The Application of Differential Privacy to Health Data

Fida Kamal Dankar  
CHEO Research Institute  
401 Smyth Road  
Ottawa, Ontario  
1-613-737-7600  
fdankar@ehealthinformation.ca

Khaled El Emam  
CHEO Research Institute  
401 Smyth Road  
Ottawa, Ontario  
1-613-737-7600  
kelemam@uottawa.ca

ABSTRACT
Differential privacy has gained a lot of attention in recent years as a general model for the protection of personal information when used and disclosed for secondary purposes. It has also been proposed as an appropriate model for health data. In this paper we review the current literature on differential privacy and highlight important general limitations to the model and the proposed mechanisms. We then examine some practical challenges to the application of differential privacy to health data. The review concludes by identifying areas that researchers and practitioners in this area need to address to increase the adoption of differential privacy for health data.

Categories and Subject Descriptors
K.4.1 [Computers and Society]: Public Policy Issues — Privacy

General Terms
Documentation, Security, Human Factors.

Keywords
Differential privacy, Microdata, Anonymity, De-identification, Randomization.

1. BACKGROUND
We consider private data analysis in the setting of a trusted data custodian that has a database consisting of rows of data (records). Each row represents information about an individual. The custodian needs to publish an anonymized version of the data, a version that is useful to data analysts and that protects the privacy of the individuals in the database.

Existing privacy models depend on the background knowledge of adversaries; these are attackers who might compromise the data by re-identifying it. Because of that dependence, disclosed data are not immune to attacks resulting from unforeseen auxiliary information. Differential privacy is a fairly new privacy model that is gaining popularity. Its appeal is that it makes almost no assumptions about the attacker’s background knowledge (however, minor assumptions about how the data is generated must be made [33]). Differential privacy tries to ensure that the removal or addition of any record in the database does not change the outcome of any analysis by much. In other words, the presence of an individual is protected regardless of the attacker’s background knowledge.

A previous attempt at a context-free privacy model was made by Dalenius in 1977. He articulated a privacy goal for statistical databases: anything that can be learned from the database about a specific individual should be learned without access to the database [6]. Attempts to formalize Dalenius’ notion centered around measuring the adversary’s prior and posterior beliefs about a specific record in the database, making sure that the change is small. This however contradicts the goal of database release, which is to change or form beliefs about people/situations. An example illustrating this is the Terry Gross height example from [10]:

“Suppose one’s exact height were considered a highly sensitive piece of information, and that revealing the exact height of an individual were a privacy breach. Assume that a database yields the average heights of women of different nationalities. An adversary who has access to the statistical database and the auxiliary information ‘Terry Gross is two inches shorter than the average Lithuanian women’ learns Terry Gross’s height, while anyone learning only the auxiliary information learns relatively little.”

According to Dalenius’ notion of privacy, learning Terry Gross’s height does constitute a privacy breach even if Terry Gross did not participate in the database. That motivated Dwork to come up with a different formulation, differential privacy: “the risk to one’s privacy...should not substantially increase as a result of participating in a statistical database” [10]. Thus an attacker should not be able to learn any information about any participant that she can not learn if the participant opts out of the database. Going back to the example above, since Terry Gross’s height can be learned without her participation in the database, it does not constitute a breach of privacy.

Differential privacy is growing rapidly and gaining popularity in the computer science literature. It is even becoming challenging to justify conducting research on and making improvements to other privacy models. That partially led to extreme opinions around the differential privacy notion, from being completely inadequate and can never be used in real life situation, to being the best and only viable privacy notion. It is for these reasons that we would like to present an overview of the current state of art on differential privacy and study its applicability to the disclosure of health data. In doing so, we
identify the areas in differential privacy that need to be explored further before it can be applicable in the health field.

In private data analysis, two settings are possible: interactive and non-interactive. In the interactive setting where the database is held by a trusted server, users pose queries about the data, and the true answer to the queries is modified to protect the privacy of the database participants. In the non-interactive setting the data custodian either computes and publishes some statistics on the data, or releases an anonymized version of the raw data.

In the first part of this paper, differential privacy is formally defined along with some of its relaxations. Section 3 presents the basic and most common ways to achieve differential privacy, then the leading research in the two data sanitization settings: interactive and non-interactive is presented and evaluated in Section 5, followed in Sections 6 and 7 by some conclusions on the applicability of differential privacy on health data.

