International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 20, No. 6 (2012) 855–876

DOI: 10.1142/S0218488512400272

World Scientific
www.worldscientific.com

SENSITIVITY-INDEPENDENT DIFFERENTIAL PRIVACY VIA PRIOR KNOWLEDGE REFINEMENT

JORDI SORIA-COMAS* and JOSEP DOMINGO-FERRER†

Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili,
UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Catalonia
**jordi.soria@urv.cat*
†*josep.domingo@urv.cat*

Received 31 March 2012
Revised 15 September 2012

We propose a new mechanism to implement differential privacy. Unlike the usual mechanism based on adding a noise whose magnitude is proportional to the sensitivity of the query function, our proposal is based on the refinement of the user's prior knowledge about the response. Our mechanism is shown to have several advantages over noise addition: it does not require complex computations, and thus it can be easily automated; it lets the user exploit her prior knowledge about the response to achieve better data quality; and it is independent of the sensitivity of the query function (although this can be a disadvantage if the sensitivity is small). Furthermore, we give a general algorithm for knowledge refinement and we show some compounding properties of our mechanism for the case of multiple queries; also, we build an interactive mechanism on top of knowledge refinement and we show that it is safe against adaptive attacks. Finally, we give a quality assessment for the responses to individual queries.

Keywords: Differential privacy; knowledge refinement; statistical databases.

1. Introduction

Differential privacy[1,2] is a privacy property of queryable databases that is normally implemented using output perturbation. The disclosure risk limitation offered by differential privacy is based on the limitation of the effect that any single individual has on a query response. If the influence of any single individual on the query response is small, publishing that response involves only a small disclosure risk for any individual.

Any mechanism used to achieve differential privacy may be seen as the application of a perturbation to the real value of the query response. The original proposal[1,2] to attain differential privacy masks the query response by adding a Laplace-distributed noise whose magnitude is proportional to the global sensitivity of the query function (the global sensitivity is the maximum variation of the query function between any two data sets differing in one record, also known as neighbor data sets). The global sensitivity of the query function may be substantially higher than the local sensitivity at a certain data set

(variation of the query function between that data set and its neighbors); hence, adding noise based on global sensitivity may overprotect most data sets. The approach in Ref. 3 tries to avoid this problem by adjusting for each data set the magnitude of the noise to a so-called smooth sensitivity based on the local sensitivities of the query function. Other mechanisms proposed to achieve differential privacy include the exponential mechanism,[4] that introduces the concept of response utility, and some mechanisms designed for specific types of queries, such as Refs. 5 and 6.

All the methods mentioned above are at some point concerned with the (global, smooth or local) sensitivity of the query function. Such mechanisms present two main problems: (i) deciding what amount of noise to add may require complex computations (such as computing the global sensitivity[1] or the smooth sensitivity[3] of the query function) and thus noise addition may be difficult to automate, and (ii) in some cases, the amount of noise that needs to be added is so large that the output may bear little resemblance to the real query response, and thus the response may be misleading.

Differential privacy was introduced as a query-response mechanism where the database is held by a trusted party. Users access the database by submitting queries to it. Queries are mediated by an access mechanism that answers them in a differentially private way. Sometimes this setting is simply too restrictive, because users may want access to the entire database. Hence, as a compromise one should allow for queries that are as flexible as possible. However, the literature on the generation of differentially private data sets is brief[5,7–10] and the available proposals are mainly centered on count queries. The reason for focusing on counts is that current methods to attain differential privacy behave well for count queries. However, even for count queries there may be a big impact on data utility, for example when the data are sparse.[7] Our proposal to reach differential privacy is intended to be one step ahead towards enabling more complex queries to more complex microdata sets.

1.1. *Contribution and plan of this paper*

We introduce a mechanism to achieve differential privacy that works by refining the prior knowledge/beliefs of the database user as much as possible, given the constraints set by differential privacy. Our mechanism depends only on the prior knowledge and on the level of protection that we want to achieve. It is completely independent from the sensitivity of the actual query function, and thus no complex sensitivity computations are required.

This mechanism avoids problem (i) above, as no complex computations are required. Regarding problem (ii), the mechanism guarantees that the response yields increased utility over the prior knowledge that the user had.

Section 2 introduces the idea of knowledge refinement for differential privacy. Section 3 describes a general knowledge refinement mechanism, both for continuous and discrete responses. Section 4 evaluates the level of differential privacy that we get for multi-component queries in terms of the components. Section 5 builds an interactive mechanism on top of knowledge refinement and shows that it is safe against malicious users. Section 6

compares the knowledge refinement approach with Laplace noise addition. Section 7 contains a discussion and Sec. 8 summarizes conclusions.

This paper extends the conference paper.[11] Specifically, new paragraphs and references have been added to the introduction above and to Sec. 7; on the other hand, Secs. 3, 5 and 6 are new.

2. Refining Prior Knowledge to Achieve Differential Privacy

The definition of differential privacy states that the probability that the response belongs to any subset of its range must be similar regardless of whether any specific individual is included or not in the data set.

Definition 1. A randomized function κ gives ε-differential privacy if, for all data sets D_1, D_2 such that one can be obtained from the other by adding or removing a single record, and all $S \subset Range(\kappa)$

$$P(\kappa(D_1) \in S) \leq \exp(\varepsilon) \times P(\kappa(D_2) \in S) \tag{1}$$

A usual approach to satisfying the requirements of Definition 1 is noise addition: first, the real value of the query response is computed and, then, a random noise is added to mask it. A Laplace distribution with zero mean and a scale parameter that depends on the variability of the query function is commonly used for noise addition.

Our proposal is not based on masking the true value of the response by adding some noise, but on modifying the prior knowledge of the database user on the response. When a query is submitted to the database, the user submits at the same time her knowledge/beliefs about the response. We think of this prior knowledge as the probability distribution that the user expects for the response. For example, in case the user has absolutely no idea about the possible result for a query f, the probability distribution to be used is the uniform distribution over the range of f (assuming that this range is bounded). The access mechanism modifies this prior knowledge to fit the real value of the response as much as possible given the constraints imposed by differential privacy.

Some users may be reluctant to provide detailed prior knowledge, because they regard doing so as giving information about themselves to the database. We should usually think of the prior knowledge as the information about the response that is publicly available. Providing the database with such a prior knowledge reveals nothing about the database user. If the database user has information that is not publicly available, she must decide whether to use it as prior knowledge or not; the more accurate the prior knowledge, the more accurate the response will be. We will see in Sec. 6 that, even when little prior knowledge is assumed, knowledge refinement may be superior, in terms of data quality, to noise addition approaches. Therefore, it may make sense to use knowledge refinement even if the database user is not willing to provide all her actual prior knowledge.

Definition 2. Given a query function f, the *prior knowledge* about the response $f(D)$ is the probability distribution P_f, defined over $Range(f)$, that the user expects for the response to f.

The more concentrated the probability mass of P_f around the real value of the response to f, the more accurate is the user's prior knowledge. In general, as the user knows the query f and the set of possible databases D, one may expect her to have some prior knowledge about the response $f(D)$. The better the knowledge the user has on the actual database D, the more accurate is the prior knowledge the user can provide to the response mechanism. If the user's prior knowledge is wrong, the accuracy of the response may suffer. However, whatever the prior knowledge, the refinement procedure guarantees that the output distribution is more accurate than the prior knowledge.

If the query function f has multiple components (dimension $n > 1$), the joint probability distribution must be provided. If the components of f are independent, specifying the marginal distribution for each component is enough to compute the joint distribution. This will also be the case if the components are not independent but the user has no knowledge about the relationship among them.

The access mechanism is run by the database holder as follows:

(1) Receive the query f and the prior knowledge P_f from the database user.
(2) Compute the actual value of the query response, $f(D)$.
(3) Modify P_f to adjust it to $f(D)$ as much as possible, given the constraints imposed by differential privacy.
(4) Randomly sample the distribution resulting from the previous step, and return the sampled value as the response to f evaluated at D.

Even though knowledge refinement works by adjusting the prior knowledge, the output is not the adjusted distribution but a sample from it. This is the usual approach in differential privacy; only a sample from the output distribution is returned. Returning the output distribution itself would leak too much information; in some cases, it could be used to determine the exact value of the query response.

