Non-Interactive Differential Privacy: a Survey [*]

David Leoni
LINA lab, University of Nantes
david.leoni@etu.univ-nantes.fr

ABSTRACT

OpenData movement around the globe is demanding more access to information which lies locked in public or private servers. As recently reported by a McKinsey publication, this data has significant economic value, yet its release has potential to blatantly conflict with people privacy. Recent UK government inquires have shown concern from various parties about publication of anonymized databases, as there is concrete possibility of user identification by means of linkage attacks. Differential privacy stands out as a model that provides strong formal guarantees about the anonymity of the participants in a sanitized database. Only recent results demonstrated its applicability on real-life datasets, though. This paper covers such breakthrough discoveries, by reviewing applications of differential privacy for non-interactive publication of anonymized real-life datasets. Theory, utility and a data-aware comparison are discussed on a variety of principles and concrete applications.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.2 [**database Management**]: Database Applications—*Statistical databases*

General Terms

Privacy-Preserving Data Publishing

Keywords

Anonymization, Differential privacy, Survey, Open Data

1. INTRODUCTION

1.1 Motivation

In a recent report by McKinsey [47] it is estimated that in the developed economies of Europe alone, government administration could save more than €100 billion ($149 billion) in operational efficiency improvements alone by leveraging big data. This term refers to the enormous quantity of information organizations around the globe collect daily. In particular, public institutions retain data about many aspects of our life, including medical, fiscal, transportation and criminal records. Private companies are also increasingly taking a bigger role in our private life by recording our Internet searches, friends network, financial transactions and transportation habits. Not everybody knows how to handle this information properly, though. UK, the leading European country in terms of Open Data, recently held a consultation [56] with public institutions and industry representatives to discuss data publishing issues. Various parties expressed a clear concern about privacy issues, prompted in part by clamorous episodes of privacy breaches occurred in the past. In 1997 the state of Massachusetts had to provide health insurance to its employees, but insurance companies wanted some information about employees' health status. It was decided to provide such data with supposedly 'anonymized' health records of the personnel. Obviously identifying information such as names and addresses were stripped, but other fields such as ZIP code, birth date and sex were kept. Unfortunately, these fields where also present in the voting records, which are public in the US. See Figure 1 for a graphical representation of these two datasets. Latanya Sweeney, then a computer science student, decided to prove privacy was at risk by crossing the data and locating personal information of William Weld, the then Governor of Massachusetts. She obtained a copy of the voting records of his constituency, and discovered there was only one person in the health records with the same ZIP, birthday and sex of

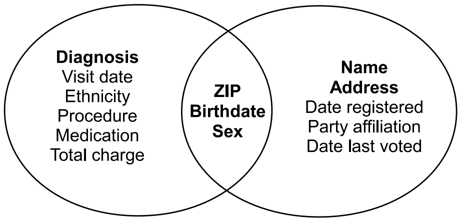


Figure 1: Crossing datasets in Sweeney case.

[*] This is the extended version of the paper presented at WOD'12 Conference held in Nantes

Pre-print
WOD '12 May 25 2012, Nantes, France

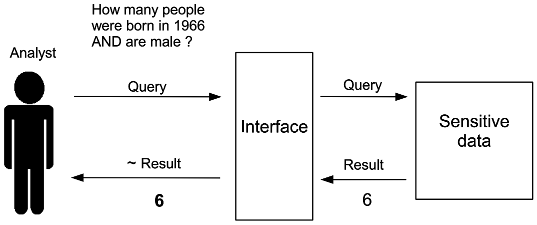


Figure 2: A generic counting query.

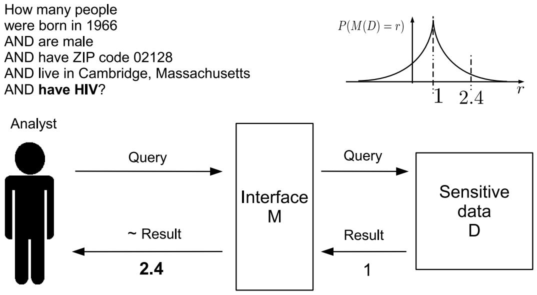


Figure 3: A sensitive query.

the governor. To prove her point, she then sent to Governor Weld his own health records. In addition to this, Sweeney also proved 87% of American citizens can be uniquely identified just by knowing their gender, ZIP code and birth date [55]. The quest for true anonymization in the field of Privacy Preserving Data Publishing (PPDP) began, and it is still not over. In 2006 Internet provider AOL released its search log containing 3 months of searches of 650,000 users. Usernames were masked with random identifiers, still, in a matter of days, a New York Times reporter identified Thelma Arnold, a 62-year old widow from Lilburn, GA as user # 4417749 [3], and her queries became known to the world. As a consequence of releasing this private dataset the CTO of AOL resigned, two employees were fired and a class action lawsuit is pending. Later the same year, Netflix, a DVD rental company released a perturbed version of one tenth of its database of movie ratings expressed by its customers. A prize of 1,000,000$ was offered to whoever improved by 10% the accuracy of the company's own recommandation algorithm. The following year the researchers Narayanan and Shmatikov proved it was possible to identify users by linking them to Imdb, a public database of movie ratings in which users voluntarily can publish their ratings [52]. This concerns prevented in 2010 NetFlix from proposing a follow-up of the prize.

1.2 Solutions

1.2.1 Interactive vs Non-Interactive

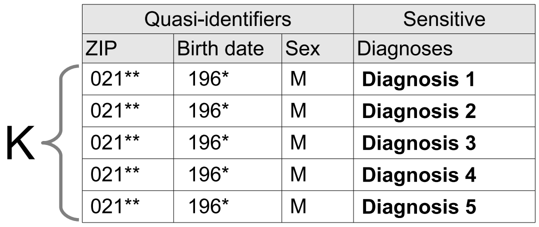
Analysts want to have precise answers to queries about anonymized data, which can be sensitive. In the so-called *interactive* setting, information is protected inside a database handled by the data owner, and access to it is allowed only through an interface. Answers provided by the interface are processed in such a way to guarantee the anonymity of the participants in the database. Let's suppose analysts are only allowed to ask counting queries. We can see in Figure 2 an example of a query request by the analyst. In this case the returned answer can be the true value in the database as the interface deems the query to be generic enough not to compromise privacy of individuals. On the other hand, in Figure 3 we see a situation in which the asked query is much more precise and sensitive. In this case the system could completely deny an answer, but this appears to be quite a drastic remedy. A more appealing solution could be to add some random noise to the true count, maybe distributed according to the Laplace distribution, in order to have most often results near the true value. This way the analyst would be provided with an idea of the original amount while still having some uncertainty about it. Still, we could have a problem. What if three different malicious analysts asked all the same question to the interface? Each of them would receive a slightly differing answer because of the random noise being added to the true count. In the worst case scenario, they could exchange the values they got and calculate an average. This operation in expectation would allow them to deduce a value much closer to the true one and would thus vanish the efforts of the anonymizing interface. In the *non-interactive* setting this problem is addressed by releasing once and for all the data which we think is of interest to most analysts, while still preserving privacy. Naturally the example we made is simplistic and with this paper we intend to prove a wealth of useful information can be published while formally maintaining strong privacy guarantees.

1.2.2 What is 'personal information'?

Over the years, several solutions to solve the problem of protecting privacy in anonymized databases have been proposed. Examples are k-anonymity [55], l-diversity [46], t-closeness [42]. All these methods suppose it is worth to distinguish data attributes into these groups: identifiers (i.e. name, surname), quasi-identifiers(i.e. ZIP code, gender, age) and sensitive (i.e. diagnosis, rentedAdultMovie). In legal terms, in the EU the Data Protection Directives [27] define personal data as 'information related to an identified or identifiable natural person'. It is a quite general definition, and for example even a house value can be classified as personal information as it might reveal its owner income. Recently, the European Data Protection Supervisor EDPS expressed its concerns [26] about a proposal on re-use of Public Sector Information (PSI) previously adopted by the European Commission [25]. In particular it was recommended that

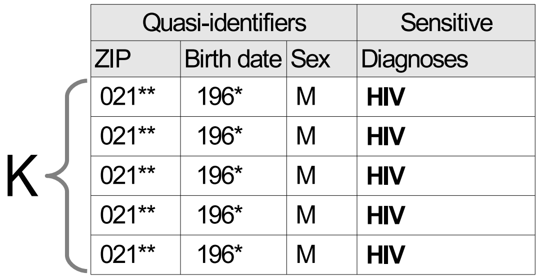
> "Where appropriate, the data should be fully or partially anonymised and license conditions should specifically prohibit re-identification of individuals and re-use of personal data for purposes that may individually affect the data subjects."

