

2.1 Differential Privacy

We begin with providing some background of differential privacy, which was originally introduced by Dwork [11] in the context of statistical databases. The main guarantee provided by this approach is that if a differentially private mechanism is applied on a database composed of the personal data of individuals, no output would become significantly more (or less) probable whether or not a participant removes this particular data from the dataset. In a nutshell, it means that for an adversary observing the output of the mechanism, the advantage gained from the presence (or absence) of a particular individual in the database is negligible. This statement is a statistical property about the behavior of the mechanism (*i.e.*, function) and holds independently of the auxiliary knowledge that the adversary might have gathered. More specifically, even if the adversary knows the whole database but one individual row, a mechanism satisfying differential privacy still protects the privacy of this row. The parameter ε is public and may take different values depending on the application (for instance it could be 0.01, 0.1, 0.25 or even 2). While it is sometimes difficult to grasp the intuition about the significance of a particular value for ε [21], a smaller value of ε implies a higher privacy level.

Differential privacy was originally designed for ensuring privacy to individuals who have contributed with their personal data to the construction of a statistical database. In this setting, each individual is a row (*i.e.*, coordinate) in this database (*i.e.*, vector). Differential privacy guarantees that *almost* no difference will be observed to the output of query performed on the database, whether or not the individual (a single row) has contributed to the database by submitting his data, and therefore this information is considered as being protected.

When the database is the profile of a user, which is a vector of items (sometimes called *the micro-data setting*), the whole vector (*i.e.*, database) is owned by a single individual. This difference impacts the interpretation that can be done when speaking about *protecting the privacy of this individual*. In particular, contrary to the first setting of statistical database, an individual does not have the choice to submit or not his data. Rather, if he chooses not to use his profile as input to the collaborative social system, then he will not benefit from the service. However in this new setting, the user is still left with the possibility of selecting a subset of items in his profile before participating. In this case, the main objective of differential privacy is to ensure that when a user adds or removes a single item from his profile, this has a small effect on the output of the computation. However, one caveat is that if the profile of the user contains nothing but items related to a particular sensitive topic (*e.g.*, cancer), then in order to get at least a little bit of utility that information has to be leaked. This observation is in line with the impossibility result of Dwork and Naor stating that if a privacy-preserving mechanism provides any utility, then it has to cause a privacy breach whose magnitude is at least proportional to the min-entropy of the utility [14]. Thus, this limitation is true for any possible privacy-preserving mechanism and is not inherent to the micro-data setting (*i.e.*, this limitation also holds for the database setting).

The difference of a single row between two profiles can be defined formally through the concept of *neighboring profiles*. Each user is associated with a profile representing his personal data, which can be defined as a vector in $\mathbb{R}^n$ (for some n fixed for all users across the system). This representation is generic enough to encompass a variety of possible user profiles. For instance, restricting the domain to $\{0,1\}^n$ can be used to represent a binary string (which is a universal representation) or a subset of items of a global domain of items.

Definition 1 (Neighboring profile)**.** *Two profiles $\boldsymbol{d}, \boldsymbol{d}^{(i)} \in \mathbb{R}^n$ are said to be neighbors if there exists an item $i \in \{1,\dots,n\}$ such that $d_k = d_k^{(i)}$ for all items $k \neq i$. This neighborly relation is denoted by $\boldsymbol{d} \sim \boldsymbol{d}^{(i)}$.*

An equivalent definition states that $\boldsymbol{d}$ and $\boldsymbol{d}^{(i)}$ are neighbors if they are identical except for the i-th coordinate. For instance, the profiles $(0,1,2)$ and $(0,2,2)$ are neighbored while the profiles $(0,1,2)$ and $(0,2,3)$ are not. Differential privacy can be defined formally in the following manner.

Definition 2 (ϵ-differential privacy [11])**.** *A randomized function $\mathcal{M} : \mathbb{R}^n \to \mathbb{R}$ is said to be ε-differentially private if for all neighboring profiles $\boldsymbol{d} \sim \boldsymbol{d}^{(i)} \in \mathbb{R}^n$, and for all outputs $t \in \mathbb{R}$ of this randomized function, the following statement holds:*

$$\Pr[\mathcal{M}(\boldsymbol{d}) = t] \leqslant \exp(\varepsilon) \Pr[\mathcal{M}(\boldsymbol{d}^{(i)}) = t] \ , \tag{1}$$

in which exp *refers to the exponential function.*

Differential privacy aims at reducing the contribution that any single coordinate of the profile can have on the output of a function. The maximal magnitude of such contribution is captured by the notion of (global) *sensitivity*.