Note that the user cannot pretend to have more knowledge than she actually has: sending a guess as P_f will most likely be wrong and worsen the response quality. Also, we show in Sec. 5 that using several different (fake) prior knowledge distributions to mount adaptive attacks does not succeed in breaking ε-differential privacy.

The critical step is the adjustment of the prior knowledge to the real query response. To perform this adjustment, we distinguish two types of queries: statistical queries and individual queries. We call statistical queries those whose outcome depends on multiple individuals, while individual queries are those that depend on a single individual. It will be shown below that a finer adjustment of the prior knowledge is feasible for individual queries. We start by focusing on statistical queries, but, before formally specifying the response mechanism, we give an example to illustrate what we intend to do.

Example 1. Assume a query function f that is known to return a value within the interval $[0, 1]$. Assume also that the database user has no further knowledge about the query response, *i.e.* her prior knowledge is $U[0, 1]$, the uniform distribution over $[0, 1]$.

To refine the prior knowledge, we modify its density by applying two multiplicative factors: $\alpha_u \geq 1$ to the points near $f(D)$, and $\alpha_d \leq 1$ to the points farther from $f(D)$. In

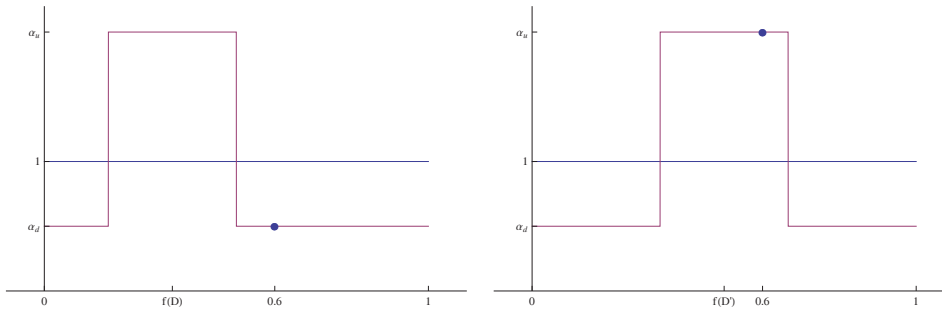

Fig. 1. Distributions for the response to $f(D)$ (left) and to $f(D')$ (right).

this way, the probability of obtaining as the response a value near the actual response $f(D)$ is increased with respect to the prior knowledge, while the probability of obtaining a distant value is decreased. Figure 1 shows the probability distribution resulting from applying the procedure described above for a pair of neighbor data sets D and D'.

To obtain ε-differential privacy, the density at a given point for the response to $f(D)$ must be a factor within the interval $[e^{-\varepsilon}, e^{\varepsilon}]$ of the density at the same point for the response to $f(D')$. Check, for example, the point 0.6 in Fig. 1: on the left-hand side distribution, the point is far from the real response and thus a factor α_d is applied; on the right-hand side distribution, the point is near the real response and the factor applied is α_u. For the ε-differential privacy condition to hold, it must be $\alpha_u/\alpha_d \leq e^{\varepsilon}$. We can also think in the reverse way: given two constants $\alpha_u \geq 1$ and $\alpha_d \leq 1$, the level of differential privacy achieved by this response mechanism is $\varepsilon = \ln(\alpha_u/\alpha_d)$.

Note that, to obtain a valid density function from the above modification, the set of points over which each of the factors α_u and α_d are applied must be selected in such a way that the total probability mass of the resulting distribution equals 1. If we denote by $\mathcal{U}_u$ the set over which we apply the factor α_u, for the total probability mass of the adjusted distribution to be 1, we must have $\alpha_u P_f(\mathcal{U}_u) + \alpha_d(1 - P_f(\mathcal{U}_u)) = 1$. If the prior knowledge is an absolutely continuous distribution, as in Example 1, for any pair of values $\alpha_u \geq 1$ and $\alpha_d \leq 1$ it is possible to select a set $\mathcal{U}_u$ in such a way that $\alpha_u P_f(\mathcal{U}_u) + \alpha_d(1 - P_f(\mathcal{U}_u)) = 1$ is satisfied. The reason is that we can select the set $\mathcal{U}_u$ to have any probability mass between 0 and 1. If the prior knowledge distribution is not absolutely continuous, it may not be possible to find a set $\mathcal{U}_u$ with the required probability mass for the given values α_u and α_d. This section assumes that such a set $\mathcal{U}_u$ exists. In Sec. 3, we specify a general algorithm that works for any prior knowledge distribution.

The following proposition formalizes the ideas discussed in the previous example.

Proposition 1. *Let $f{:}\mathcal{D} \to \mathbb{R}^n$ be a query function and let P_f be the prior knowledge for $f(D)$. Let $\alpha_u \geq 1$ and $\alpha_d \leq 1$ be such that $\alpha_u = e^{\varepsilon}\alpha_d$. Let $\mathcal{U}_u$ be an environment of $f(D)$ satisfying $\alpha_u P_f(\mathcal{U}_u) + \alpha_d(1 - P_f(\mathcal{U}_u)) = 1$. The response mechanism that returns a value randomly sampled from the distribution obtained by modifying P_f through multiplication*

of the probability mass of the points in $\mathcal{U}_u$ by α_u, and multiplication of the probability mass of the points outside $\mathcal{U}_u$ by α_d, satisfies ε-differential privacy.

When the query f returns a value related to a single individual, the mechanism in Proposition 1 can be improved. In that case, there are only two possibilities for the response: (i) if the individual we are asking about is not in the database, the distribution of the response equals the prior knowledge distribution, and (ii) if the individual is in the database, the distribution for the response will be the result of refining the prior knowledge. To satisfy ε-differential privacy, we only need to guarantee that the distribution resulting from (i) and (ii) does satisfy the limitation on the knowledge gain imposed by differential privacy. In other words, the output distribution need only be compared to the prior knowledge. The conditions that must hold are $1 \leq \alpha_u \leq e^{\varepsilon}$ and $e^{-\varepsilon} \leq \alpha_d \leq 1$.

Note that, by choosing $\alpha_u = e^{\varepsilon}$ and $\alpha_d = e^{-\varepsilon}$, the level of differential privacy that we can guarantee for a statistical query function (depending on multiple individuals) is 2ε, while for an individual query (whose outcome depends on a single individual), we double the guarantee to ε.

Proposition 2. *Let $f{:}\mathcal{D} \rightarrow \mathbb{R}^n$ be an individual query in the above sense and let P_f be the prior knowledge distribution for f. Let $\alpha_u = e^{\varepsilon}$ and $\alpha_d = e^{-\varepsilon}$. Let $\mathcal{U}_u$ be an environment of $f(D)$ satisfying $\alpha_u P_f(\mathcal{U}_u) + \alpha_d(1 - P_f(\mathcal{U}_u)) = 1$. The response mechanism that returns a value randomly sampled from the distribution obtained by modifying P_f through multiplication of the probability mass of the points in $\mathcal{U}_u$ by α_u, and multiplication of the probability mass of the points outside $\mathcal{U}_u$ by α_d, satisfies ε-differential privacy.*

3. A General Algorithm for Knowledge Refinement

Propositions 1 and 2 above state that, given appropiate factors α_u and α_d and a set $\mathcal{U}_u$ with the required probability mass, the knowledge refinement mechanism satisfies ε-differential privacy. However, some details were left aside in the previous section: (i) how is the set $\mathcal{U}_u$ selected?, and (ii) can we still apply knowledge refinement if a set $\mathcal{U}_u$ with the required probability mass does not exist? This section gives a more detailed view of the knowledge refinement mechanism and answers the two aforementioned questions.

Knowledge refinement works by increasing the probability mass of the points near $f(D)$, and by decreasing the probability mass of the rest of points in such a way that the total probability mass equals one. In Example 1 there was a natural way to determine the set $\mathcal{U}_u$: the points closest to $f(D)$ in absolute value. However, such a natural way does not always exist, as illustrated in the next example.

Example 2. To determine the form of the set $\mathcal{U}_u$ for a query function with two components, say $f = (f_1, f_2)$, we use a distance function defined over the range of f, namely $d : Range(f_1) \times Range(f_2) \rightarrow [0, \infty)$. If d does not treat f_1 and f_2 symmetrically, then one component is given priority over the other. In fact, there is no natural way to define d and hence $\mathcal{U}_u$. Such definitions are application-dependent.