2. Differential Privacy: The Principles

Generally speaking, differential privacy requires that the answer to any query be “probabilistically indistinguishable” with or without a particular row in the database. In other words:

Given an arbitrary query \( f \) with domain \( \mathcal{P} \) and range \( P \) (\( f : \mathcal{P} \rightarrow P \)), and two databases \( D \) and \( D' \), drawn from population \( \mathcal{P} \), that differ in exactly one record, if \( K_f \) is a randomized function used to respond to query \( f \), then \( K_f \) gives \( \varepsilon \)-differential privacy if for any \( s \in \text{Range}(K_f) \)

\[
\Pr[K_f(D) \in s] \leq e^\varepsilon \Pr[K_f(D') \in s]
\]

In [1], the authors interpret the ratio \( \frac{\Pr[K_f(D) \in s]}{\Pr[K_f(D') \in s]} \) as: “knowledge gain ratio from one data set over the other”. Differential privacy requires that the knowledge gain be bounded by \( e^\varepsilon \). Even if the participant removed her data from the database, the limited knowledge gain implies that no output would become significantly more or less likely [11]. The parameter \( \varepsilon \) is public. Values typically are 0.01, or 0.1, and sometimes \( \ln 2 \) or \( \ln 3 \) [10], [11].

3. Achieving Differential Privacy

The notion of differential privacy has led to the introduction of several mechanisms that preserve it. The development of further mechanisms is an active area of research. In this section and in Sections 5 we outline the leading mechanisms in this area. In order to satisfy differential privacy for a query output, [9] suggests the use of Laplace distributed noise: if \( r \) is the correct answer to a query \( f \) on a database \( D \) (i.e. \( f : \mathcal{P} \rightarrow P \), \( D \in \mathcal{P} \) and \( f(D) = r \)), then a differentially private mechanism would output the response \( r + y \), where \( y \) is the noise drawn at random from a Laplace distribution with mean 0 and scale \( \Delta f \). \( \Delta f \) represents the maximum value for \( ||f(D') - f(D)|| \) for all \( D', D \in \mathcal{P} \) differing in one row.

\( \Delta f \) is the global sensitivity of the query \( f \) over domain \( \mathcal{P} \), it is independent of the database \( D \), and one must consider all databases \( D' \in \mathcal{P} \) to determine its value. This is unlike the traditional noise addition where the noise is proportional to the variance of the data. That shows a fundamental characteristic of differential privacy which is that it seeks to protect even extreme values; note that if the data is skewed, this results in high sensitivity, and the noise would be large compared to the variance of the data, leading to non-useful output [42].

The above mechanism can not be applied for a query whose outcome is not real, as adding noise to non-real values does not make sense. In [37], the authors propose the exponential mechanism where they assume the existence of a score function that evaluates the quality of the query output (the score function could be the utility of the output): Given a query \( f \) with range \( P \), the mechanism assigns probabilities to the different elements in \( P \) based on their score, a higher score means a more desired output and hence a higher probability. If such a score function exists, then a differentially private output could be produced based on the sensitivity of the score function. The authors argue that, similar to the Laplace mechanism, if the sensitivity of the score function is low, then high quality output can be obtained. Alternative mechanisms that satisfy differential privacy have recently surfaced. These will be discussed in Section 5.

4. Variants of Differential Privacy

A relaxed version of differential privacy, described in [32], \((\varepsilon, \delta)\)-differential privacy requires that \( \varepsilon \)-differential privacy be satisfied with a probability of at least \( 1 - \delta \), in other words, \( \varepsilon \)-differential privacy can be violated for some tuples, however the probability of that occurring is bounded by \( \delta \). Formally:

For any two database; \( D \) and \( D' \), drawn from a population \( \mathcal{P} \), that differ in exactly one record, if \( K_f \) is a randomized function used to respond to an arbitrary query \( f \), then \( K_f \) gives \((\varepsilon, \delta)\)-differential privacy if with probability \( 1 - \delta \), for any \( s \in \text{Range}(K_f) \) the following holds:

\[
\Pr[K_f(D) \in s] \leq e^\varepsilon \Pr[K_f(D') \in s]
\]

An alternative relaxation, \((\varepsilon, \tau)\)-differential privacy, requires that for any \( s \in \text{Range}(K_f) \)

\[
\Pr[K_f(D) \in s] \leq e^\varepsilon \Pr[K_f(D') \in s] + \tau
\]

\((\varepsilon, \tau)\)-differential privacy is similar to \( \varepsilon \)-differential privacy but allows an additive error factor of \( \tau \) [41].