Definition 3 (Global sensitivity [15])**.** *The global sensitivity $S(f)$ of a function f is the maximum absolute difference obtained on the output over all neighboring profiles:*

$$S(f) = \max_{\boldsymbol{d} \sim \boldsymbol{d}^{(i)}} |f(\boldsymbol{d}) - f(\boldsymbol{d}^{(i)})| \ , \tag{2}$$

where $\boldsymbol{d} \sim \boldsymbol{d}^{(i)}$ means that $\boldsymbol{d}$ and $\boldsymbol{d}^{(i)}$ are neighboring profiles (cf. *Definition 1*)*.*

Dwork proposed a technique called the *Laplacian mechanism* [15] that achieves ε-differential privacy by adding noise to the output of a function proportional to its global sensitivity. The noise is distributed according to the Laplace distribution (with PDF $\frac{1}{2\sigma}\exp(-|x|/\sigma)$, in which $\sigma = S(f)/\varepsilon$ is a scale parameter).

The novel mechanism that we propose in this paper (to be detailed later) achieves heterogeneous differential privacy by modifying the sensitivity of the function to be released (and therefore the function itself) before applying the standard Laplacian mechanism.

2.2 Preliminaries

Before delving into the details of our approach, we need to briefly introduce some preliminary notions on matrices and sets such as the concept of *shrinkage matrix* [22]. A shrinkage matrix is a linear transformation that maps a vector to another vector with less magnitude, possibly distorting it by changing its direction.

Definition 4 (Shrinkage matrix)**.** *A matrix A is called a shrinkage matrix if and only if $A = \operatorname{diag}(\alpha_1, \dots, \alpha_n)$ such that each diagonal coefficient is in the range $0 \leqslant \alpha_i \leqslant 1$.*

Definition 5 (Semi-balanced set)**.** *A set $D \subseteq \mathbb{R}^n$ of column vectors is semi-balanced if and only if for all shrinkage matrices $A = \operatorname{diag}(\alpha_1, \dots, \alpha_n)$, and for all $\boldsymbol{x} \in D$, we have $A\boldsymbol{x} \in D$.*

For instance, the set

$$\{\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2 \mid 0 < x_1, x_2 < 1\}$$

is a semi-balanced set (imagine a square from $(0,0)$ to $(1,1)$ in the Euclidean plane).

3 Heterogeneous Differential Privacy

In this section, we introduce the novel concept of *heterogeneous differential privacy* (HDP). We start by giving the necessary definitions in Section 3.1, before describing in Section 3.2 how to construct the Stretching Mechanism, which ensures heterogeneous differential privacy. More precisely, we first detail how to construct the privacy-preserving estimator in Section 3.2.1. Afterwards, we discuss why and how the privacy vector expressing the privacy expectations of a user should also be kept private in Section 3.2.3. Finally, an upper bound on the distortion induced by the Stretching Mechanism is provided in Section 3.2.4.

3.1 Definitions

We now define HDP-specific notions such as the concept of *privacy vector*, which is a key notion in HDP. This vector contains the privacy requirements of each coordinate (*i.e.*, item) in the input profile (*i.e.*, vector) of a user, and is defined as follows.

Definition 6 (Privacy vector)**.** *Given a profile $\boldsymbol{d} \in D$ in which D is a semi-balanced set of column vectors composed of n coordinates, let $\boldsymbol{v} \in [0,1]^n$ be the privacy vector associated with the profile* **d***. The owner of item d_i is responsible for choosing the privacy weight v_i associated to this item (by default v_i is set to be 1 if it was not explicitly specified by the owner). A privacy weight v_i of zero corresponds to* absolute privacy *while a value of* 1 *refers to* standard privacy*, which in our setting directly correspond to the classical definition of ε-differential privacy.*

The mere presence of the privacy vector introduces potential privacy breaches, thus this vector should also be protected. Therefore, we need to ensure that in addition to the profile, the privacy vector $\boldsymbol{v}$ also remains private, such that each entry v_i of this vector should only be known by its owner. Otherwise, the knowledge of a privacy weight of a particular item might leak information about the profile itself. For instance, learning that some items have a high privacy weight may reveal that the user has high privacy expectations for and is therefore interested in this specific type of data. We define *heterogeneous differential privacy* in the following manner.