Table 1. Example distance functions for univariate query functions depending on the type of query result.

Query result	$Range(f)$	Distance
continuous	$\mathbb{R}$	$d(x,y) = \lvert x-y \rvert$
nominal	$\{c_1, \ldots, c_n\}$	$d(c_i,c_j) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases}$
ordinal	$\{c_1, \ldots, c_n\}$	$d(c_i,c_j) = \lvert i-j \rvert$

Table 1 shows some distance functions that are appropriate for a query with a single component in terms of the type of the result. We do not provide any distance for multivariate queries because such distances are very application-dependent, as pointed out in Example 2.

Note that when we feed the knowledge refinement algorithm with a certain distance function, we are instructing it with the sets that we want to favor. Given a value $f(D)$, we modify the probability that the prior knowledge assigns to the points in $Range(f)$ according to the distance d. If a point at distance r is being applied a factor α_1, all points at distance r must be applied the same factor, and points at a shorter distance must be applied a factor α_2 with $\alpha_2 \geq \alpha_1$. Therefore, the set $\mathcal{U}_u$ of points that has its probability increased must be of the form $\mathcal{U}^1_{f(D),r}$ or $\mathcal{U}^2_{f(D),r}$, for some $r \in [0,\infty)$, where:

$$\begin{aligned} \mathcal{U}^1_{f(D),r} &= \{x \in Range(f) : d(f(D),x) \leq r\}\,, \\ \mathcal{U}^2_{f(D),r} &= \{x \in Range(f) : d(f(D),x) < r\}\,. \end{aligned} \tag{2}$$

The set $\mathcal{U}_d$ of points that has its probability decreased is the complement of $\mathcal{U}_u$, that is, $\mathcal{U}_d = Range(f) \setminus \mathcal{U}_u$.

We want to choose two multiplicative factors α_u and α_d to modify the probability mass of $\mathcal{U}_u$ and $\mathcal{U}_d$, respectively. Factors α_u and α_d must be selected so that differential privacy holds and the total probability mass of the resulting modified distribution equals one.

Table 2 shows the form of factors α_u and α_d for the two types of queries considered in Sec. 2: individual and statistical. For the case of individual queries, the differential privacy condition need only hold between the distribution of the response and the prior knowledge. Any pair of values $\alpha_u \in [1, e^\varepsilon]$ and $\alpha_d \in [e^{-\varepsilon}, 1]$ yields ε-differential privacy; however, $\alpha_u = e^\varepsilon$ and $\alpha_d = e^{-\varepsilon}$ yield the greatest knowledge gain.

For statistical queries, the condition must hold for each pair of distributions for the response to the query over data sets that differ in a single record. Therefore, we must have $\alpha_u/\alpha_d \leq e^\varepsilon$. Same as for individual queries, the greatest knowledge gain is achieved when $\alpha_u/\alpha_d = e^\varepsilon$. The actual values of α_u and α_d must belong to the intervals $[1, e^\varepsilon]$ and $[e^{-\varepsilon}, 1]$, respectively, but they can be freely chosen, as long as $\alpha_u/\alpha_d? \leq e^\varepsilon$ holds and the total probability mass is one:

$$\alpha_u P_f(\mathcal{U}_u) + \alpha_d P_f(\mathcal{U}_d) = 1\,. \tag{3}$$

Table 2. Form of the factors α_u and α_d for individual and statistical queries.

Type of query	Factors
individual	$\alpha_u = e^{\varepsilon}$, $\alpha_d = e^{-\varepsilon}$
statistical	$\alpha_u \in [1, e^{\varepsilon}]$, $\alpha_d \in [e^{-\varepsilon}, 1]$ with $\alpha_u/\alpha_d = e^{\varepsilon}$

For statistical queries, the specific values selected for α_u and α_d determine the maximum knowledge gain for the points in $\mathcal{U}_u$ and $\mathcal{U}_d$, where the gain is understood as the modification w.r.t. the prior knowledge P_f. Assuming that $\alpha_u/\alpha_d = e^{\varepsilon}$ holds, a greater value for α_u provides increased knowledge gain for the points in $\mathcal{U}_u$, but it also results in a greater value for α_d, because otherwise $\alpha_u/\alpha_d \leq e^{\varepsilon}$ would not be satisfied; this implies decreasing the knowledge gain for the points in $\mathcal{U}_d$ with respect to the prior knowledge.

For fixed values of the factors α_u and α_d, from Eq. (3) and $P_f(\mathcal{U}_d) = 1 - P_f(\mathcal{U}_u)$, we have:

$$P_f(\mathcal{U}_u) = \frac{\alpha_u - 1}{\alpha_u - \alpha_d}$$

$$P_f(\mathcal{U}_d) = \frac{1 - \alpha_d}{\alpha_u - \alpha_d}$$

For continuous prior knowledge, it is always possible to select sets $\mathcal{U}_u$ and $\mathcal{U}_d$ with the above probability masses. In this case, the knowledge refinement mechanism is very simple: apply factor α_u to $\mathcal{U}_u$ and factor α_d to $\mathcal{U}_d$, as stated in Propositions 1 and 2.

For other kinds of prior knowledge, the sets $\mathcal{U}_u$ and $\mathcal{U}_d$ with the required probability masses may not exist. In such cases, we still want to apply the factor α_u to the greatest possible set of points closest to $f(D)$, and the factor α_d to the greatest possible set of points farthest from $f(D)$, thus achieving the maximum knowledge gain at such points. We denote $\mathcal{U}'_u$ the set that is applied factor α_u, and $\mathcal{U}'_d$ the set that is applied factor α_d. For the remaining points we adjust their factor to have a total probability mass of one. See Algorithm 1 for a detailed description of the process; this algorithm is run by the database holder.

It is easy to check that the total probability mass of the distribution equals one, no matter whether the **then** or the **else** option of the **if** statement of Algorithm 1 is taken. Regarding the differential privacy condition, we have already seen that it holds for the **then** case. For the **else** case, differential privacy also holds, because α_{ud} belongs to the interval $[\alpha_d, \alpha_u]$.

Differential privacy is usually criticized for the low utility of the results it provides.[12] Several relaxations of ε-differential privacy have been proposed; in particular, the authors of[13] propose (ε, δ)-differential privacy (*a.k.a* (ε, δ)-indistinguishability), and (ε, δ)-probabilistic differential privacy. The former property relaxes the strict requirement of differential privacy by adding a non-zero δ. The latter property allows arbitrarily large knowledge gains within probability δ. Let us briefly review (ε, δ)-privacy and sketch how prior knowledge refinement can achieve it.

Algorithm 1 Knowledge refinement algorithm to respond to query $f(D)$ for a general prior knowledge

Input parameters: query f, prior knowledge P_f of the database user, distance function d, factors α_u and α_d from the database holder.

(1) Compute the actual value of the query response, $f(D)$.
(2) Modify P_f to adjust it to $f(D)$ as much as possible, given the constraints imposed by differential privacy. This is done as follows:
   (a) Let $p_u = (\alpha_u - 1)/(\alpha_u - \alpha_d)$.
   (b) Let $p_d = (1 - \alpha_d)/(\alpha_u - \alpha_d)$.
   (c) **if** there exists a set $\mathcal{U}_u$ of the form $\mathcal{U}^1_{f(D),r}$ or $\mathcal{U}^2_{f(D),r}$ (see Expression 2) with $P_f(\mathcal{U}_u) = p_u$ **then**

   Build the distribution of the response to $f(D)$ by applying the factor α_u to $\mathcal{U}_u$, and α_d to $Range(f) \setminus \mathcal{U}_u$.

   else

      i. Find the maximal set $\mathcal{U}'_u$ of the form $\mathcal{U}^1_{f(D),r}$ or $\mathcal{U}^2_{f(D),r}$ with $P_f(\mathcal{U}'_u) < p_u$.
      ii. Find the maximal set $\mathcal{U}'_d$ of the form $Range(f) \setminus \mathcal{U}^1_{f(D),r}$ or $Range(f) \setminus \mathcal{U}^2_{f(D),r}$ with $P_f(\mathcal{U}'_d) < p_d$.
      iii. Let $p_{ud} = 1 - P_f(\mathcal{U}'_u) - P_f(\mathcal{U}'_d)$ be the probability of the points not in $\mathcal{U}'_u \cup \mathcal{U}'_d$
      iv. Let $\alpha_{ud} = (1 - \alpha_u p_u - \alpha_d p_d)/(1 - p_u - p_d)$ be the factor to be applied to $Range(f) \setminus (\mathcal{U}'_u \cup \mathcal{U}'_d)$
      v. Build the distribution of the response to $f(D)$ by applying:
         - factor α_u to points in $\mathcal{U}'_u$
         - factor α_d to points in $\mathcal{U}'_d$
         - factor α_{ud} to points in $Range(f) \setminus (\mathcal{U}'_u \cup \mathcal{U}'_d)$.