The *purpose limitation* is a difficult issue to solve in a context where PSI is put on the Internet for everybody to see. European transgressors who try to identify persons whose data is contained in a published anonymized dataset may be fined, but how to deal with non-European ones? Also, how is it possible to measure the degree of anonymization of a given dataset in order to decide if it is too risky to be published on the Internet? For example, the UK Office for National Statistics is going to release data collected in 2011 anonymized with a record-swapping system [57], which involves selecting households which are deemed too identifiable and swapping them with other households which are not too far in the same geographical region and have similar



Quasi-identifiers			Sensitive
ZIP	Birth date	Sex	Diagnoses
021**	196*	M	**Diagnosis 1**
021**	196*	M	**Diagnosis 2**
021**	196*	M	**Diagnosis 3**
021**	196*	M	**Diagnosis 4**
021**	196*	M	**Diagnosis 5**

Figure 4: Anonymized dataset under k-anonymity



Quasi-identifiers			Sensitive
ZIP	Birth date	Sex	Diagnoses
021**	196*	M	**HIV**
021**	196*	M	**HIV**
021**	196*	M	**HIV**
021**	196*	M	**HIV**
021**	196*	M	**HIV**

Figure 5: Problem with anonymized dataset under k-anonymity

values. Tables containing origin-destination data are considered too hard to anonymize in a satisfying way so they are licensed only to restricted users. What are the theoretical basis for this distinction, if any? The EDPS calls for a 'proactive approach' which should be taken by authorities, meaning privacy issues should be analyzed at the earliest stages and involved people informed throughout all the data process release.

1.2.3 From k-anonymity to differential privacy

Linkage attacks shown before demonstrate how quasi-identifiers can be used to significantly increase the accuracy in identity disclosure, making the distinction with identifiers purely artificial. In the already mentioned Sweeney case, where Massachusetts Governor health data was revealed by crossing public voting records and anonymized health data, Latanya Sweeney proposed the so-called k-anonymity model to prevent disclosure attacks. We will show an example of anonymization through k-anonymity and why it may still fail to work under certain circumstances. Let's suppose we release a medical records table like the one in Figure 4. In this table quasi-identifiers have been generalized in such a way to have at least k rows in the database with the same quasi-identifiers. This would prevent an eventual attacker to discover exactly which diagnosis the governor has among the (at least) k available. Still, this model has a problem, exemplified in Table **??**.

What could happen if by chance all the people in Governor's group had HIV? We would conclude the Governor himself is affected by this illness, and thus his privacy would be compromised. Also, sometimes the sole fact of knowing somebody is or is not in a database may provide a malicious user with valuable information to carry out an attack. So, how do we reach the so called *privacy by design*, when a data release process is devised to prevent disclosure with formal guarantees? To respond to these issues the concept of *differential privacy* was introduced by Dwork [18] to prevent attackers from being capable even to detect the presence or absence of a given person in a database. Differential privacy falls in the category of so called perturbative methods, which attempts to create uncertainty in the released data by adding some random noise. If database participants are independent from each other, differential privacy promises that even if an attacker knows everything about every user in the db but one, by looking at the published statistics he won't be able to determine the identity of the remaining individual. Kieron O'Hara, in his 2011 independent transparency and privacy review to UK government [53] mentions differential privacy as a cutting-edge technology that judges the *computation* of the anonymization algorithm as privacy-preserving or otherwise, rather than trying to make an impossible distinction between identifying and non-identifying *data*. This might sound promising, but O'Hara claims differential privacy appears to be limited to the interactive setting. Is this really true? Recent results in the non-interactive setting are encouraging. In what follows, we formalize some concepts about differential privacy.

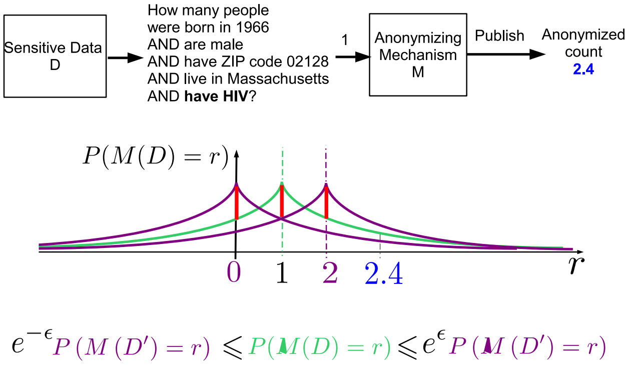


Figure 6: A sensitive count probability for neighboring databases.

1.3 Basic definitions

We use $P(A)$ to indicate the probability of the occurrence of event A and define $\|x\|_1$ as the sum of all elements in vector x.

DEFINITION 1 (DATABASE). *Given a database universe $\mathcal{D}$ we define a database $D \in \mathcal{D}$ as multiset of $|D|$ tuples from a universe $\mathcal{U}$ of people. Each person has h attributes $A_1, A_2, ..., A_h$. We say two databases D_1, D_2 are neighbors if they differ in one tuple. We indicate such condition as $|D_1 \Delta D_2| = 1$, where $D_1 \Delta D_2 = (D_1 \cup D_2) \setminus (D_1 \cap D_2)$.*

1.4 Differential privacy

Randomized algorithms to publish sensitive data are called mechanisms. Since we are addressing the problem of statistical disclosure at large, we use $\mathcal{R}$ to denote a wide range of output possibilities for the mechanism designers, whose goal is to devise a mechanism function $\mathcal{D} \rightarrow \mathcal{R}$. One possible choice of $\mathcal{R}$ could be $\mathcal{D}$ itself, meaning we are going either to release a new database composed by synthetic individuals who hopefully follow the same distribution of the original participants or we publish a perturbed version of the original database, with real data randomly modified to satisfy differential privacy criteria. An another possible and popular choice of $\mathcal{R}$ is the set of queries q_j counting how many individuals u_i satisfy a given property $\gamma_j(u_i)$. A mechanism

Table 1: e^{ϵ} values

ϵ	e^{ϵ}
0.01	1.01
0.1	1.10
$\ln 2 = 0.69$	2
1	2.71
$\ln 3 = 1.10$	3

in order to be ϵ-differentially private must satisfy the following definition first introduced by Dwork [20], which in recent years has become popular among researchers in the field of statistical disclosure:

Definition 2 (ϵ-dp). *Given a randomized mechanism $M : \mathcal{D} \rightarrow \mathcal{R}$ and a real value $\epsilon > 0$, we say M satisfies ϵ-differential privacy if $\forall D_1, D_2 \in \mathcal{D}$ such that $|D_1 \Delta D_2| = 1$ and $\forall R \subseteq \mathcal{R}$ the following equation holds:*

$$P\left(M\left(D_1\right) \in R\right) \leqslant e^{\epsilon} P\left(M\left(D_2\right) \in R\right)$$

Equivalently, as we can exchange the two databases D_1 and D_2 we can write the following equation:

$$e^{-\epsilon} P\left(M\left(D_2\right) \in R\right) \leqslant P\left(M\left(D_1\right) \in R\right) \leqslant e^{\epsilon} P\left(M\left(D_2\right) \in R\right)$$

Differential privacy guarantees the following: a data release mechanism is ϵ-differentially private if, for any couple of neighboring databases D_1 and D_2 differing in one person, any participant u in the database, and any possible output r of the release mechanism, the presence or absence of a participant u (in db terms, D_1 and D_2 differing in one row) causes at most a multiplicative e^{ϵ} change in the probability of the mechanism outputting r.