Definition 7 ((Heterogeneous) $(\varepsilon, \boldsymbol{v})$-differential privacy)**.** *A randomized function $\mathcal{M} : D \to \mathbb{R}$ is said to be $(\varepsilon, \boldsymbol{v})$-differentially private if for all items i, for all neighboring profiles $\boldsymbol{d} \sim \boldsymbol{d}^{(i)}$, and for all possible outputs $t \in \mathbb{R}$ of this function, the following statement holds:*

$$\Pr[\mathcal{M}(\boldsymbol{d}) = t] \leqslant \exp(\varepsilon v_i) \Pr[\mathcal{M}(\boldsymbol{d}^{(i)}) = t] \ , \tag{3}$$

in which exp *refers to the exponential function.*

Since a privacy weight $v_i \leqslant 1$, heterogeneous differential privacy implies the standard notion of ε-differential privacy as shown by the following remark.

Remark 1 (Equivalence of $(\varepsilon, \boldsymbol{v})$-DP and ε-DP.)**.** *Let $\overline{\varepsilon} = \varepsilon \overline{v}$ and $\underline{\varepsilon} = \varepsilon \underline{v}$, such that $\overline{v} = \max_i v_i$ (the maximum privacy weight) and $\underline{v} = \min_i v_i$ (the minimum privacy weight). Then, we have: $\underline{\varepsilon}$-DP $\implies (\varepsilon, \boldsymbol{v})$-DP and $(\varepsilon, \boldsymbol{v})$-DP $\implies \overline{\varepsilon}$-DP. As a consequence, $(\varepsilon, \mathbf{1})$-DP holds* if and only if *ε-DP also holds, in which $\mathbf{1} = (1, \cdots, 1)$.*

Finally, we rely on a variant of the notion of global sensitivity, implicitly introduced [23, Lemma 1], that we call *modular global sensitivity*.

Definition 8 (Modular global sensitivity [23])**.** *The modular global sensitivity $S_i(f)$ is the global sensitivity of f when $\boldsymbol{d}$ and $\boldsymbol{d}^{(i)}$ are neighboring profiles that differ on exactly the item i.*

In a nutshell, the modular global sensitivity reflects the maximum difference that a *particular item* i can cause by varying its value (over its entire domain) while keeping all other items fixed.

3.2 The Stretching Mechanism

Thereafter, we describe a generic mechanism achieving heterogeneous differential privacy that we coin as the *Stretching Mechanism*. We assume that the privacy preferences for each item are captured through a privacy vector $\boldsymbol{v}$ (*cf.* Definition 6). Given an arbitrary *total* function $f : D \to \mathbb{R}$, in which D is a semi-balanced set of columns vectors of n coordinates, and whose global sensitivity $S(f)$ is finite, we construct a randomized function $\hat{f}(\boldsymbol{d}, \boldsymbol{v}, \varepsilon)$ *estimating* f while satisfying $(\varepsilon, \boldsymbol{v})$-differential privacy.

Before delving into the details of this method, we provide a little intuition on how and why it works. A lemma in [23, Lemma 1] asserts that the Laplacian mechanism $\mathcal{M}(\boldsymbol{d}) = f(\boldsymbol{d}) + \mathsf{Lap}(\sigma)$ with mean 0 and standard deviation σ provides

$$\Pr[\mathcal{M}(\boldsymbol{d}) = t] \leqslant \exp(\varepsilon_i) \Pr[\mathcal{M}(\boldsymbol{d}^{(i)}) = t] \ , \tag{4}$$

in which $\varepsilon_i = S_i(f)/\sigma$. In other words, differential privacy can be achieved by setting the perturbation induced by the Laplacian mechanism to be proportional to the modular global sensitivity [23] instead of the standard global sensitivity. Therefore, a natural approach for enforcing heterogeneous differential privacy is to manipulate the modular global sensitivity $S_i(f)$ by modifying the function f itself.