(3) Randomly sample the distribution resulting from the previous step, and return the sampled value as the response to f evaluated at D.

Definition 3. A randomized function gives (ε, δ)-differential privacy if, for all data sets D_1, D_2 such that one can be obtained from the other by adding or removing a single record, and all $S \subset Range(\kappa)$

$$P(\kappa(D_1) \in S) \leq \exp(\varepsilon) \times P(\kappa(D_2) \in S) + \delta \tag{4}$$

As ε-differential privacy implies (ε, δ)-differential privacy, Algorithm 1 can be used to obtain (ε, δ)-differential privacy. However, a simple modification to Algorithm 1 can offer better data utility while still satisfying (ε, δ)-differential privacy (but no longer ε-differential privacy). We do not provide a formal algorithm with the required modifications, but the idea is to use the extra margin δ to increase the probability at $f(D)$ and reduce it at the points farthest from $f(D)$.

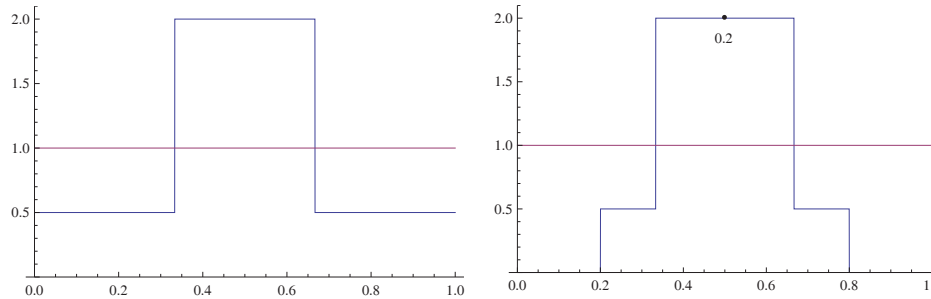

Fig. 2. Distribution of the response to an individual query when $f(D) = 0.5$, for $\ln 2$-differential privacy (left) and $(\ln 2, 0.2)$-differential privacy (right).

Just like it happened for ε-differential privacy, the improvement of (ε, δ)-privacy for individual queries is greater than for statistical queries. For an individual query, we only need to compare the distribution of the response with the prior knowledge (see Fig. 2). As the prior knowledge is not modified, we can modify the response by adding δ to the probability mass of $f(D)$, and subtract δ from the tails of the distribution.

For a statistical query, we also want to increase the probability mass of the actual response $f(D)$, while reducing the probability mass of the set $S'_{f(D)}$ of points farthest from $f(D)$. Although other schemes are possible, a sensible choice is to have the probability mass of $f(D)$ increased by the same amount δ', whatever the data set D. As we have to keep the total probability mass equal to one, we must decrease the probability of $S'_{f(D)}$ by δ'. Now, since we can select data sets D_1 and D_2 such that $f(D_1)$ belongs to $S'_{f(D_2)}$, for Inequality (4) to hold for $S'_{f(D_2)}$, it must be $\delta' = \delta/2$ (it can also be $\delta' < \delta/2$, but then we are not taking advantage of the whole δ margin).

4. Differential Privacy in Multicomponent Queries

The knowledge refinement mechanism as introduced in Sec. 2 is independent of the number of components of the query function. However, for the case of multicomponent queries, we can relate the level of differential privacy for the multicomponent query to the level of differential privacy of the components. If we have a query $f = (f_1, \ldots, f_n)$ and for each of the components, f_i, we get an ε_i-differentially private response, then we get a $\sum_{i=1}^{n} \varepsilon_i$-differentially private response for f. This is in fact a property of ε-differential privacy, hence a proof for our specific mechanism is not required (see[4]).

The above result on multicomponent queries can be improved when each of the queries refers to a disjoint set of individuals. For the noise addition mechanism, it easy to see that, when performing queries $f_1, \ldots, f_n$ that refer each to a disjoint set of individuals, the global sensitivity equals the maximum of the sensitivities of the individual queries.[1] The reason is that, by adding or removing a single individual from the data set, only one of the queries is affected. This is a good property, as it guarantees $\max\{\varepsilon_i\}$-differential privacy instead of $\sum \varepsilon_i$-differential privacy. Our goal is to show that this property can also

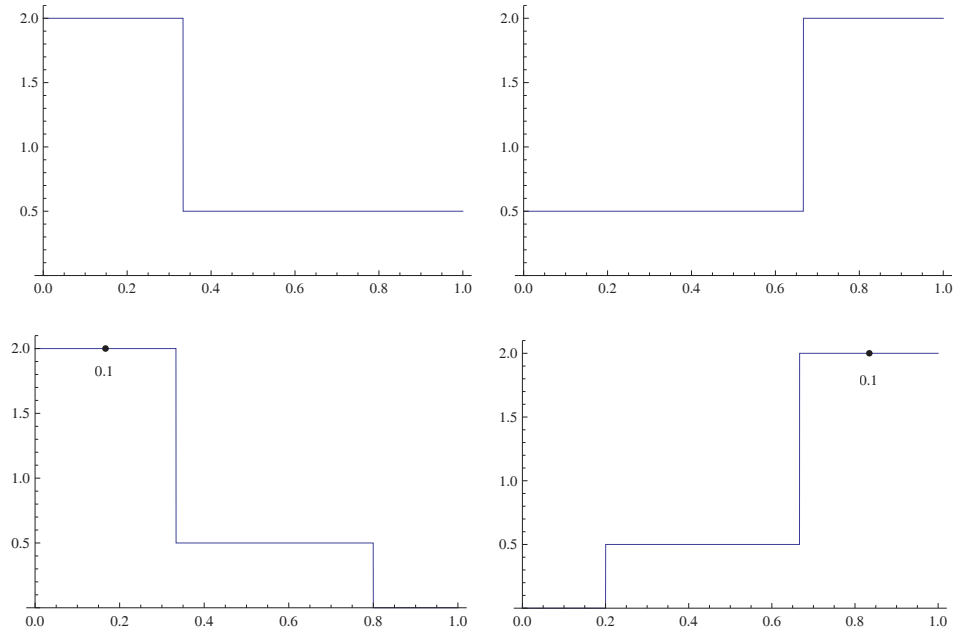

Fig. 3. Distribution for the response to a statistical query when $f(D) = 0.166$ (left) and $f(D) = 0.833$ (right), for $\ln 2$-differential privacy (top) and $(\ln 2, 0.2)$-differential privacy (bottom).

be achieved for our proposal. In fact, we will show further on that this is also a general property of differential privacy. We start with an example.

Example 3. Let D be a database with two attributes: an identifier ID and a Boolean attribute B. Let f_1 and f_2 be queries that return the value of B for individuals 1 and 2, respectively. Let the prior knowledge for both queries be the independent uniform distribution over the set $\{0, 1\}$, which assigns a prior probability 0.5 to each of the possible outcomes for each query. To respond to f_1 in an ε-differentially private way with $\varepsilon = 1$, we select factors $\alpha_u = e^{\varepsilon}$ and $\alpha_d = e^{-\varepsilon}$ that modify the prior knowledge. The same factors are selected for f_2. Now we want to check whether the combination of responses to f_1 and f_2 is still ε-differentially private.