1.4.1 Differential privacy for counting queries

Suppose we want to release the count of people with HIV from a hypothetical medical database D. We must then devise a mechanism $\mathcal{L}$ that when executed on databases differing in one person probably outputs the same count. Differential privacy is a constraining model but it still allows us to have good outputs close to the true count at a rate exponentially greater than values far from it. For counting queries, if a database D_2 differs in one person from D_1 then there are three possibilities: the count can remain the same, or differ from the original database count by $+1$ or -1. It turns out that if we add to the true value noise distributed according to the Laplace distribution differential privacy is satisfied, and in Figure 6 we can see an example of it. We included in the picture three times the same distribution, one for the database D at hand centered around 1, the supposedly true count, and two other distributions for possible neighboring databases D', (of course there are neighboring databases with the same count as D). Differential privacy constraints will guarantee that the distance between the distributions, evidenced by the vertical bold lines when it is at its maximum, will never be too big. The only parameter of differential privacy formula is ϵ, and it governs the amount of noise we are going to add to the count. In Figure 7 two possible values are shown. A big ϵ will induce less noise and thus more precise results with peaked distributions. Such result will please analysts because of the increased precision but it will also make people worry about their privacy. On the other hand, a small ϵ will generate less noise and thus the probability distributions will be forced to be nearer to each other, getting also larger as the graph shows. In this case results will be more likely to be far from the true ones, thus better protecting people privacy. In Table 1 we report some frequently used values of ϵ and e^{ϵ}.

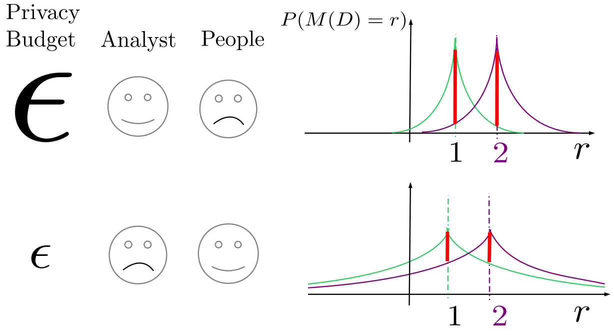


Figure 7: Contribution of ϵ to the added noise.

1.4.2 Differential privacy for numerical functions

We said that the Laplace noise is suitable for counting queries, but it turns out it can be added also to any numerical function $f : \mathcal{D} \rightarrow \mathbb{R}$ about the dataset we want to publish. Still, there is a catch: the amount of noise we must add is linked to the so-called global sensitivity of f:

Definition 3 (global sensitivity of a function). *We define the global sensitivity $\Delta(f)$ of a function $f : \mathcal{D} \rightarrow \mathbb{R}^w$, $w \in \mathbb{N}^+$, as*

$$\Delta\left(f\right) = \max_{\substack{D_1, D_2 \in \mathcal{D} \\ |D_1 \Delta D_2| = 1}} \left\| f\left(D_1\right) - f\left(D_2\right) \right\|_1$$

A function has low sensitivity if the addition or removal of one person to any database can only change the outcome of the function evaluation by a small amount. Notice how for a single counting function $c : \mathcal{D} \rightarrow \mathbb{N}$ the global sensitivity is low ($\Delta(c) = 1$) and thus the noise to add is limited. If instead we apply first a generic function f on a db D_1, and then on a neighboring db D_2, if f changes a lot it means we will need to add more noise to probably obtain the same output. For this reason in this paper we are going to describe principally methods that are based on calculating noisy counts, as they introduce less error in the output. Generally ϵ for counting queries is taken to be between 0.1 and 0.5. We can now formalize the procedure described so far with the so-called Laplace Mechanism:

Definition 4 (Laplace mechanism [20]). *Given a database $D \in \mathcal{D}$, and a function $f : \mathcal{D} \rightarrow \mathbb{R}^w$ with $w \in \mathbb{N}^+$ and global sensitivity Δ, an ϵ-differentially private mechanism $\mathcal{L}$ for releasing f is to publish $\mathcal{L}(D) = f(D) + X$, where X is a vector of random variables each drawn from a Laplace distribution Lap($\Delta(f)/\epsilon$). Laplacian density is $g\left(x\right) = \frac{1}{2b} \exp\left(-\left|z\right|/b\right)$, which is a symmetric distribution with variance $2b^2$.*

From the definition we can see it is possible to output not only one but many values (that is, statistics) about a database provided we add the right amount to noise to them. The noise is proportional to the global sensitivity $\Delta(f)$ of the vector of values f we are releasing.

1.5 Differential privacy weaknesses

1.5.1 Relaxations

Noise introduced by the randomization can produce results far from the true ones, thus leading to scarce utility of the published output for data consumers. Many relaxations of differential privacy exists to address this problem and the major one is (ϵ, δ)-differential privacy:

Definition 5 ((ϵ, δ)-dp [19]). *Given a randomized mechanism $M : \mathcal{D} \to \mathcal{R}$ we say M satisfies (ϵ, δ)-differential privacy if $\forall D_1, D_2 \in \mathcal{D}$ such that $|D_1 \Delta D_2| = 1$ and $R \subseteq \mathcal{R}$ the following equation holds:*

$$P\left(M\left(D_1\right) \in R\right) \leqslant e^{\epsilon} P\left(M\left(D_2\right) \in R\right) + \delta$$

We can see it only differs from the previous definition in the additional δ factor added to the right hand side of the equation. The meaning is to allow the release mechanism to fail providing regular differential privacy with a frequency given by δ. There are no hard and fast rules for setting ϵ and δ. It is generally left to the data releaser, and usually δ is taken to be very small, $\delta \leqslant 10^{-4}$. $(\epsilon, 0)$-dp is the same as ϵ-dp. Among the other relaxations we mention also (ϵ, δ)-probabilistic differential privacy:

Definition 6 ((ϵ, δ)-pdp [45]). *Given a randomized mechanism $M : \mathcal{D} \to \mathcal{R}$ and $\epsilon > 0$, $0 < \delta < 1$ constants we say M satisfies (ϵ, δ)-probabilistic differential privacy if $\forall D_1 \in \mathcal{D}$ we can divide the output space $\mathcal{R}$ into two sets $R, R' \subseteq \mathcal{R}$ such that*

$$P\left(M\left(D_1\right) \in R'\right) \leqslant \delta$$

and $\forall D_2 \in \mathcal{D}$ such that $|D_1 \Delta D_2| = 1$ and $\forall R \subseteq \mathcal{R}$ the following equations hold:

$$P\left(M\left(D_1\right) \in R\right) \leqslant e^{\epsilon} P\left(M\left(D_2\right) \in R\right)$$

This definition guarantees that algorithm M achieves ϵ-differential privacy with high probability $(1 - \delta)$. The set R' contains all outputs that are considered privacy breaches according to ϵ-differential privacy; the probability of such an output is bounded by δ. A mechanism satisfying (ϵ, δ)-pdp satisfies also (ϵ, δ)-dp and is thus stronger, but the converse does not hold.

1.5.2 Is differential privacy good enough?

Some people say even differential privacy is not enough to adequately protect individuals from data disclosure. Kifer and Machanavajjhala in [36] point out that differential privacy really works only if individuals are truly independent from each other. When there is no independence the participation of somebody in the db can be inferred just by looking at other (supposedly known and in relation with the "victim") entries. As a consequence, they claim we are forced to take into consideration adversarial knowledge, even if differential privacy apparently freed us from such a burden. In [10] a classifier is built to prove attacks against differentially private and l-diverse data (l-diversity [46] is a supposedly 'inferior', purely syntactical method) releases can have a quite similar accuracy. From a practical point of view, Dankar and El Emam [14] address several issues of differential privacy in the context of health care. They evidence a lack of real-life deployments of differentially private datasets, which might cause difficulties in assessing responsibilities if privacy breaches occur (was the ϵ value appropriate, who else used with success such an ϵ? etc...). It might also be difficult to explain the level of anonymization guaranteed to patients, as ϵ is a parameter of a formula quite theoretical in nature. Furthermore, since published data is obtained through randomization, sometimes it may look hard to believe - i.e. a randomized census dataset may indicate there are people living at the center of a lake. As a consequence, analysts might be lead to mistrust the approach (or who applied it). Lee and Clifton in [39] perform a study on how to set the right ϵ value to obtain a desired privacy level. In the conclusions they claim that any discussion of a differentially private mechanism requires a discussion of how to set an appropriate ϵ for that mechanism, a challenge that may be as or more difficult than developing the mechanism itself. Fu *et al* in [28] also question the utility of the ϵ since it is public and it is not that clear how it should be set. As a solution, Fu proposes the l'-diverted privacy model where ϵ parameter is set to 0. To avoid introducing distortions given by random perturbations, Bhaskar *et al* in [5] observe sufficiently large databases may already include enough entropy to induce sufficient uncertainty in the analyst without the need to add further noise to the results.