3.2.1 Constructing the Estimator

Let $T : [0,1]^n \to \mathbb{R}^{n \times n}$ be a function taking as input a privacy vector $\boldsymbol{v}$ and returning as output a shrinkage matrix, with the property that $T(\mathbf{1}) = I$, such that I is the identity matrix and $\mathbf{1} = (1, \cdots, 1)$. Let also R be a mapping sending a function $f : D \to \mathbb{R}$ and a privacy vector $\boldsymbol{v} \in [0,1]^n$ to the function $R(f, \boldsymbol{v}) : D \to \mathbb{R}$ with $R(f, \boldsymbol{v})(\boldsymbol{d}) = f(T(\boldsymbol{v}) \cdot \boldsymbol{d})$. Recall that the Laplace distribution centered at 0 with scale parameter σ has the following probability density function

$$h(x) = \frac{1}{2\sigma} \exp(-|x|/\sigma) \ . \tag{5}$$

Finally, let N be a Laplacian random variable with parameter $\sigma = \sigma(f, \varepsilon) = S(f)/\varepsilon$, in which $S(f)$ refers to the global sensitivity of the function f and ε the privacy parameter. The following statement proves that this *Stretching Mechanism* R satisfies heterogeneous differential privacy. In the following, all the proofs are omitted but will be present in the full version.

Theorem 1 (Achieving HDP via stretching mecanism)**.** *Given a privacy vector $\boldsymbol{v}$, if the function $T(\boldsymbol{v})$ satisfies $S_i(R(f, \boldsymbol{v})) \leqslant v_i S(f)$ then the randomized function $\hat{f}(\boldsymbol{d}, \boldsymbol{v}, \varepsilon) = R(f, \boldsymbol{v})(\boldsymbol{d}) + N$ satisfies $(\varepsilon, \boldsymbol{v})$-differential privacy.*

In a nutshell, $T(\boldsymbol{v})$ is a shrinkage matrix, whose shrinking factor in each coordinate is computed independently of all other coordinates. More precisely, the shrinking factor for a particular item depends only on the privacy weight associated to this coordinate. The value used by the mechanism is the lowest amount of shrinkage (*i.e.*, distortion) still achieving the target modular global sensitivity of that coordinate. In the following section we provide an explicit construction of $T(\boldsymbol{v})$ for which we prove that by Lemma 1 the condition of Theorem 1 is satisfied, and therefore that $\hat{f}$ achieves $(\varepsilon, \boldsymbol{v})$-differential privacy.

3.2.2 Computing the Shrinkage Matrix

The HDP mechanism $\hat{f}(\boldsymbol{d}, \boldsymbol{v}, \varepsilon)$ adds Laplacian noise to a modified function $R(f, \boldsymbol{v})(\boldsymbol{d}) = f(T(\boldsymbol{v}) \cdot \boldsymbol{d})$. In this section, we specify how to construct $T(\boldsymbol{v})$ such that $\hat{f}$ satisfies HDP. Thereafter, we use R to denote $R(f, \boldsymbol{v})$ for the sake of simplicity. Let $T(\boldsymbol{v}) = \operatorname{diag}(\boldsymbol{w})$ for some $\boldsymbol{w} \in [0,1]^n$ to be computed from the privacy vector $\boldsymbol{v}$ and $S(R, \boldsymbol{w})$ be the sensitivity of $R = f(T(\boldsymbol{v}) \cdot \boldsymbol{d}) = f(\operatorname{diag}(\boldsymbol{w}) \cdot \boldsymbol{d})$ given $\boldsymbol{w}$. Similarly, let $S_i(R, \boldsymbol{w})$ be the modular global sensitivity of R given $\boldsymbol{w}$. We denote by $(\boldsymbol{w}_{-i}, w'_i)$ the vector resulting from replacing the item w_i in $\boldsymbol{w}$ to w'_i (*e.g.*, $(\mathbf{1}_{-i}, w_i) = (1, \ldots, w_i, \ldots, 1)$). Each w_i can be computed from v_i by solving the following optimization problem:

$$\begin{array}{ll} \max & w_i \ , \\ \text{subject to:} & S_i(R, (\mathbf{1}_{-i}, w_i)) \leqslant v_i S(f) \ . \end{array} \tag{6}$$

Note that a solution satisfying this constraint always exists and is reached by setting w_i to 0. The w_i's are never released after they have been computed locally by the rightful owner, and the modular global sensitivity $S_i(R)$ is only used in the proof and is not revealed to the participants, in the same manner as the noise generated. The participants only have the knowledge of the global sensitivity $S(f)$. Thus, the only way in which the profile $\boldsymbol{d}$ could leak is through its side effects to the output, which we prove to achieve ε-DP in Theorem 2.