For the sake of simplicity, we assume that both individuals are in D, and that $f_1(D) = 0$ and $f_2(D) = 0$. For the rest of cases we would proceed in a similar way. Figure 4 shows the prior knowledge and the output distribution for both query functions f_1 and f_2. Indeed, by setting $\alpha_d = e^{-\varepsilon}$ and adjusting the probability mass to one instead of setting $\alpha_u = e^{\varepsilon}$, we have

$$P(K_{f_1}(D) = 1 | f_1(D) = 0) = P(K_{f_1}(D) = 1 | f_2(D) = 0) = 0.5\alpha_d = 0.5e^{-1} = 0.1839$$

$$P(K_{f_1}(D) = 1 | f_1(D) = 1) = P(K_{f_1}(D) = 1 | f_2(D) = 1) = 1 - 0.5\alpha_d = 0.8161\,.$$

Table 3 shows the joint distribution for the output of (f_1, f_2), which is obtained by multiplying the output distributions for f_1 and f_2.

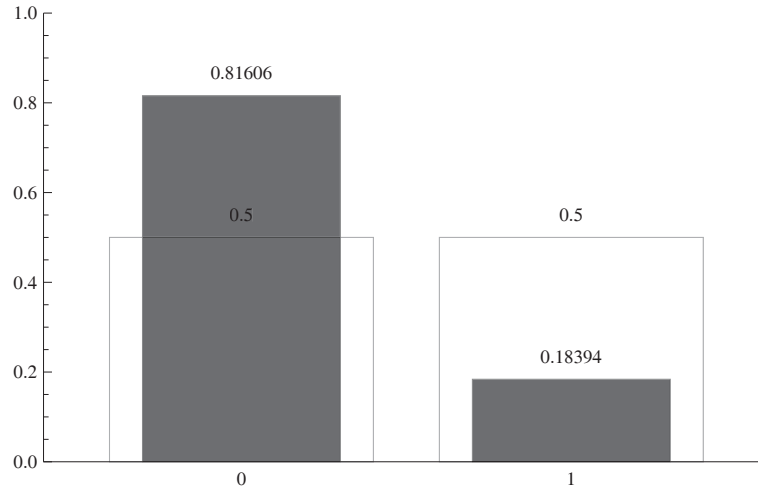

Fig. 4. Prior knowledge about attribute B and distribution of the ε-differentially private response to query functions f_1 and f_2, assuming that the actual value for attribute B is 0.

Table 3. Distribution of the differentially private response to the two-component query (f_1, f_2) when the true values are $f_1(D) = f_2(D) = 0$.

		K_{f_1}: 0	K_{f_1}: 1
	K_{f_2}		
0	$1 - 0.5\alpha_d$	$(1 - 0.5\alpha_d)^2$	$(1 - 0.5\alpha_d)0.5\alpha_d$
1	$0.5\alpha_d$	$(1 - 0.5\alpha_d)0.5\alpha_d$	$0.25\alpha_d^2$

For ε-differential privacy to hold for the two-component query $f = (f_1, f_2)$, the ratio of the response distribution at D and the response distribution at any D' that results from D by adding or removing a single individual must be within the range $[e^{-\varepsilon}, e^{\varepsilon}]$. As f_1 and f_2 are related to individuals 1 and 2, any modification to D that does not affect the records for those individuals leaves the distribution of responses unchanged. As we are assuming that individuals 1 and 2 are in D, the only modifications to be considered are the removal of one of these individuals. Table 4 shows the distributions of responses when individual 1 or 2 are removed. We use K_f to denote the distribution of the response to query f. It can be seen that the respective ratios between the distribution in Table 3 and the ones in Table 4 are within $[e^{-\varepsilon}, e^{\varepsilon}] = [e^{-1}, e]$; specifically, the ratios take only two values, $\alpha_d = e^{-1}$ and $2 - \alpha_d = 2 - e^{-1}$.

We now state and prove in general the property illustrated in the previous example.

Proposition 3. *Let D be a data set and let $(f_1, \ldots, f_n)$ be a set of query functions related to disjoint sets of individuals. Let K_{f_i} be a random variable that provides ε_i-*

Table 4. Distribution of the response to query $f = (f_1, f_2)$ when either individual 1 is missing (top) or individual 2 is missing (bottom), and when the attribute value for the non-missing individual is 0.

K_{f_2}	K_{f_1}: 0	K_{f_1}: 1
	0.5	0.5
0	$1 - 0.5\alpha_d$	$0.5(1 - 0.5\alpha_d)$	$0.5(1 - 0.5\alpha_d)$
1	$0.5\alpha_d$	$0.25\alpha_d$	$0.25\alpha_d$

K_{f_2}	K_{f_1}: 0	K_{f_1}: 1
	$1 - 0.5\alpha_d$	$0.5\alpha_d$
0	0.5	$0.5(1 - 0.5\alpha_d)$	$0.25\alpha_d$
1	0.5	$0.5(1 - 0.5\alpha_d)$	$0.25\alpha_d$

differential privacy for f_i, and assume that K_{f_i} is independent from K_{f_j} for any $i \neq j$. Then $(K_{f_1}, \ldots, K_{f_n})$ provides $\max\{\varepsilon_i\}$-differential privacy for $(f_1, \ldots, f_2)$.

Proof. Let D' be a data set obtained from D by adding or removing a single user. We want to check that the following inequalities hold for any subset S of the range of $(K_{f_1}, \ldots, K_{f_n})$:

$$e^{-\max\{\varepsilon_i\}} \leq \frac{P((K_{f_1}(D), \ldots, K_{f_n}(D)) \in S)}{P((K_{f_1}(D'), \ldots, K_{f_n}(D')) \in S)} \leq e^{\max\{\varepsilon_i\}}$$

It is easy to show that the above inequality holds for the case of S being the Cartesian product of sets S_i, with S_i a subset of the range of $K_{f_j}(D)$, or when the probability distribution of $(K_{f_1}, \ldots, K_{f_n})$ is absolutely continuous. For a general set S and a non absolutely continuous distribution, the inequalities still hold. However, such a general proof requires the use of some concepts of measure theory and, for space reasons, we omit such details here. We will show that the inequalities hold for the case of $S = S_1 \times \ldots \times S_n$.

The probabilities $P((K_{f_1}(D), \ldots, K_{f_n}(D)) \in S)$ and $P((K_{f_1}(D'), \ldots, K_{f_n}(D')) \in S)$ can be written as the product of probabilities $\prod P(K_{f_i}(D) \in S_i)$ and $\prod P(K_{f_i}(D') \in S_i)$, respectively. By adding or removing a single individual, only one of the queries is affected. Say the affected query is f_j for some $j \in \{1, \ldots, n\}$. By removing the factors that are both in the numerator and the denominator, the inequalities that we need to check become

$$e^{-\max\{\varepsilon_i\}} \leq \frac{P(K_{f_j}(D) \in S_j)}{P(K_{f_j}(D') \in S_j)} \leq e^{\max\{\varepsilon_i\}},$$

which holds because K_{f_j} satisfies ε_j-differential privacy, and $\varepsilon_j \leq \max\{\varepsilon_i\}$. □

Protocol 1 Interactive Laplace noise addition mechanism

(1) The database holder initializes the access mechanism with the following parameters:
   - ε, the maximum level of leakage allowed;
   - λ, the amount of noise to be added to every response (λ is the parameter of the Laplace noise distribution); for fixed ε, the greater λ, the more queries the access mechanism will be able to answer.

(2) Let $i := 1$.

(3) **while** queries are answered by the access mechanism **do**
   (a) The user submits a query f_i (for $i > 1$, f_i may depend on responses to previous queries $(f_1, \ldots, f_{i-1})$).
   (b) **if** $\Delta(f_1, \ldots, f_i)/\lambda \leq \varepsilon$ **then** the access mechanism returns $f_i(D) + \text{Laplace}(\lambda)$ as response; **else** it returns nothing.
   (c) $i := i + 1$

5. Interactive Queries and Adaptive Attacks

Differential privacy is usually presented as an interactive query-response mechanism where the data set is held by a trusted party to whom users send their queries. Despite this claimed interactivity, the formal definition of differential privacy (Definition 1) is based on a single query, thereby removing the complexities that interactivity would introduce. Malicious users may try to use interaction to exploit potential vulnerabilities of the access mechanism. When using Laplace noise addition the user can, for example, use the knowledge acquired from previous answers to forge the new query. For knowledge refinement the problem is even more compelling, since, besides the query function, the user also feeds the access mechanism with a prior knowledge distribution and optionally with a distance function.