1.6 Mechanisms

The two main mechanisms are the already described Laplace mechanism [20] and the Exponential mechanism [49].

For the analysis whose outputs are not real or make no sense after adding noise (such as i.e. strings or partitionings), McSherry and Talwar propose the Exponential Mechanism, defined as:

Definition 7 (exponential mechanism [49]). *A mechanism $M : \mathcal{D} \to \mathcal{R}$ is said to be exponential if it selects an output $r \in \mathcal{R}$ from the output domain by taking into consideration its score of a given utility function $q : (\mathcal{D} \times \mathcal{R}) \to \mathbb{R}$ in a differentially private manner. The exponential mechanism assigns exponentially greater probabilities of being selected to outputs of higher scores so that the final output would be close to the optimum with respect to q. The chosen utility function q should be insensitive to changes of any particular record, that is, have a low sensitivity. Let the sensitivity of q be $\Delta q = \max\limits_{\forall r, D_1, D_2} \left|q\left(D_1, r\right) - q\left(D_2, r\right)\right|$ for $|D_1 \Delta D_2| = 1$ then the mechanism $M\left(D, q\right) = \left\{ \textit{return } r \textit{ with probability} \quad \propto \quad \exp\left(\frac{\epsilon \cdot q(D,r)}{2\Delta q}\right)\right\}$ gives ϵ-privacy*

This mechanism allows for example to publish the most frequent eye color of persons in a room. Other mechanisms are Li *et al*' Matrix mechanism [40], the Geometric mechanism (a discretized version of the Laplace mechanism) by Ghosh et al [29] and Dwork et al's Gaussian mechanism [19].

2. MEASURING UTILITY

Broadly speaking, the utility of a mechanism is its capability to minimize the error, which is a measure of the distance between original input db/statistics on it and noisy output db/statistics . As we will explain later in Section 3, only utility of restricted classes of queries can be guaranteed in the non-interactive setting. Blum, Ligett, and Roth [6] showed that in such setting it is possible to answer exponentially sized families of counting queries so in this paper we will mostly look at solutions for publishing data that are useful for such queries. However, the choice of suitable statistics is a difficult problem as these statistics need to mirror the sufficient statistics of applications that will use the sanitized database, and for some applications the sufficient statistics are hard to characterize. Popular approaches to measure utility are (α, β)-usefulness [6], relative error with correction for small queries [59, 8] and without correction [11, 60], absolute error [12, 16, 44], variance of the error[11, 59, 16], euclidean distance [44, 32]. In the following, we are going to define them more precisely.

Definition 8 ((α, β) −usefulness[6]). *A privacy mechanism M is (α, β)-useful for queries in class C if with probability $1-\beta$, for every $Q \in C$ and every dataset $D \in \mathcal{D}$, for $\tilde{D}=M\left(D\right)$, $\left|Q\left(\tilde{D}\right) - Q\left(D\right)\right| \leqslant \alpha$*

It is adopted in [6],[60] (only for a basic cell based algorithm), and [7]. (α, δ)-usefulness is effective to give an overall estimation of utility, but according to [9] fails to provide intuitive experimental results. [9, 8, 59] experimentally measure the utility of sanitized data for counting queries by relative error adopting this formula:

Definition 9 (relative error [59]). *Let Q be a query and $M : \mathcal{D} \rightarrow \mathcal{R}$ a privacy mechanism. We denote relative error as $\mathrm{rel}\left(Q\right) = \frac{\left|Q\left(\tilde{D}\right)-Q\left(D\right)\right|}{\max\left(Q\left(D\right),s\right)}$ where s is a sanity bound that mitigates the effects of the queries with excessively small selectivities. In both [59] and [9] s is set to 0.1% of $|D|$.*

When the database is considered as a vector of reals (so $h = 1, A_1 = \mathbb{R}$) the euclidean distance can be used as utility. Li *et al* in [44] measure the error as the euclidean distance between original and noisy database $\mathrm{Err}\left(D\right) = \left\|D - M\left(D\right)\right\|_2$, claiming their mechanism is capable in such a way to guarantee the utility for any class of queries. Hardt *et al* [32] measure the euclidean distance between query responses.

3. THEORETICAL BOUNDS

Many theoretical results show the existing relations between the parameter ϵ or eventually δ for (ϵ, δ)-dp and the quality of the released database. First analysis about differential privacy [17, 24, 21] proved strong negative results about the amount of information that could be published with reasonable accuracy. In particular, Dinur *et al* [17] showed that, in order to avoid database reconstruction (*blatant non-privacy*), the minimum amount of noise to be added to subset sum queries answers of a database made of a bit per user is of magnitude $\Omega\left(\sqrt{|D|}\right)$. This fact lead initially many researchers to concentrate on scenarios in which the number of queries that can be answered by a statistical database system is limited, such as in the interactive setting. This restriction allowed only answering a sublinear number of queries in the size of database $|D|$. By using notions from learning theory, Blum, Ligett and Roth proved the possibility of non-interactive data release satisfying differential privacy for queries with polynomial VC-dimension, such as predicate queries. By carefully selecting a class of concepts C with functions $c : \mathcal{U} \rightarrow \{0, 1\}$ as elements, it is possible to privately answer counting queries with noise that grows only logarithmically with the number of queries asked (or more generally with the VC-dimension of the query class). Utility is guaranteed by the (α, β)-usefulness criteria, which tell us that with high probability $1-\beta$ all errors are bound by α. Dwork in [23] provides a non-interactive mechanism which extends to arbitrary low-sensitivity queries rather than only counting queries. The extension makes crucial use of the relaxation to (ϵ, δ)-privacy as motivated by De in [15], in which relations between ϵ-dp and (ϵ, δ)-dp are discussed. Although Dwork *et al* [20] prove any function with low sensitivity can be computed privately, De shows that answering arbitrary low-sensitivity queries requires more noise than answering counting queries.

4. METHODS

With differential privacy as soon as we look at the data we must ask ourselves if we are leaking information. Analysis of the disclosure amount represented by parameter ϵ can be difficult, but fortunately several mechanism properties can be used to analyze and construct algorithms. Among them, we find:

Theorem 1 (sequential composition [48]). *Let M_1 preserves ϵ_1-dp and M_2 preserves ϵ_2-dp.*
Then $M\left(D\right) = \left(M_1\left(D\right), M_2\left(D\right)\right)$ preserves $\epsilon_1 + \epsilon_2$-dp

Theorem 2 (parallel composition [48]). *Let M_1 preserves ϵ_1-dp and M_2 preserves ϵ_2-dp.*
Then $M\left(D\right) = \left(M_1\left(D_1\right), M_2\left(D_2\right)\right)$ preserves $\max\left(\epsilon_1, \epsilon_2\right)$-dp

Theorem 3 (post-processing [48]). *If $M : \mathcal{D} \rightarrow \mathcal{R}$ preserves ϵ-dp and $f : \mathcal{R} \rightarrow \mathcal{R}'$ is any arbitrary (database independent) function, then $f\left(M\right) : \mathcal{D} \rightarrow \mathcal{R}'$ preserves ϵ-dp.*

We can observe sampling is an inherent source of randomness so we can treat its use as a mechanism:

Theorem 4 (sampling[11]). *Given a mechanism M which provides ϵ-differential privacy, and $0 < p < 1$, including each element of the input into a sample S with probability p and outputting $M(S)$ is $2pe^{\epsilon}$-differentially private*

These properties permit to split the privacy budget ϵ and to optimally assign parts of it to various mechanism tasks. Also, since in some cases the randomization might produce contradictory results, accuracy can be augmented by fixing the conflicting data after the randomization algorithm has produced its output. To perform this operation usually only the result of a differentially private mechanism is used without accessing again the database, so for the post-processing property no further ϵ privacy budget must be spent. We will talk more about consistency checks in Section 4.3.

Several methods have been proposed to address the issue of releasing differentially private data. Broadly speaking, they can be divided in the categories of histogram construction, sampling and filtering, partitioning, dimensionality reduction. In the following, we are going to review them. The notation $\tilde{O}$ indicates complexity with hidden logarithmic factors.