Lemma 1. *If $T(\boldsymbol{v}) = \operatorname{diag}(\boldsymbol{w})$ such that for all i:*

$$S_i(R, (\mathbf{1}_{-i}, w_i)) \leqslant v_i S(f) \tag{7}$$

(the constraint of (6)) then R satisfies:

$$S_i(R, \boldsymbol{w}) \leqslant v_i S(f) \tag{8}$$

for all i.

3.2.3 Hiding the Privacy Vector

By themselves, the privacy weights could lead to a privacy breach if there are release publicly [24, 23]. For instance, learning that the user has set a high weight on a particular item might be indicated that the user possesses this item on his profile and that he has a high privacy expectation about it. Thus, the impact of the privacy weights on the observable output of the mechanism should be characterized. The following theorem states that when the profile $\boldsymbol{d}$ is fixed, the randomized function $\hat{f}$ satisfies ε-differential privacy over neighboring privacy vectors $\boldsymbol{v} \sim \boldsymbol{v}^{(i)}$. Thus, the privacy vector can also be considered to be hidden and protected by the guarantees of differential privacy.

Theorem 2 (Protecting the privacy vector with ε-DP)**.** *The randomized function $\hat{f}$ provides ε-differential privacy for each individual privacy weight of $\boldsymbol{v}$. This means that for all neighboring privacy vectors $\boldsymbol{v} \sim \boldsymbol{v}^{(i)}$, for all outputs $t \in \mathbb{R}$ and profiles $\boldsymbol{d}$, the following statement holds:*

$$\Pr[\hat{f}(\boldsymbol{d}, \boldsymbol{v}, \varepsilon) = t] \leqslant \exp(\varepsilon) \Pr[\hat{f}(\boldsymbol{d}, \boldsymbol{v}^{(i)}, \varepsilon) = t] \ . \tag{9}$$

3.2.4 Estimating the Distortion Induced by HDP

Intuitively, if $\underline{w}$ is the minimum of $\boldsymbol{w}$ (the diagonal of $T(\boldsymbol{v})$), and $\boldsymbol{d}$ is the input profile, then the distortion introduced by *stretching* the function as measured in terms of absolute additive error is bounded by $1 - \underline{w}$ times the norm of $\boldsymbol{d}$ multiplied by the norm of the gradient of the *semi-stretched* function at $\boldsymbol{d}$.

More formally, let f be a continuous and differentiable function on a semi-balanced set D, and let $(\boldsymbol{v}, \boldsymbol{d}) \in [0,1]^n \times D$ be respectively, the privacy vector and the profile considered. The following theorem provides a bound on the distortion introduced on the output by modifying the global sensitivity of the function f as done by the HDP (*i.e.*, stretching) mechanism described previously.

Theorem 3 (Bound on the distortion induced by the Stretching Mechanism)**.** *Let $f : D \to \mathbb{R}$ be a function from a semi-balanced set D to the reals, and let $\boldsymbol{v} \in [0,1]^n$ be a privacy vector and $T : [0,1]^n \to \mathbb{R}^{n \times n}$ be a function taking a privacy vector to a shrinkage matrix. Finally, let R be a mapping sending a function f and a privacy vector $\boldsymbol{v}$ to the function $R(f, \boldsymbol{v}) : D \to \mathbb{R}$ such that $R(f, \boldsymbol{v})(\boldsymbol{d}) = f(T(\boldsymbol{v}) \cdot \boldsymbol{d})$ for all vectors $\boldsymbol{d}$. The distortion* (i.e.*, distance) between f and $R(f, \boldsymbol{v})$ is bounded by:*

$$|f(\boldsymbol{d}) - R(f, \boldsymbol{v})(\boldsymbol{d})| \leqslant \max_{0 \leqslant c \leqslant 1} (1 - \underline{w}) \|\nabla f(B \cdot \boldsymbol{d})\| \|\boldsymbol{d}\| \ , \tag{10}$$

where $B = cI + (1-c)T(\boldsymbol{v})$, $\underline{w} = \min_i w_i$ is the minimum of $\boldsymbol{w}$ (the diagonal of $T(\boldsymbol{v})$), and ∇f is the gradient of the function f.