5.1. *Interactive access mechanisms*

To implement interactivity, a protocol is built on top of the non-interactive access mechanism. The idea is quite simple; when a query is submitted, the access mechanism analyzes if answering the query is too disclosive, in which case the query is simply discarded. To determine if answering a new query is too disclosive, all the queries submitted by a user so far, including the new query, are treated as a single multicomponent query and ε-differential privacy is enforced for it. Protocol 1 describes the protocol for the interactive Laplace noise access mechanism introduced in Ref. 1; in the protocol, $\Delta(\cdot)$ stands for sensitivity.

We now present an interactive knowledge refinement mechanism parallel to the Laplace-based one. As knowledge refinement does not depend on the sensitivity of the query function, our interactive mechanism does not need to compute sensitivities and is therefore simpler than the Laplace-based one. Also, we will allow the database user to select the amount of leakage ε_i independently for each query f_i. The only requirement is that the access mechanism will refuse answering query f_i (and successive queries) if the leakage of the multicomponent query $(f_1, \ldots, f_i)$ exceeds ε.

Protocol 2 Interactive mechanism for knowledge refinement

(1) The database holder initializes the access mechanism with ε, the maximum level of leakage allowed.
(2) Let $i := 1$.
(3) **while** queries are answered by the access mechanism **do**
   (a) The user submits a query $q_i = (f_i, P_{f_i}, d_i, \varepsilon_i)$, where f_i is the query function, P_{f_i} is the prior knowledge distribution for the query, d_i is the distance function to be used and ε_i is the desired level of leakage (for $i > 1$, q_i may depend on responses to previous queries $(q_1, \ldots, q_{i-1})$).
   (b) **if** $\sum_{j=1}^{i} \varepsilon_j \leq \varepsilon$ **then** the access mechanism returns a response to f_i resulting from applying knowledge refinement to P_{f_i} with distance d_i so that ε_i-differential privacy is guaranteed; **else** it returns nothing.
   (c) $i := i + 1$

If the d-th query is the last query answered by the interactive mechanism of Protocol 2, by construction the user obtains at most a knowledge gain ε for $(f_1, \ldots, f_d)$. This holds regardless of the prior knowledge distributions and distance functions chosen by the user for each query.

By submitting the desired level of leakage ε_i for each query, in Protocol 2 the database user is allowed to trade more accurate answers in some queries for less accurate answers in other queries. Protocol 1 could be modified to permit such flexibility as well: the user could be asked to choose the noise parameter λ_i for the i-th query, and the condition checked by the access mechanism would become

$$\sum_{j=1}^{i} \Delta(f_j)/\lambda_j \leq \varepsilon\,.$$

Since $\Delta(f_1, \ldots, f_i) \leq \Delta(f_1) + \ldots + \Delta(f_i)$, when $\lambda_1 = \ldots = \lambda_i$ the modified condition above may result in less queries being answered than the condition in Protocol 1.

5.2. *Adaptive attacks*

The interactive mechanisms of Protocols 1 and 2 guarantee, respectively for Laplace noise and knowledge refinement, that the responses to any sequence of adaptive queries $(q_1, \ldots, q_d)$ will not violate ε-differential privacy. However, the following question can be raised: is there any sequence of adaptive queries $(q_1, \ldots, q_d)$ and a way to combine the responses to this sequence that allows an attacker to obtain an estimator of $f(D)$ that does not satisfy ε-differential privacy?

We show that such an attack cannot succeed. Our proof is completely general; it does not depend on the access mechanism used to attain differential privacy. Let $F : \mathbb{R}^d \to \mathbb{R}$ be the function used by the attacker to combine the responses to $q_1, \ldots, q_d$; let these responses be samples of the random vector $K_{f_1}(D), \ldots, K_{f_d}(D)$. The attacker computes $F(K_{f_1}(D), \ldots, K_{f_d}(D))$ and takes it as the response to $f(D)$. We are not interested in

determining F or even in determining whether $F(K_{f_1}(D), \ldots, K_{f_d}(D))$ is a good estimate for $f(D)$. The following result will suffice.

Proposition 4. *For any function F, if $(K_{f_1}(D), \ldots, K_{f_d}(D))$ satisfies ε-differential privacy, then $F(K_{f_1}(D), \ldots, K_{f_d}(D))$ also satisfies ε-differential privacy.*

Proof. We need to check that, for each pair of data sets D and D' that differ in a single individual and for each set $S \in Range(F(K_{f_1}, \ldots, K_{f_d}))$, it holds that

$$\frac{P(F(K_{f_1}(D), \ldots, K_{f_d}(D))) \in S)}{P(F(K_{f_1}(D'), \ldots, K_{f_d}(D'))) \in S)} \leq e^{\varepsilon}.$$

Since $P(F \circ X \in S) = P(X \in F^{-1}(S))$, we can express the previous inequality as

$$\frac{P((K_{f_1}(D), \ldots, K_{f_d}(D)) \in F^{-1}(S))}{P((K_{f_1}(D'), \ldots, K_{f_d}(D')) \in F^{-1}(S))} \leq e^{\varepsilon}$$

which holds because $(K_{f_1}(D), \ldots, K_{f_d}(D))$ satisfies ε-differential privacy. □

The following corollary follows from the previous proposition.

Corollary 1. *Whatever the attacker's strategy, her estimate for $f(D)$ always satisfies ε-differential privacy.*

6. Quality of the Response to Individual Queries

We have defined an individual query, f, to be one that depends on a single individual. We can think of it as a query that returns the value of some attribute for some specific individual.

Typical differential privacy mechanisms based on noise addition provide low data quality responses for individual queries. The reason is that, as any individual can take any value in $Range(f)$, the sensitivity of the query equals the length of $Range(f)$. When using knowledge refinement, the quality of the response depends to a great extent on the prior knowledge available.

In this section, we provide some data quality comparisons between Laplace noise addition and knowledge refinement for individual queries. Comparisons will be based of specific query functions. The first one is based on a query function that returns a Boolean value; we show how the distribution for the differentially private response gets closer to the real response by refining prior knowledge than by adding Laplace noise. The second comparison is based on a continuous function with range $[0, 1]$; we show that, even if we have no prior knowledge, knowledge refinement provides better data quality for individual queries.

6.1. *Data quality for a Boolean attribute*

Consider a simple database D with two attributes: an identifier ID and a Boolean attribute B that may take values 0 and 1. We assume that B is very sensitive and that, to limit the

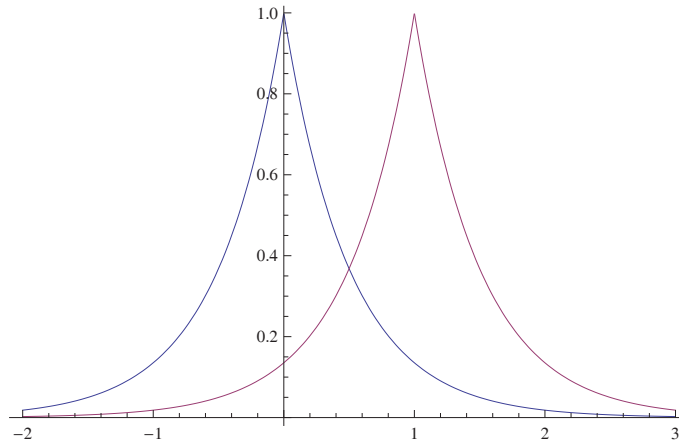

Fig. 5. Response distributions with Laplace noise addition.

disclosure risk, access to the database must be mediated by a query-response mechanism satisfying differential privacy, with $\varepsilon = 1$. Let $f : \mathcal{D} \to \{0, 1\}$ be a query that asks the value of attribute B for a specific individual.

To achieve differential privacy via Laplace noise addition, we must first compute the sensitivity of function f. Assuming that f returns $\frac{1}{2}$ if the individual is not in the database, the L_1-sensitivity of f is $\frac{1}{2}$. Therefore, to achieve differential privacy for $\varepsilon = 1$, we must add a Laplace distribution $L(0, \frac{1}{2})$ to the true value of the query response. Figure 5 shows the distribution of the responses for both possible values of B, 0 and 1.