In what follows, we introduce the definition of usefulness that is adopted in the differential privacy literature as well as the main queries considered: predicate and counting queries:

A mechanism is \((\alpha, \delta)\)-useful if every query output is within \( \alpha \) of the correct output with a probability of at least \( 1 - \delta \), i.e.

\[
\Pr[|K_f(D) - f(D)| \leq \alpha] \geq 1 - \delta.
\]

Count queries count the number of individuals in the database that satisfy a certain predicate (example: how many individuals have grey hair and are under 40 years?). It follows that the sensitivity of a counting query is 1.

Predicate queries are normalized counting queries, in other words they count the fraction of individuals in the database that satisfy a certain predicate. Note that the sensitivity of a predicate query is \( 1/n \) when the database is of size \( n \).
5. MECHANISMS OF DIFFERENTIAL PRIVACY

5.1 Interactive Differential Privacy

5.1.1 Mechanisms

Before presenting the interactive mechanisms, we present a negative result concerning privacy from noise addition:

**Result 1:** Dinur and Nissim [7] showed that in a database consisting of bits, if a user wants to know the sum of some random subsets of these bits, then no private mechanism can provide accurate answers for many queries. In fact, no private mechanism can answer $n$ queries with error $O(\sqrt{n})$, because an adversary would be able to use the queries’ outputs to reconstruct $1-O(1)$ fraction of the database, a condition referred to as “blatant non-privacy”.

Dwork introduced the first and most commonly used mechanism that provides $\varepsilon$-differential privacy in query-response situations, it is Dwork’s independent Laplace mechanism [9]. In this mechanism, the data curator adds appropriate amounts of noise to each query answer as follows:

Noise is drawn at random from a Laplace distribution with mean 0 and scale $\Delta f/\varepsilon$ and added independently to each query response (Section 3), thus making sure that every query is perturbed appropriately. However since each query answer leaks more information and reduces privacy, what is the privacy level of several query outputs taken together? Dwork proved that the mechanism is composable, i.e. given a sequence of differentially private computations: $f_1, ..., f_k$ assigned budgets privacy $\varepsilon_1, ..., \varepsilon_k$ respectively (i.e. the noise added to the output of $f_i$ is drawn at random from a Laplace distribution with mean 0 and scale $\Delta f_i/\varepsilon_i$), then the overall computation has a worst case parameter of $\sum \varepsilon_i$ [9].

Given its composability, two standard approaches exist for the above mechanism depending on whether the queries are known ahead of time or not:

- If the queries $f_1, ..., f_k$ are known ahead of time, then the standard approach is for the data custodian to assign a total privacy budget $\varepsilon$, the user then can divide the privacy budget among the different queries as needed (more important queries can have higher budget and hence be less noisy). However it is important to note that the type of analysis (queries) that is of interest to the analyst is not usually known ahead of time, in fact sanitized data are usually shared by several data analysts with different goals.

- On the other hand, if the queries are chosen adaptively, i.e. if each query is formulated after obtaining the result of the previous query/queries (which is a more realistic scenario), then the approach is for the data custodian to have a preset privacy budget $\varepsilon$. That budget will decrease every time a query is answered, and the curator will keep on providing query answers until the budget runs out. Since different users of the database may collude, there should be one privacy budget for all users [11][10]. Once the budget runs out, the database has to shut down.

Now, assume that each query $f_i, i \in \{1, ..., k\}$ has sensitivity $\Delta$, such that $\Delta \leq 1$, and that each is assigned an equal budget $\varepsilon/k$. If a fixed utility is required (supplied by fixed parameters $\alpha, \delta$), then this sets a bound on the magnitude of noise allowed per query. Since the noise magnitude is a function of $\varepsilon$ and $k$ (Laplace($\Delta k/\varepsilon$)), then the noise constraint shows the interdependence between the two quantities: $\varepsilon$ and $k$.

In Dwork’s independent Laplace mechanism this interdependence is as follows:

**Result 2:** Assume that the user is interested in predicate queries $f_1, ..., f_k$ (or any set of low sensitivity queries such as sum or linear queries), and in having $\alpha$-usefulness, if Laplace noise is added independently to every query output (Dwork’s mechanism), then the privacy parameter $\varepsilon$ (and hence the noise magnitude) grows linearly with the number of queries $k$ answered [9].

The result implies that in order to have sublinear noise (noise to the order $O(n^c)$ with $c<1$) the mechanism can not answer $O(n)$ queries.