This bound is particularly useful in situations in which the norm of the gradient of the function f is bounded from above by a constant. However, even if the norm of the gradient is not bounded by a constant, the bound can still be useful. For instance, in the case of the scalar product function, the bound on the distortion will be $(1 - \underline{w})\|\boldsymbol{d}\|^2$, due to the fact that the gradient of the scalar product function is equal to $\|B \cdot \boldsymbol{d}\| \leqslant \|\boldsymbol{d}\|$ (since B is a shrinkage matrix). One restriction on the application of this bound is that the function f to be protected should have a finite global sensitivity, and therefore the scalar product function mentioned has to restrict its domain to be finite, thus preventing the distortion bound from being infinite.

4 HDP in Practice

To assess the practicality of our approach, we have applied the HDP mechanism on a collaborative social system [1], and evaluated its impact on a related semantic clustering task. In this collaborative social system, each user (*i.e.* node) is associated with a profile. A profile is a binary vector, in which each coordinate represent, for example, a particular URL, and the value of the coordinate is 1 if the user bookmarked or liked that URL. The objective of the semantic clustering task is to assign each node with the k-closest neighbors according to a given similarity metric. In this paper, we use the classical *cosine similarity* (introduced later) to quantify the similarity between two profiles. The task is carried out using a fully distributed protocol, therefore the nodes compute locally (*i.e.*, without relying on a central authority) their similarity with other profiles.

4.1 Applying HDP to Semantic Clustering

In the context of distributed semantic clustering, we are interested in providing heterogeneous differential privacy guarantees to the profiles of nodes (*i.e.*, users). More precisely, we consider the scenario in which a particular user can assign a privacy weight, between 0 and 1, to each item of his profile. The value 0 corresponds to the strongest privacy guarantee in the sense that the presence (or absence) of this item will not affect the outcome (the clustering) at all, while the value 1 is the *lowest* level of privacy possible in our framework (however it still provides the standard guarantees of ε-differential privacy). Thus, the privacy weights of a user directly reflect his privacy attitudes with respect to particular items of his profile, and as a side effect determines the influence of this item in the clustering process. In particular, an item with a higher weight will contribute more to the clustering process, while a item with a lower weight will influence less the resulting clustering.

The *cosine similarity* between two profiles $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as

$$\frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2} \ , \tag{11}$$

in which the operation "$\cdot$" denotes the scalar product. In the following, we apply HDP to the scalar product function and use this modified version to compute the cosine similarity on profiles represented as binary vectors.

Given two profiles $\boldsymbol{x}$ and $\boldsymbol{y}$ and their corresponding indicator functions $\boldsymbol{x}$ and $\boldsymbol{y}$, let $\mathsf{SP}(\boldsymbol{x}, \boldsymbol{y}) = \sum_i x_i y_i$ refers to the scalar product between the two profiles. The privacy vector $\boldsymbol{v}$ is composed of two parts, one for the profile $\boldsymbol{x}$ and the other for the profile $\boldsymbol{y}$: $(\boldsymbol{v}^{\boldsymbol{x}}, \boldsymbol{v}^{\boldsymbol{y}})$. Consider the matrix $T(\boldsymbol{v}) = \mathrm{diag}(v)$ and let $R(\mathsf{SP}, \boldsymbol{v}) = \mathsf{SP}(T(\boldsymbol{v}^{\boldsymbol{x}}) \cdot \boldsymbol{x}, T(\boldsymbol{v}^{\boldsymbol{y}}) \cdot \boldsymbol{y})$ be the Stretching Mechanism, in which T is the stretch specifier. This mechanism R satisfies the premise of Theorem 1 and therefore the choice of $T(\boldsymbol{v}) = \mathrm{diag}(\boldsymbol{v})$ also ensures HDP, as proven in the following lemma.