Assuming that the user is only interested in a 0/1 response, any value below $\frac{1}{2}$ is taken as 0, and any value above $\frac{1}{2}$ as 1. The distribution for the response thus obtained is:

$$K_f(D) = \begin{cases} 0 & f(D) + L(0, \frac{1}{2}) < 0.5 \\ 1 & \text{otherwise.} \end{cases}$$

If $f(D)$ equals 0, $K_f(D)$ follows a Bernoulli distribution with parameter 0.184. If $f(D)$ equals 1, the distribution of $K_f(D)$ is a Bernoulli with parameter 0.816. Note that this is completely independent from the true distribution of attribute B, and from any previous knowledge that the user might have on it. Hence, differential privacy via Laplace noise addition does not let the user exploit prior knowledge.

Let us assume that attribute B is 1 only with probability 0.01. For a user with this information, using the response obtained from the differential privacy mechanism is actually misleading, as the result will be 1 with probability

$$\begin{aligned} P(K_f(D) = 1) &= P(K_f(D) = 1 | f(D) = 0) P(f(D) = 0) + P(K_f(D) \\ &\quad = 1 | f(D) = 1) P(f(D) = 1) = 0.184 \cdot 0.99 + 0.816 \cdot 0.01 = 0.19\,. \end{aligned}$$

We could increase the parameter ε to get a more accurate response. However, by doing so we would be reducing the privacy guarantees.

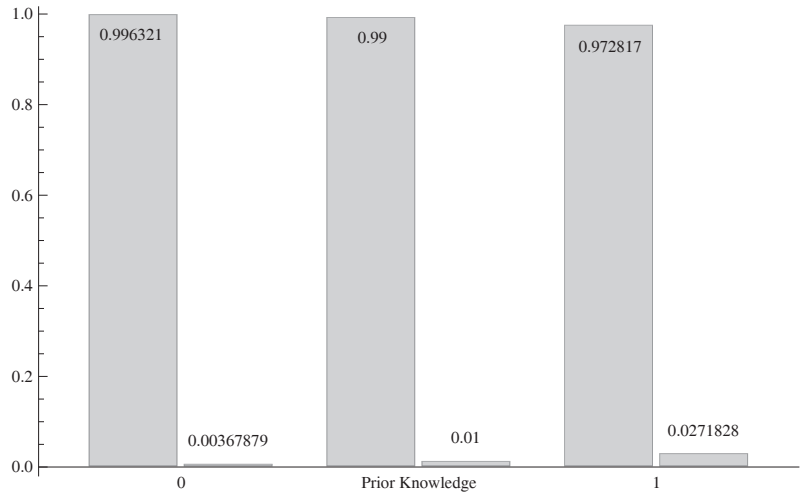

Fig. 6. Response distribution with prior knowledge refinement.

Now, we turn to the refinement mechanism and, same as before, we assume that the user knows that B equals 1 with probability 0.01. Take $\alpha_u = e^{\varepsilon} = e$ and $\alpha_d = e^{-\varepsilon} = e^{-1}$. Hence,

$$
\begin{aligned}
&P(K_f(D) = 1 | f(D) = 0) = P(f(D) = 1) \cdot \alpha_d = 0.003678 \\
&P(K_f(D) = 0 | f(D) = 0) = 1 - 0.003678 = 0.9963222 \\
&P(K_f(D) = 1 | f(D) = 1) = P(f(D) = 1)) \cdot \alpha_u = 0.027182 \\
&P(K_f(D) = 0 | f(D) = 1) = 1 - 0.027182 = 0.972817\,.
\end{aligned}
$$

Note that, as this is not an absolutely continuous distribution, we had to do some adjustment to have a total probability mass equal to one: instead of adjusting α_u and α_d, we directly adjusted $P(K_f(D) = 0 | f(D) = 0)$ and $P(K_f(D) = 0 | f(D) = 1)$. Figure 6 depicts the distribution of the response for both possible values of attribute B and for the prior knowledge.

Now, the probability of obtaining a response 1 is

$$
\begin{aligned}
P(K_f(D) = 1) = {} & P(K_f(D) = 1 | f(D) = 0) P(f(D) = 0) + P(K_f(D) = 1 | f(D) \\
& = 1) P(f(D) = 1) = 0.003678 \cdot 0.99 + 0.02182 \cdot 0.01 = 0.003912\,.
\end{aligned}
$$

As 0.003912 is much closer to 0.01 than 0.19, we conclude that, despite both mechanisms providing the same level of privacy, the output distribution is much closer to the actual distribution of the attribute when using the mechanism based on knowledge refinement. Therefore, knowledge refinement outperforms Laplace noise addition for Boolean attributes released under differential privacy.

Table 5. Comparison between the distribution of the response to $f(D)$ for Laplace noise addition and knowledge refinement for several values of ε when $f(D) = 0.5$.

	Laplace noise addition		Knowledge refinement		
ε	Variance	$P(K_f(D) \in [0,1])$	Variance	$size(\mathcal{U}_u)$	$P(K_f(D) \in \mathcal{U}_u)$
0.1	200	0.476	0.077	0.475	0.525
$\ln(2)$	4.16	0.549	0.046	0.333	0.667
1	2	0.607	0.034	0.269	0.731
2	0.5	0.684	0.012	0.119	0.881

6.2. *Data quality for a continuous attribute*

Let $f : \mathcal{D} \to [0,1]$ be a query function that returns a value in the interval $[0,1]$. We have fixed the range of f to be able to obtain some numerical results, but a similar comparison can be done for other ranges. We compare the response obtained by using Laplace noise addition and knowledge refinement with a uniform $U[0,1]$ prior knowledge.

When using Laplace noise addition, the response to $f(D)$ is $K_f(D) = f(D) + Laplace(0, 1/\varepsilon)$. When using knowledge refinement, the prior knowledge is modified by increasing the probability of the set $\mathcal{U}_u$ containing the points closer to $f(D)$ by a factor α_u, and decreasing the probability of the rest by a factor α_d. We saw in Sec. 3 that $\mathcal{U}_u$ must satisfy $P_f(\mathcal{U}_u) = (\alpha_u - 1)/(\alpha_u - \alpha_d)$, which in the case of a uniform prior knowledge within the interval $[0,1]$ coincides with the size of $\mathcal{U}_u$. We also saw (Table 2) that, for an individual query, the factors are $\alpha_u = e^{\varepsilon}$ and $\alpha_d = e^{-\varepsilon}$.

Table 5 shows a comparison of the distribution for the response to $f(D)$ for several values of ε when $f(D) = 0.5$. For Laplace noise addition, we have computed the variance of the response, as well as the probability for the response to be within the range $[0,1]$. For knowledge refinement, we have computed the variance of the response, the size of $\mathcal{U}_u$, and the probability for the response to be in $\mathcal{U}_u$. The results in the table show that knowledge refinement behaves much better than Laplace noise addition, but perhaps this is better observed by comparing the actual distributions. Figure 7 shows the distributions for the response when using Laplace noise addition and knowledge refinement with the same values of ε used in the table.

7. Discussion

In previous sections we have highlighted that the knowledge refinement mechanism lets the database user exploit her prior knowledge to obtain a more accurate response. In Sec. 6 we saw that, for the case of individual queries, knowledge refinement provides a much more accurate response even when there is no prior knowledge.