4.1 Computing histograms

A histogram is a disjoint partition of the database points with the number of points which fall into each partition. In Figure 8 we can see an example of a simple database, made

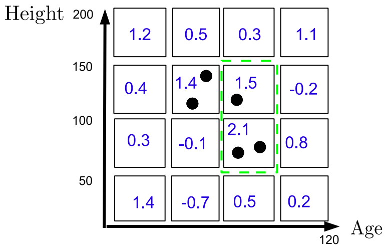


Figure 8: A noisy histogram release.

by only two attributes, *Height* and *Age*. The actual data we are going to publish is represented by the noisy counts of each cell. If an analyst wants to know how many people are present under the dashed rectangle he may then just sum the single noisy counts of the cells contained in the query region. Publishing a noisy version of the histogram is appealing because of its usefulness for counting queries, which constitute the basis of many data mining tasks. However, the quality of queries executed on the histogram may be low. As we have already seen in Section 1.4, the amount of Laplace noise to add to a single cell count is modest and thus acceptable. On the other hand, if a query requires the sum of n histogram cell counts, since each of them has some noise the total noise sums up n times and can quickly become intolerable. This is explained by the fact the total variance of the error also sums up. There are ways to limit the variance, though.

4.1.1 Exploiting linear combinations

Suppose we have four cells having respectively counts c_1, c_2, c_3, c_4 and some analyst is interested in the total count of them $c(T) = \sum_{i=1}^{4} c_i$. If we publish the vector of 4 single noisy counts $\tilde{c} = c + l$, where l is a vector of four samples drawn from the Laplace distribution, each published noisy count will have variance V. The analyst will than have to sum each of the noisy values for a total variance of $4V$. But, knowing the analyst is interested in the sum, we could also provide him with the noisy sum: $\tilde{c}(T_4) = \sum_{i=1}^{4} c_i + l$, with l being only one single value this time. This would have just variance V, albeit in this case we would have to pay an extra ϵ budget because, for the sequential composition property (Theorem 1), we are publishing twice information about the same individuals. The interesting fact about this procedure is that the analyst, having counts of overlapping regions, could further exploit this knowledge to get better approximations of *unpublished* counts. Suppose he also wanted to know the total count $c(T_3)$ of c_1, c_2 and c_3: he could add the three single noisy counts, but then the error variance would be $3V$. A better alternative is given by performing a subtraction of the noisy count of c_4 from the total noisy count: $\tilde{c}(T_3) = \tilde{c}(T_4) - \tilde{c}_4$. For the properties of variance, we would have to sum (by subtraction) only two published noisy counts thus having a variance of $2V$. More generally, we can calculate the counting queries we desire as linear combinations of other noisy counting queries to minimize error variance. These linear combinations can be conveniently stored as coefficients in matrices, and for this reason these principles were adopted in the so-called Matrix Mechanism by Li *et al* [40]. The knowledge of the query workload that analysts desire from the dataset is exploited to obtain a different set of noisy queries, called the strategy, from which the answer to the workload can be then computed without accessing again the database. Knowing the workload beforehand is important because a differentially private release cannot maximize utility for all type of queries, so restricting the query space helps calibrating at best the coefficients to assign to each cell. Although the process helps to significantly reduce the error variance, the computational cost to calculate the optimal strategy matrix given a generic query workload is polynomial in $|\mathcal{U}|$. Any dataset with several attributes A_i leads to huge contingency matrices of size $|\mathcal{U}| = \Pi_i |A_i|$, making Li's method quickly inapplicable. To solve the issue of finding the optimal strategy matrix and improve the Matrix Mechanism it is possible to exploit dependencies in the query workload as done by Li in [41] (considering the (ϵ, δ)-dp relaxation), Yuan [41], and Cormode [13]. While Li's and Yuan's works use a fixed privacy budget ϵ to obtain each noisy count, Cormode [13] adopts a non-uniform scheme. Intuitively, it doesn't make sense to add the same amount of noise to a single cell and to a whole group of cells. Counts of big areas are less likely to be significantly distorted by error, so we can afford to assign a smaller ϵ budget to their noisy count. Conversely, since single cells can have small values, the magnitude of which we would like to preserve, it is worth to spend more budget to reduce their error variance. Unfortunately, Cormode points out that finding the strategy, the optimal budget allocation and a way to obtain the desired answers from the published queries given the query workload is computationally very expensive, so two of the three former parameters must be fixed beforehand to obtain results in an acceptable time. In particular, Cormode focuses on workloads of k-way marginals, demonstrating how, given a strategy, it is possible to lower the error by calculating an optimal privacy budget distribution and consistent results in time substantially linear in $|\mathcal{U}|$ while guaranteeing formal error bounds. We note how all the methods discussed so far are independent of the dataset, and only consider the query workload as input. Once the strategy (and eventually the budget allocation) has been calculated, it can be used with any instance of the database and the anonymization can be performed in time $O(|\mathcal{U}|)$ by just adding Laplace noise according to the parameters previously discovered.

4.1.2 Adopting fixed strategies

It is also possible to adopt a fixed strategy and assume it will provide sufficient utility for most workloads of interest, although we already know this is not theoretically possible. Anyway, even adopting a fixed strategy matrix will not spare us the burden to add noise to $|\mathcal{U}|$ cells, which can still be prohibitive for certain datasets. Among the other works suffering in general from $|\mathcal{U}|$ size we find [22, 16, 59, 60, 2, 33]. Xiao in [59] operates a transform on the counts and adds noise in the wavelet domain in time $O(|\mathcal{U}| + |D|)$. Similar techniques via post-processing with overlapping information are suggested by Hay in [33]. In both cases the best utility is obtained for range queries, where attributes values are ordered and sums of contiguous cells are requested by the analyst.

4.2 Sampling and filtering

One possible solution to the histogram problem is to take advantage of sparsity of data present in many databases. This condition occurs when the number of cells $|\mathcal{U}^+|$ with

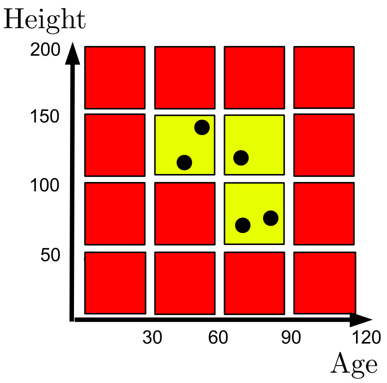


Figure 9: Data sparsity put in evidence.

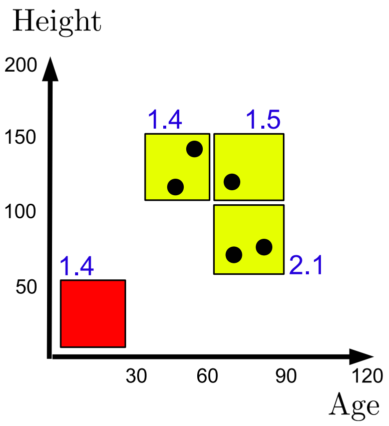


Figure 10: Noisy sample of cells.

positive count in the contingency table in the database at hand is much bigger than zero-valued entries. To prove this fact Cormode *et al* in [12] define sparsity ρ as $\rho = \left|\mathcal{U}^{+}\right| / \Pi_i \left|A_i\right|$. Table 2 is an example of the fact many natural datasets have low density in the single-digit percentage range, or less. Applying differential privacy naively generates output which is $1/\rho$ times larger than the data size. In the examples table, $1/\rho$ ranges from tens to thousands. which is clearly not practical for today's large data sizes. Among the methods which exploit data sparsity we find [12, 44, 7]. In [7] this definition of m-sparse queries is proposed:

Definition 10 (m-sparse query [7]). *We say that a linear query Q is m-sparse if it takes non-zero values on only m universe elements, and that a class of queries is m-sparse if each query it contains is m'-sparse for some $m' \leqslant m$.*

Dataset	Density ρ
OnTheMap [58]	3-5%
Census Income [51]	0.4-4%
UCI Adult Data [37]	0.14%

Table 2: Real-life datasets with their sparsity value

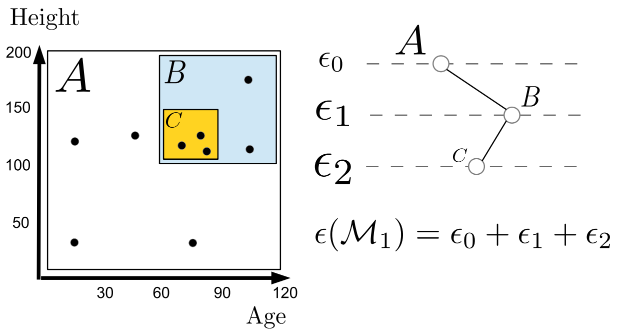


Figure 11: Simple partitioning with associated spatial index and privacy cost.