The mechanism above has bad implications on several fronts as described in [35]: When the number of users is large, the budget per user becomes limiting as lower budget requires larger noise to be added. Moreover, the issue of dividing this budget among users presents another difficulty. Even when a small number of users is expected, the budget can be limiting to creative research (like experimenting with different hypotheses). That drove research in the interactive setting to concentrate on ways to reduce the noise magnitude per query, thus allowing more queries per budget:

A line of research by Blum et al [4] tried to circumvent Result 1 by creating a mechanism that can answer arbitrary many queries while preserving differential privacy. The catch is that, the queries whose output is useful belong to a predefined “class of queries”:

Let $R$ be a discrete set of data items, let $D$ be a database of size $n$ whose rows are drawn from $R$ ($D \in R^n$), now let $C$ be a “concept” class, that is a set of computable functions from $R \rightarrow \{0, 1\}$. The authors use the exponential mechanism [37] to create a synthetic database $D'$ that approximates outputs corresponding to all concepts in $C$. In fact, for fixed usefulness parameters, the privacy parameter $\varepsilon$ grows proportional to the VC-dimension of $C$, a measure of the complexity of $C$, which is approximately $\log_2 |C|$. The result implies that utility can be guaranteed even for an exponential number of queries, however the mechanism works only for discrete databases ($R$ was assumed to be discrete), generalizing it to non-discrete databases comes at the expense of its usefulness [4], moreover it is non-efficient, in fact, it is superpolynomial in the parameters of the problem (i.e. it is not bounded above by any polynomial), making it impossible for practical applications. Another drawback is that the mechanism...
solves the problem for the interactive database release only when the type of analysis that is of interest to the analyst is known ahead of time (set $C$ should be known before $D'$ can be generated).

In an attempt to overcome some of the above limitations, Dwork et al [12] presented a mechanism that is $(\varepsilon, \tau)$-differentially private but with an improved running time (polynomial in $|R1|$ and $k$). The mechanism’s utility is better than the independent Laplace mechanism: for fixed usefulness parameters, the privacy parameter $\varepsilon$ grows linearly with $O(\sqrt{2k\log k})$. However, the result applies only in the non-adaptive case, i.e. the queries need to be known ahead of time.

Similarly, in [29] and [36] the authors addressed the issue of reducing noise and allowing more queries per a given budget, however the techniques require that the all queries be known ahead of time.

In [41], the authors present a novel non-efficient mechanism, the median mechanism, that interactively answers $k$ predicate queries $f_1, ..., f_k$ as they arrive while guaranteeing $(\varepsilon, \tau)$-differential privacy. Their aim was to ameliorate the limitation on the number of predicate queries that can be answered in an interactive setting. The mechanism does that by determining the correlation between different queries on the fly. In essence, the authors prove that queries can be divided into hard and easy queries, where hard queries completely determine the answers to the easy queries (up to a small error). In fact, they proved that, given a set of queries $f_1, ..., f_i$, and their outputs $r_1, ..., r_i$, and given an easy query $f_i$, the majority of databases that produce consistent results on the previous query answers $r_1, ..., r_i$, would answer the current easy query $f_i$ accurately, thus the output of $f_i$ would be deducible from $r_1, ..., r_i$, and hence does not use the privacy budget. Moreover, they proved that among any set of $k$ queries, there are $O(k\log|X|)$ hard queries and the rest are easy queries. The hard queries are answered using Dwork’s independent Laplace perturbations, while the easy queries are answered using the median result from databases that are consistent with previous answers. The classification of queries into hard and easy is done at a low privacy cost. The authors proved the following result:

Result 4: For fixed usefulness parameters, the privacy parameter $\varepsilon$, grows linearly with $O(\log k\log|X|)$ where $X$ is the database domain.

That means that the median mechanism can answer exponentially more predicate queries than Dwork’s independent Laplace mechanism. More importantly, the privacy and utility guarantees hold even when the queries are chosen adaptively. However the setback is that the algorithm is non-efficient, its run time is exponential in the problem’s parameters, and is hence unusable in practice. For more information, the reader is referred to [41].

Recently, the authors in [30] presented a new efficient mechanism, the multiplicative weights mechanism, it achieves $\varepsilon$-differential privacy (rather than $(\varepsilon, \tau)$) and for fixed usefulness parameters, the privacy parameter $\varepsilon$ grows linearly with $O(\sqrt{k})$.