Other advantages of prior knowledge refinement are:

- *Simplicity*. Mechanisms such as Laplace noise addition are based on the addition of a random noise whose magnitude depends on the variation of the query function across neighbor data sets, also known as sensitivity. To calibrate the random noise, the sensitivity of the function must be computed, which may be quite complex. The mechanism

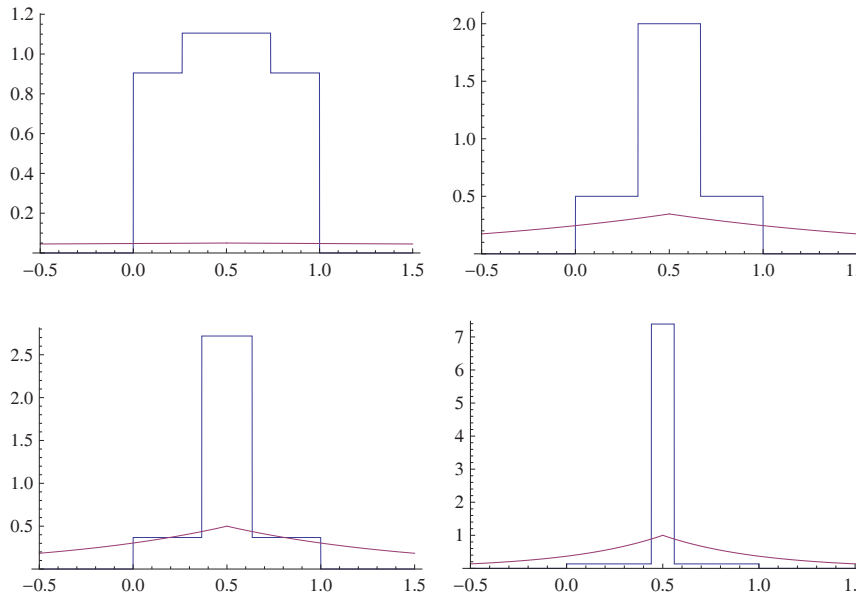

Fig. 7. Distribution for the response to $f(D)$, when $f(D) = 0.5$, for Laplace noise addition (distribution with unbounded support) and knowledge refinement (distribution with support $[0, 1]$) for $\varepsilon = 0.1$ (top left), $\varepsilon = \ln(2)$ (top right), $\varepsilon = 1$ (bottom left), and $\varepsilon = 2$ (bottom right).

based on the refinement of the prior knowledge only depends on the prior knowledge (it is independent from the sensitivity of the query function), and thus it is easier to implement, especially in a non-supervised environment.

- *Generality*. As said above, Laplace noise addition requires computing the sensitivity of the query function, and this can only be done if the query function takes values in a metric space. This introduces some complexities when the function returns categorical information. The mechanism based on prior knowledge refinement does not impose any requirement on the query function, and thus it can be applied without extra overhead to functions returning categorical information.
- *Consistency*. Knowledge refinement lets the database user easily restrict the response to a set of values consistent with the query function, by having the prior knowledge assign a probability mass of zero to the set of inconsistent values. For example, in Table 5 we saw that Laplace noise sends the response outside the query function range $[0, 1]$ with great probability, while knowledge refinement always keeps the response within range. Querying categorical attributes is another example. It is usual to have some combinations of categories that do not make sense. For example, if the attributes are "employed" (Y/N), and "unemployment benefits" (Y/N), a response Y for both attributes does not make sense. When using a noise addition mechanism, there is no way to avoid that combination of values, while, when using knowledge refinement, to avoid that combination we only have to use a prior knowledge distribution that assigns zero probability mass to it.

Despite the advantages listed above, there are some situations for which the proposed mechanism is not appropriate. If the range of values that the function may return is large compared to the variability between neighbor data sets, and the database user does not have precise knowledge of the response, then a method based on noise addition produces better data quality. This may be the case of statistical queries where the user has no prior knowledge of the result. However, when querying about a specific individual, the proposed method results in much greater response quality.

8. Conclusions

We have introduced a novel mechanism to attain differential privacy. This mechanism is based on refining the prior knowledge that the user may have about the query response. This refinement is performed taking into account the constraints imposed by differential privacy.

The refinement mechanism presents several advantages over the usual noise addition mechanism. It is easier to implement, especially in a non-supervised environment, as it does not require potentially complex computations (such as determining the sensitivity of the query function). The fact that it lets users exploit their prior knowledge may lead to a level of data quality not reachable by mechanisms independent of the user knowledge. For example, we showed in the examples of Sec. 6 that the distribution of the response was closer to the real distribution when using the refinement mechanism. For query functions with great sensitivity, the amount of noise added by noise addition mechanisms, such as Ref. 1, may render the response useless. In contrast, the data quality that results from our proposal is independent from the sensitivity of the query function; yet this has the drawback that, for small sensitivities, our approach may be inferior to noise addition.

We have also analyzed the behavior of our approach for multicomponent queries. A generic property of differential privacy guarantees that, if a ε_i-differentially private response is provided for a query f_i, for $i = 1$ to n, a $\sum \varepsilon_i$-differentially private response is provided for the query $(f_1, \ldots f_n)$. We have seen that this can be improved if each query f_i refers to a disjoint set of individuals. In this case, we achieve $\max\{\varepsilon_i\}$-differential privacy, instead of $\sum \varepsilon_i$-differential privacy.

Interactive mechanisms for Laplace noise addition and knowledge refinement have also been described. Such interactive mechanisms take as input parameter the maximum level of leakage ε allowed by the database holder, and queries are answered until that level of leakage is reached. The knowledge refinement interactive mechanism is superior to the Laplace noise interactive mechanism in that it does not need to compute sensitivities. We have shown that any interactive mechanism providing ε-differential privacy is safe against adaptive attacks; whatever the strategy used by an attacker to combine query responses, ε-differential privacy holds.

Acknowledgments and Disclaimer

This work was partly supported by the Government of Catalonia under grant 2009 SGR 1135, by the Spanish Government through projects TSI2007-65406-C03-01

“E-AEGIS”, TIN2011-27076-C03-01 “CO-PRIVACY” and CONSOLIDER INGENIO 2010 CSD2007-00004 “ARES”, and by the European Comission under FP7 projects “DwB” and “Inter-Trust”. The second author is partially supported as an ICREA Acadèmia researcher by the Government of Catalonia. The authors are with the UNESCO Chair in Data Privacy, but they are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization.

References

1. C. Dwork, F. McSherry, K. Nissim and A. Smith, Calibrating noise to sensitivity in private data analysis, in *Proc. 3rd Theory of Cryptography Conf.* (*TCC 2006*), LNCS 3876, Springer, 2006, pp. 265–284.
2. C. Dwork, Differential privacy, in *Automata, Languages and Programming* (*ICALP 2006*), LNCS 4052, Springer, 2006, pp. 1–12.
3. K. Nissim, S. Raskhodnikova and A. Smith, Smooth sensitivity and sampling in private data analysis, in *Proc. 39th annual ACM Symp. Theory of Computing* (*STOC 2007*), ACM, 2007, pp. 75–84.
4. F. McSherry, Mechanism design via differential privacy, in *Proc. 48th Annual Symp. Foundations of Computer Science* (*FOCS 2007*), IEEE Computer Society, 2007, pp. 94–103.
5. B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F.McSherry and K. Talwar, Privacy, accuracy, and consistency too: a holistic solution to contingency table release, in *Proc. 26th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems* (*PODS 2007*), ACM, 2007, pp. 273–282.
6. X. Xiao, G. Wang and J. Gehrke, Differential privacy via wavelet transforms, *IEEE Trans. Knowledge and Data Engineering* **23**(8) (2011) 1200–1214.
7. A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke and L. Vilhuber, Privacy: from theory to practice on the map, in *Proc. ICDE 2008*, IEEE, 2008, pp. 277–286.
8. J. Abowd and L. Vilhuber, How protective are synthetic data?, in *Privacy in Statistical Databases* (*PSD 2008*), LNCS 5262, Springer, 2008, pp. 239–246.
9. A. Charest, How can we analyze differentially-private synthetic datasets?, *J. Privacy and Confidentiality* **2**(2) (2010).
10. A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, E. Schulte Nordholt, K. Spicer and P.-P. de Wolf, *Statistical Disclosure Control* (Wiley, 2012).
11. J. Soria-Comas and J. Domingo-Ferrer, Differential privacy through knowledge refinement, in *2012 ASE/IEEE Int. Conf. Privacy, Security, Risk and Trust (PASSAT 2012)*, Amsterdam, The Netherlands, September 3–6, 2012.
12. R. Sarathy and K. Muralidhar, Evaluating Laplace noise addition to satisfy differential privacy for numeric data, *Trans. Data Privacy* **4**(1) (2011) 1–17.
13. C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov and M. Naor, Our data, ourselves: privacy via distributed noise generation, in *Advances in Cryptology* (*EUROCRYPT 2006*), LNCS 4004, Springer, 2006, pp. 486–503.