For the sampling and filtering category the idea is to avoid publishing huge contingency tables by filtering out entries with small counts, which are often in significant quantity in many databases. Looking at the database example in Figure 9, ideally we would like to publish only noisy counts of the three occupied cells. But just publishing those three counts would inform adversaries the other cells are empty. This would violate differential privacy, which requires to protect the privacy even of people *outside* the database. If we filter the noisy counts, probably counts of empty cells are going to be low and filtered out. Because of randomness in the final release we might still find some empty cell with the noisy count big enough to pass the filter (we can see an example in Figure 10 supposing the threshold is set to 1.0). Cormode *et al* [12] adopt a variety of filtering techniques - highpass filtering and priority sampling being the most useful - to override the costly operation of materializing a complete noisy contingency table. Their method is suited for sparse datasets and provides minimum error variance for subset sum queries. If s is the desired number of cells to publish the expected running time is $O(s + |\mathcal{U}^{+}|)$. Chen *et al* [8] consider the publication of trajectories of individuals. Each trajectory step can be done in a universe of locations L for a maximum of 10 steps to still guarantee privacy. Chen's method counts the frequencies of occurrence of each trajectory in the database by building a prefix tree. Each node in the tree represents a location and the noisy frequency with which the path from the root to the node can be found in the database. For differential privacy constraints, fake nodes must be added to the tree to represent non-existing trajectories, but to limit their number a filtering method similar to Cormode's is adopted. A further consistency checking pass at the end is performed to minimize error variance, and the algorithm complexity is $O(|D| \cdot |L|)$. For search log analysis Korolova in [38] and Gotz in [30] propose a mechanism to release noisy aggregated user query and clicked url counts by filtering out excessively small counts. However, such approach breaks the association between distinct query-url pairs in the output since all the user-IDs are removed, which might be useful in only a few applications. Therefore, in [34] a sampling method is proposed to allow analysis in exactly the same fashion and for the same purpose as the original data. However, (ϵ, δ)-pdp is adopted to provide formal guarantees because relaxations are indispensable in search log publishing as proven in [30].

4.3 Partitioning

Partitioning is indicated for ordered attributes such as spatial data. Like in algorithms computing histograms, the universe $\mathcal{U}$ is divided into regions but in this case the shape of the cells or their number is not fixed and an attempt is made to find an optimal subdivision of the space. Regions

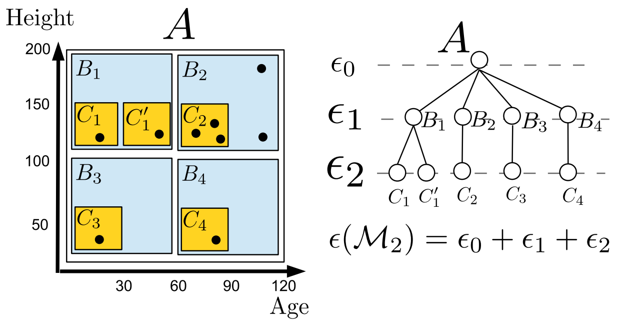


Figure 12: A more complex partitioning with associated spatial index and privacy cost.

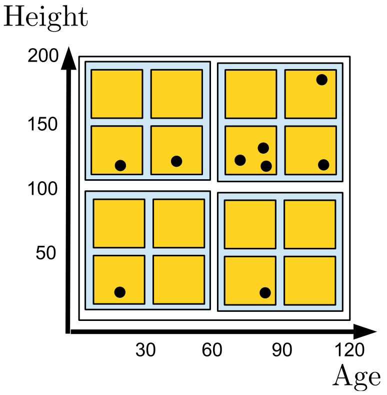
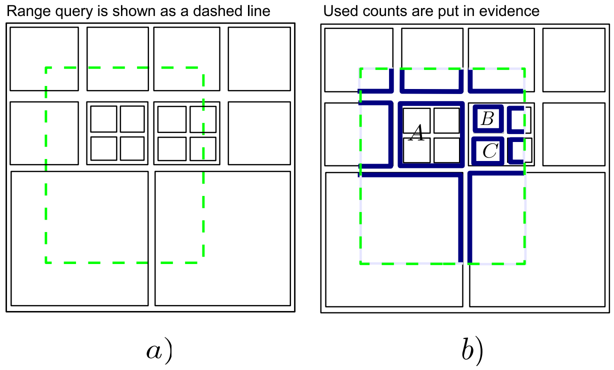


Figure 13: A quad-tree partitioning.

may be overlapping and for each found region a noisy count of the people inside is taken. For these reasons partitioning can be seen as the construction of a spatial index, an example of which can be seen in Figure 11. To each node in the index there is associated a corresponding region, a noisy count of individuals in the region and also an ϵ privacy budget to calibrate the noise. In this example we chose to assign a greater budget to deeper levels in the tree, meaning their counts will be more precise. Only the index with its noisy counts and region shapes is published. The goal is to optimize the results of range queries, where the analyst asks for the number of people lying under a given query area, usually expressed as an hyperrectangle. This calculation involves the sum of already published noisy counts so a strategy to allow the user to minimize the total noise variance must also be provided. In Figure 14 a possible strategy suggested by Cormode in [11] is proposed, where counts corresponding to the greatest regions completely enclosed by the query area are taken and summed. Some regions on the borders of the query will be only partially covered. In this case only the fraction of their noisy count corresponding to the part actually under the query might be taken into the sum.


Range query is shown as a dashed line
Used counts are put in evidence
A
B
C

a) *b)*

Figure 14: A possible range query over a partitioning.

4.3.1 Privacy cost of partitioning

Here we report some properties partitionings can have:

Definition 11 (Regular decomposition). *The underlying partitioning scheme is fixed, so lines' position is not determined by the data in the db. An example of this is the quad-tree partitioning as in Figure 13, where space is recursively subdivided into four regions until the desired level of granularity is reached.*

Definition 12 (Containment). *Each node region fully contains the regions of its descendant.*

Definition 13 (Disjointness). *For every node all of its children regions must be disjoint.*

Theorem 5 (Nested counts). *Assuming the partitioning satisfies containment and disjointness properties and a privacy budget $v.\epsilon$ is assigned to each node v, then for the composition properties of differential privacy the privacy cost of releasing the index is the maximum of the sum of the node budgets along any possible path from the root to the leaves. Since each path corresponds to a subset of users disjoint from the other paths, for the disjointness property of differential privacy (Theorem 2) only the maximum privacy cost among all paths is taken.*

We can now derive the total privacy cost of the example of Figure 11. Since the count of region A will also include the counts of region B and of region C, for the composition property of differential privacy (Theorem 1) this triple intersection will cost a total privacy budget of $\epsilon(M_1) = \epsilon_0+\epsilon_1+\epsilon_2$. What would be the cost if we took counts of other regions, as mechanism M_2 does in Figure 12? According to Theorem 5, the privacy budget would be exactly the same. In this partitioning case, there could be consistency anomalies in the noisy counts. In particular, the sum of the noisy counts of regions B_1, B_2, B_3, B_4 might not be equal to the noisy count of the root region A. To solve this issue techniques can be adopted to make the counts consistent while keeping error variance low.

4.3.2 Approaches to partitioning

A popular approach to partitioning is with *kd*-trees: at each round, an attribute is chosen and points in the database are split in 2 disjoint sets according to some criteria. Usually uniformity in the number of points on both sides of the splitting line is considered by choosing the median. Noisy counts of the two newly founded partitions are then published and partitioning is done recursively. Because of noise, the sum of noisy counts of subpartitions may not be equal to the partition containing them. To solve this issue, consistency checks can then be applied to the counts to make them appear consistent. The idea of differentially private data-partitioning index structures is suggested in the context of private record matching in [35]. The approach there is based on using an approximate mean as a surrogate for median (on numerical data) to build *kd*-trees. The approach of Xiao *et al* [60] imposes a fixed resolution grid over the base data. It then builds a *kd*-tree based on noisy counts in the grid, splitting nodes which are not considered 'uniform', and then populates the final leaves with 'fresh' noisy estimated counts. Quad-tree partitioning simply imposes a recursive fixed grid in which at each round the space is divided into four rectangular cells of the same size. See Figure 13 for an example. In [11] by Cormode *et al*, a comparison between several median finding methods, Hilbert R-trees and quadtrees partitioning is performed and privacy budget is allocated in a geometrically increasing way to counts during the partitioning of 2D data. An ordinary least squares (OLS) estimator is devised to achieve consistency and minimum error variance in time linear in the size of the published tree. Quad-tree partitioning is found to be fast and superior in quality of the output to all the other tested methods.