Lemma 2. *Consider a matrix $T(\boldsymbol{v}) = \mathrm{diag}(\boldsymbol{v})$ and a mechanism $R(\mathsf{SP}, \boldsymbol{v}) = \mathsf{SP}(T(\boldsymbol{v}^{\boldsymbol{x}}) \cdot \boldsymbol{x}, T(\boldsymbol{v}^{\boldsymbol{y}}) \cdot \boldsymbol{y})$, such that $\boldsymbol{x}$ and $\boldsymbol{y}$ correspond to profiles and $v^{\boldsymbol{x}}$ and $v^{\boldsymbol{y}}$ to their associated privacy vectors. In this situation, the following statement is always true: $S_i(R(\mathsf{SP}, \boldsymbol{v})) \leqslant v_i S(\mathsf{SP})$ for all i.*

The previous lemma proves that the proposed modified version of scalar product is differentially private, while the next lemma simply states that if we rely on this differentially private version of scalar product to compute the cosine similarity (or any similar metric), the outcome of this computation will still be differentially private. A standard (*i.e.*, non-heterogeneous) version of the following post-processing lemma can be found in the literature [25], which we have generalized to heterogeneous differential privacy.

Lemma 3 (Effect of post-processing on HDP)**.** *If a randomized function $\hat{f}$ satisfies $(\varepsilon, \boldsymbol{v})$-differential privacy, then for any randomized function $g : \mathsf{Range}(\hat{f}) \to \mathbb{R}$ independent of the input, the composed function $g \circ \hat{f}$ satisfies also $(\varepsilon, \boldsymbol{v})$-differential privacy. The randomness of the function g is assumed to be independent of the randomness of $\hat{f}$ in order for this property to hold.*

Due to lack of space we are unable to fully present our experimental evaluation results, although they are present in the full version where we conduct and in-depth study of the utility. Our results, carried on real-life datasets with synthetic privacy weights, show that the utility is high (close to the utility achieved in non-private settings) as long as the majority of privacy weights are above 0.5. The details are present in the full version.

5 Conclusion

In this work, we have introduced the novel concept of *heterogeneous differential privacy* that can accommodate for different privacy expectations not only per user but also per item as opposed to previous models that implicitly assume uniform privacy requirements. We have also described a generic mechanism achieving HDP called the *Stretching Mechanism*, which protects at the same time the items of the profile of user and the privacy vector representing his privacy expectations across items of the profile. We applied this mechanism for the computation of the cosine similarity and evaluate its impact on a distributed semantic clustering task by using the recall as a measure of utility.

Although the Stretching Mechanism can be applied to a wealth of functions, it is nonetheless not directly applicable to some natural functions, such as the ℓ_0 norm and $\min$. Indeed, when computing the ℓ_0 norm (*i.e.*, the number of non-zero coordinates in a given vector), each coordinate contributes either zero or one regardless of its value. Since the Stretching Mechanism modifies this value, this mechanism would always output the true exact value as long as no privacy weight has been set to exactly zero. For the case of $\min$, due to the fact that the Stretching Mechanism shrinks each coordinate by a factor corresponding to its privacy weight, the resulting output may not have anymore a relation to the intended semantics of the function $\min$.

Another challenge is to enable users to estimate the amount of distortion in the output that they received out of an heterogenous differentially private mechanism. For instance, for functions such as the sum, recipients will not be able to estimate the correct value without being given the distortion. Although the distortion has an upper bound given by Theorem 3, the information needed to compute the upper bound is private. Therefore, releasing the distortion (or even its upper bound) would constitute a violation of privacy. We believe this issue could be solved partially by releasing an upper bound using the traditional Laplacian mechanism at an additional cost of an ε amount of privacy. Another important future work includes the characterization of functions that have a low and high distortion. Indeed, functions having a high distortion are not really suitable for our HDP mechanism. We also leave as open the question of designing a different mechanism than the Stretching Mechanism achieving HDP with a lower distortion.

References

[1] M. Bertier, D. Frey, R. Guerraoui, A.-M. Kermarrec, and V. Leroy, "The Gossple Anonymous Social Network," in *Middleware'10*, 2010, pp. 191–211.

[2] Y. Zeng, N. Zhong, X. Ren, and Y. Wang, "User Interests Driven Web Personalization Based on Multiple Social Networks," in *Proc. of the 4th Intl. Workshop on Web Intelligence & Communities*. ACM, 2012, pp. 9:1–9:4.

[3] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The State-of-the-Art in Personalized Recommender Systems for Social Networking," *Artificial Intelligence Review*, vol. 37, no. 2, pp. 119–132, 2012.