4.4 Dimensionality reduction

Dimensionality reduction methods usually consider the database as a matrix and apply random projections on it. In this line of research we find [7], in which for the class of linear counting queries that are m-sparse a method based on releasing a perturbed random projection of the private database together with the projection matrix is described. Running time is polynomial in the database size $|D|$, m, and $\log|\mathcal{U}|$. In [63] compression is applied to obtain a reduced synthetic database D' of size $|D'| \ll |D|$ in polynomial time. Li et al [44] apply compressive sensing to obtain a perturbed database from sparse data through decompression in time $\tilde{O}(|D|)$. Li *et al* in [43] for set-valued data obtain a set of frequent itemsets called basis for which any frequent itemset is a subset of some basis.

5. APPLICATIONS

In recent years differential privacy has been successfully applied to a wide range of real-world data, although generally with no quality assessment by final users of anonymized datasets. In [45] (ϵ, δ)-pdp is introduced to model spatial data. This solution is then compared by Cormode with his work in [12]. Xiao *et al* in [60] apply a *kd*-tree technique on CENSUS data [51], and results are found superior to Inan's *et al* hierarchical tree method [35]. Moreover, the open source HIDE platform [61] is provided to experiment with four differentially private algorithms: [33, 35, 59, 9]. Cormode later in [11] found his algorithm to give less error than Inan's [35] and Xiao's works [60]. In [9] MSNBC [50] and STM [54] datasets represented as set-valued boolean data are considered. The only comparison is performed for MSNBC against basic noisy datacube method of Dwork's[20], as STM has big universe $|\mathcal{U}|$ size and few methods are capable to handle this situation. A successive paper of Li *et al* [43] for set-valued data is applied to Retail Dataset of an anonymous Belgian retail store [4], the Mushroom Dataset [4], AOL Search log [1], Pumsb_star from PUMS (Public Use Microdata Sample) [4] and Kosarak Dataset [4], where each transaction is the clickstream of a user of a Hungarian website. Li notes that the method of Chen [9] applied to these datasets would generate either an empty synthetic dataset or a dataset which is highly inaccurate. In his opinion, Chen's method would provide reasonable performance only when the number of items is small. We note this statement contrasts with Chen's results about the STM dataset, which has a fairly large universe size. The STM dataset represented as sequences of locations is also considered again by Chen in [8], although location coordinates nor time intervals are taken into account. In [30] publication of counting queries for search logs is considered, but dataset origin is not specified. In [34] AOL search log [3] is adopted for experimental tests. [59] performs experiments on CENSUS data [51] using binning to have $|\mathcal{U}| \approx$16,000,000.

6. SYNTHETIC DATABASES

There have been few attempts to devise mechanisms of the kind $M : \mathcal{D} \to \mathcal{D}$, because privacy in these cases is more difficult to preserve. Outputs can be either a synthetic database - in which individuals follow the same distribution as in the original database - or just a perturbed version, where rows are directly taken from the original database with some modification to guarantee anonymity. Perturbed database release is considered in [9, 8, 44]. Synthetic data is released with methods proposed in [63, 45, 34]. Gupta *et al* in [31] considers the release of a complete relational database with many tables for performance testing purposes. Anonymization of single tables is performed with the Matrix Mechanism to take into account possible query workloads.

7. CONCLUSIONS

Differential privacy provides formal guarantees that public opinion needs when privacy is at stake, yet for many years such requirements were judged by researchers too strict to be applicable. Recently, several breakthrough results changed this mood. We presented a variety of methods - partitioning, dimensionality reduction, sampling and filtering - which have been successfully applied to many real-life datasets. Some methods were also shown for histogram publishing, which, albeit unfeasible on certain databases with big universe size, can still be used in practice on some real life datasets. Most of the papers we discussed about use a plain ϵ-dp model which seems to indicate relaxations may not really be needed except in problematic cases like search log publishing. Differential privacy can be applied efficiently with formal guarantees to set-valued data [43, 9], sparse data for subset sum counting queries [12], sequences of short length [8], bidimensional spatial data [11] and for general purpose queries [44]. When data is not sparse and $|\mathcal{U}|$ is not too big Xiao's wavelet method [59] can be used with success. If the query workload is known with correlated queries further improvements can be given by Yuan's method [62] albeit at a greater computational cost. When the workload is made of k-way marginals Cormode's [13] permits to reach low error with consistent results in time substantially linear in $|\mathcal{U}|$, while guaranteeing formal error bounds. For the difficult case of search log publishing Hong *et al* [34] showed it is even possible to publish a perturbed database while maximizing utility. For these reasons time is ripe for the Open Data movement to start considering the adoption of differential privacy and provide people with adequate guarantees about the way their data is handled. Research has still to be done to impose constraints on output data in order to avoid inconsistencies and to properly anonymize highly dimensional non-sparse data and preserving utility of general classes of queries. In this regard, publication of synthetic or perturbed datasets seems a promising approach, which needs careful query utility examination.

8. REFERENCES

[1] America Online. AOL Database. http://gregsadetsky.com/aol-data/.

[2] Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., and Talwar, K. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (New York, NY, USA, 2007), PODS '07, ACM, pp. 273–282.

[3] Barbaro, M., and Zeller, T. A face is exposed for aol searcher no. 4417749. *New York Times* (2006).

[4] Bayardo, R. Frequent Itemset Mining Dataset Repository. http://fimi.ua.ac.be/data.

[5] Bhaskar, R., Bhowmick, A., Goyal, V., Laxman, S., and Thakurta, A. Noiseless database privacy. In *Proceedings of the 17th international conference on The Theory and Application of Cryptology and Information Security* (Berlin, Heidelberg, 2011), ASIACRYPT'11, Springer-Verlag, pp. 215–232.

[6] Blum, A., Ligett, K., and Roth, A. A learning theory approach to non-interactive database privacy. In *Proceedings of the 40th annual ACM symposium on Theory of computing* (New York, NY, USA, 2008), STOC '08, ACM, pp. 609–618.

[7] Blum, A., and Roth, A. Fast private data release algorithms for sparse queries. *ArXiv e-prints* (nov 2011).

[8] Chen, R., Fung, B. C. M., and Desai, B. C. Differentially private trajectory data publication. *CoRR* (2011), –1–1.

[9] Chen, R., Mohammed, N., Fung, B. C. M., Desai, B. C., and Xiong, L. Publishing set-valued data via differential privacy. *PVLDB 4*, 11 (2011), 1087–1098.

[10] Cormode, G. Personal privacy vs population privacy: learning to attack anonymization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2011), KDD '11, ACM, pp. 1253–1261.

[11] Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., and Yu, T. Differentially private spatial decompositions. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering* (Washington, DC, USA, 2012), ICDE '12, IEEE Computer Society, pp. 20–31.

[12] Cormode, G., Procopiuc, C., Srivastava, D., and Tran, T. T. L. Differentially private summaries for sparse data. In *Proceedings of the 15th International Conference on Database Theory* (New York, NY, USA, 2012), ICDT '12, ACM, pp. 299–311.

[13] Cormode, G., Procopiuc, C. M., Srivastava, D., and Yaroslavtsev, G. Accurate and Efficient Private Release of Datacubes and Contingency Tables. *ArXiv e-prints* (July 2012).

[14] Dankar, F. K., and El Emam, K. The application of differential privacy to health data. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops* (New York, NY, USA, 2012), EDBT-ICDT '12, ACM, pp. 158–166.

[15] De, A. Lower bounds in differential privacy. In *Proceedings of the 9th international conference on Theory of Cryptography* (Berlin, Heidelberg, 2012), TCC'12, Springer-Verlag, pp. 321–338.