[4] Z. Wen and C.-Y. Lin, "How Accurately Can One's Interests Be Inferred from Friends?" in *Proc. of the 19th Intl. Conf. on World Wide Web*, ser. WWW'10. ACM, 2010, pp. 1203–1204.

[5] F. Liu, C. Yu, and W. Meng, "Personalized Web Search for Improving Retrieval Effectiveness," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 1, pp. 28 – 40, 2004.

[6] E. Toch, Y. Wang, and L. Cranor, "Personalization and Privacy: a Survey of Privacy Risks and Remedies in Personalization-Based Systems," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1-2, pp. 203–220, 2012, 10.1007/s11257-011-9110-z.

[7] M. Alaggan, S. Gambs, and A.-M. Kermarrec, "Private Similarity Computation in Distributed Systems: From Cryptography to Differential Privacy," in *OPODIS'11*, A. F. Anta, G. Lipari, and M. Roy, Eds. Springer, 2011, pp. 357–377.

[8] ——, "BLIP: Non-Interactive Differentially-Private Similarity Computation on Bloom Filters," in *SSS12*, 2012, to appear.

[9] F. McSherry and I. Mironov, "Differentially Private Recommender Systems: Building Privacy into the Net," in *KDD'09*. ACM, 2009, pp. 627–636.

[10] S. Venkatasubramanian, *Privacy-Preserving Data Mining*, ser. Advances in Database Systems. Springer US, 2008, vol. 34, ch. Measures of Anonymity, pp. 81–103.

[11] C. Dwork, "Differential Privacy: a Survey of Results," in *TAMC'08*, M. Agrawal, D.-Z. Du, Z. Duan, and A. Li, Eds., 2008, pp. 1–19.

[12] I. Mironov, O. Pandey, O. Reingold, and S. P. Vadhan, "Computational Differential Privacy," in *CRYPTO'09*, S. Halevi, Ed. Springer, 2009, pp. 126–142.

[13] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. Vadhan, "The Limits of Two-Party Differential Privacy," Electronic Colloquium on Computational Complexity, Tech. Rep. 106, 2011.

[14] C. Dwork and M. Naor, "On the Difficulties of Disclosure Prevention in Statistical Databases or the Case for Differential Privacy," *Journal of Privacy and Confidentiality*, vol. 2, no. 1, pp. 93–107, 2010.

[15] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *TCC'06*, 2006, pp. 265–284.

[16] A. Beimel, K. Nissim, and E. Omri, "Distributed Private Data Analysis: on Simultaneously Solving *How* and *What*," *CoRR*, vol. abs/1103.2626, 2011.

[17] S. Preibusch and A. R. Beresford, "Privacy-Preserving Friendship Relations for Mobile Social Networking," in *Proc. of the W3C Workshop on the Future of Social Networking*, 2009.

[18] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, "Analyzing Facebook Privacy Settings: User Expectations vs. Reality," in *Proc. of the Internet Measurement Conf.*, 2011, pp. 61–70.

[19] D. Zwick and N. Dholakia, "Models of Privacy in the Digital Age: Implications for Marketing and E-Commerce," 1999.

[20] N. Zhang and W. Zhao, "Privacy-Preserving Data Mining Systems," *Computer*, vol. 40, no. 4, pp. 52–58, 2007.

[21] J. Lee and C. Clifton, "How Much is Enough? Choosing ϵ for Differential Privacy," in *Proc. of the 14th Intl. Information Security Conf. (ISC'11)*, X. Lai, J. Zhou, and H. Li, Eds. Springer, 2011, pp. 325–340.

[22] A. Jeffrey, *Matrix Operations for Engineers and Scientists: An Essential Guide in Linear Algebra*. Springer Netherlands, 2010, ch. Linear Transformations and the Geometry of the Plane, pp. 239–272.

[23] P. Dandekar, N. Fawaz, and S. Ioannidis, "Privacy Auctions for Inner Product Disclosures," *CoRR*, vol. abs/1111.2885, 2011.

[24] A. Ghosh and A. Roth, "Selling Privacy at Auction," in *Proc. of the 12th ACM Conf. on Electronic Commerce (EC-2011)*, Y. Shoham, Y. Chen, and T. Roughgarden, Eds. ACM, 2011, pp. 199–208.

[25] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What Can We Learn Privately?" in *FOCS'08*. IEEE Computer Society Washington, DC, USA, 2008, pp. 531–540.