[16] Ding, B., Winslett, M., Han, J., and Li, Z. Differentially private data cubes: optimizing noise sources and consistency. In *Proceedings of the 2011 international conference on Management of data* (New York, NY, USA, 2011), SIGMOD '11, ACM, pp. 217–228.

[17] Dinur, I., and Nissim, K. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (New York, NY, USA, 2003), PODS '03, ACM, pp. 202–210.

[18] Dwork, C. A firm foundation for private data analysis. *Commun. ACM 54*, 1 (Jan. 2011), 86–95.

[19] Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: privacy via distributed noise generation. In *Proceedings of the 24th annual international conference on The Theory and Applications of Cryptographic Techniques* (Berlin, Heidelberg, 2006), EUROCRYPT'06, Springer-Verlag, pp. 486–503.

[20] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third conference on Theory of Cryptography* (Berlin, Heidelberg, 2006), TCC'06, Springer-Verlag, pp. 265–284.

[21] Dwork, C., McSherry, F., and Talwar, K. The price of privacy and the limits of lp decoding. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing* (New York, NY, USA, 2007), STOC '07, ACM, pp. 85–94.

[22] Dwork, C., Naor, M., Reingold, O., Rothblum, G. N., and Vadhan, S. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the 41st annual ACM symposium on Theory of computing* (New York, NY, USA, 2009), STOC '09, ACM, pp. 381–390.

[23] Dwork, C., Rothblum, G. N., and Vadhan, S. Boosting and differential privacy. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science* (Washington, DC, USA, 2010), FOCS '10, IEEE Computer Society, pp. 51–60.

[24] Dwork, C., and Yekhanin, S. New efficient attacks on statistical disclosure control mechanisms. In *Proceedings of the 28th Annual conference on Cryptology: Advances in Cryptology* (Berlin, Heidelberg, 2008), CRYPTO 2008, Springer-Verlag, pp. 469–480.

[25] European Commission. Proposal for a directive of the european parliament and of the council COM(2011) 877 final, December 2011.

[26] European Data Protection Supervisor (EDPS). Opinion EDPS/08/12, Apr. 2012.

[27] European Parliament. Directive 95/46/EC (OJ L 281/95), October 95.

[28] Fu, A. W.-C., Wang, J., Wang, K., and Wong, R. C.-W. Small count privacy and large count utility in

data publishing. *CoRR abs/1202.3253* (2012).

[29] Ghosh, A., Roughgarden, T., and Sundararajan, M. Universally utility-maximizing privacy mechanisms. In *Proceedings of the 41st annual ACM symposium on Theory of computing* (New York, NY, USA, 2009), STOC '09, ACM, pp. 351–360.

[30] Gotz, M., Machanavajjhala, A., Wang, G., Xiao, X., and Gehrke, J. Publishing search logs: A comparative study of privacy guarantees. *IEEE Trans. on Knowl. and Data Eng. 24*, 3 (Mar. 2012), 520–532.

[31] Gupta, V., Miklau, G., and Polyzotis, N. Private database synthesis for outsourced system evaluation. In *Proceedings of the 5th Alberto Mendelzon International Workshop on Foundations of Data Management, Santiago, Chile, May 9-12, 2011* (2011).

[32] Hardt, M., and Talwar, K. On the geometry of differential privacy. In *Proceedings of the 42nd ACM symposium on Theory of computing* (New York, NY, USA, 2010), STOC '10, ACM, pp. 705–714.

[33] Hay, M., Rastogi, V., Miklau, G., and Suciu, D. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow. 3* (September 2010), 1021–1032.

[34] Hong, Y., Vaidya, J., Lu, H., and Wu, M. Differentially private search log sanitization with optimal output utility. In *Proceedings of the 15th International Conference on Extending Database Technology* (New York, NY, USA, 2012), EDBT '12, ACM, pp. 50–61.

[35] Inan, A., Kantarcioglu, M., Ghinita, G., and Bertino, E. Private record matching using differential privacy. In *Proceedings of the 13th International Conference on Extending Database Technology* (New York, NY, USA, 2010), EDBT '10, ACM, pp. 123–134.

[36] Kifer, D. No free lunch in data privacy. *Security 99*, 17 (2011), 193–204.

[37] Kohavi, R., and Becker, B. http://archive.ics.uci.edu/ml/datasets/Adult.

[38] Korolova, A., Kenthapadi, K., Mishra, N., and Ntoulas, A. Releasing search queries and clicks privately. In *Proceedings of the 18th international conference on World wide web* (New York, NY, USA, 2009), WWW '09, ACM, pp. 171–180.

[39] Lee, J., and Clifton, C. How much is enough? choosing ϵ for differential privacy. In *Proceedings of the 14th international conference on Information security* (Berlin, Heidelberg, 2011), ISC'11, Springer-Verlag, pp. 325–340.

[40] Li, C., Hay, M., Rastogi, V., Miklau, G., and McGregor, A. Optimizing linear counting queries under differential privacy. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (New York, NY, USA, 2010), PODS '10, ACM, pp. 123–134.

[41] Li, C., and Miklau, G. An adaptive mechanism for accurate query answering under differential privacy. *Proc. VLDB Endow. 5*, 6 (Feb. 2012), 514–525.

[42] Li, N., Li, T., and Venkatasubramanian, S. t-closeness: Privacy beyond k-anonymity and l-diversity. In *In ICDE* (2007).

[43] Li, N., Qardaji, W., Su, D., and Cao, J. Privbasis: frequent itemset mining with differential privacy. *Proc. VLDB Endow. 5*, 11 (July 2012), 1340–1351.

[44] Li, Y. D., Zhang, Z., Winslett, M., and Yang, Y. Compressive mechanism: utilizing sparse representation in differential privacy. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society* (New York, NY, USA, 2011), WPES '11, ACM, pp. 177–182.

[45] Machanavajjhala, A., Kifer, D., Abowd, J. M., Gehrke, J., and Vilhuber, L. Privacy: Theory meets practice on the map. In *ICDE'08* (2008), pp. 277–286.

[46] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data 1* (March 2007).

[47] McKinsey Global Institute. Big data: The next frontier for innovation, competition, and productivity, 2011.

[48] McSherry, F. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Commun. ACM 53* (Sept. 2010), 89–97.

[49] McSherry, F., and Talwar, K. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science* (Washington, DC, USA, 2007), FOCS '07, IEEE Computer Society, pp. 94–103.

[50] Microsoft. http://archive.ics.uci.edu/ml/datasets/MSNBC.com+Anonymous+Web+Data.

[51] Minnesota Population Center (MPC). http://www.ipums.org.

[52] Narayanan, A., and Shmatikov, V. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy* (Washington, DC, USA, 2008), IEEE Computer Society, pp. 111–125.

[53] O'Hara, K. Transparent government, not transparent citizens: A report on privacy and transparency for the cabinet office, Sept. 2011.

[54] Societè de transport de Montrèal. http://www.stm.info.

[55] Sweeney, n. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10* (October 2002), 557–570.

[56] UK Cabinet office. Making open data real: A public consultation, 2011.

[57] UK Office for National Statistics. Evaluating a statistical disclosure control (sdc) strategy for 2011 census outputs, 2011.

[58] US Census Bureau. US Census Bureau, http://lehdmap.did.census.gov/.

[59] Xiao, X., Wang, G., and Gehrke, J. Differential privacy via wavelet transforms. *IEEE Trans. on Knowl. and Data Eng. 23*, 8 (Aug. 2011), 1200–1214.

[60] Xiao, Y., Xiong, L., and Yuan, C. Differentially private data release through multidimensional partitioning. In *Proceedings of the 7th VLDB conference on Secure data management* (Berlin, Heidelberg, 2010), SDM'10, Springer-Verlag, pp. 150–168.

[61] Xiong, L., and Gardner, J. HIDE. http://www.mathcs.emory.edu/hide/index.html.

[62] Yuan, G., Zhang, Z., Winslett, M., Xiao, X., Yang, Y., and Hao, Z. Low-rank mechanism: optimizing batch queries under differential privacy. *Proc. VLDB Endow. 5*, 11 (July 2012), 1352–1363.

[63] Zhou, S., Ligett, K., and Wasserman, L. Differential privacy with compression. In *Proceedings of the 2009 IEEE international conference on Symposium on Information Theory - Volume 4* (Piscataway, NJ, USA, 2009), ISIT'09, IEEE Press, pp. 2718–2722.