In other words the algorithm achieves $O(\sqrt{k})$ noise per query, while the independent Laplace mechanism achieves $O(k)$, moreover it is efficient and works for adaptively chosen predicate queries.

5.1.2 Discussion

In the interactive setting, the challenge is to find a differentially private mechanism that can (a) answer random queries (any type of queries), (b) answer a large number of queries while providing non-trivial utility for each of the queries, (c) be efficient (d) achieve $\varepsilon$-differential privacy (the stronger privacy guarantee) and (e) answer queries adaptively. As discussed in the previous sub-section, many algorithms attempted to achieve some of the above requirements only to fail in others:

Dwork’s independent Laplace mechanism [9] achieves requirements (a), (c), (d) and (e), Blum et al presented an elegant mechanism in [4], that achieves (b) and (d), Roth and Rothshgarden presented the median mechanism [41] that achieves (b), (d) and (e), and finally, Hardt and Rothblum presented the multiplicative weights mechanism [30] that achieves (b), (c), (d) and (e).

Therefore, the result in [4], [41] and especially [30] are striking in their generality, and succeeded considerably in relaxing the limit on the number of queries per database. However their applicability is limited to counting/predicate queries and only [30] can be used in general ([4] and [41] can only be used with small size problems). On the other hand, the issue of assigning a budget for a database and the issue of dividing this budget among the users was not tackled in any of these papers and still presents another challenge in practical problems [35].

5.2 Non-Interactive Differential Privacy

In this sub-section, we survey some of the available mechanisms for non-interactive release. The mechanisms are divided into ones that rely solely on noise addition, and the more recent ones that do not.

5.2.1 Noise-Based Mechanisms

Result 1 has bad implications also on the non-interactive approach, because in the non-interactive approach, one expects to receive accurate responses to all queries.

As mentioned in sub-Section 5.1, Blum, Ligett and Roth [4] studied database releases from a learning theory perspective. Given a database $D$ and a class of concepts $C$ (as defined in 5.1), the authors presented a mechanism to release a synthetic database that is useful for all queries in class $C$. However, the mechanism works only for discrete databases, is non-efficient, and requires that the class $C$ of queries be known ahead of time.

The author in [11] studied the problem of histogram release while satisfying differential privacy. The histogram to be released is treated as the output of a specific query with sensitivity 1, the problem is the following:

Assume we have a database with $n$ rows each describing one individual. And that each row consists of $k$ binary attributes: $a_1, ..., a_k$. Then the database can be transformed into a contingency table, also known as a table of counts or histogram, it describes, for each setting of the $k$ attributes (which amounts to $2^k$ settings), the number of rows satisfying this setting, Table 2 shows the contingency table generated from the example shown in Table 1.
The author shows how to add noise to each entry of this contingency table to make it differentially private:

Since the contingency table is a histogram, its sensitivity is 1 (the addition or removal of any database row affects the counts in one location by at most 1). Hence we can add independently generated noise to the $2^k$ cells of the contingency table with distribution $\text{Laplace}(1/\epsilon)$.

The problem with this approach is that the amount of noise generated when computing marginals, i.e. queries that involve a large number of entries, (refer to Table 3) could be large [11], [39]. For example, the variance of the 1-way table described by any one attribute, is $2^{k-1}\epsilon^2$. This is considered unacceptable especially when $n \ll 2^k$ [11]. In other words, for a count query that involves the sum of a fraction of the entries in the noisy contingency table, the noise variance could be around $O(2^k)$. If the attributes $a_i$ are not binary and have domains of sizes $\alpha_1, \ldots, \alpha_k$ respectively, then the contingency table size would be $\alpha_1 \alpha_2 \ldots \alpha_k$ which could be huge as databases often have several attributes with large domains.

Alternatively, in [11], the author suggests that if low order marginals are sufficient for data analysts (these are smaller tables of count projected to a subset of the attributes, Table 3 shows an example of the marginal for the two attributes: age and HIV), then the data curator can release a set of say $C$ marginals. Each marginal has sensitivity 1, hence the amount of noise added to each cell of the released marginals should be proportional to $\sqrt{C}/\epsilon$, when $n$ is large compared to $\sqrt{C}/\epsilon$, the authors suggest that this will yield good accuracy per cell. However, with this approach, there will be inconsistencies between the different marginals as noise is added independently.

Another (non-efficient) alternative introduced by [3] is to construct a synthetic database that is positive and integral and that preserves all low order marginals up to a small error using Fourier transforms.

In [24], the authors evaluate the mechanism of [3] using three real-life databases. They concluded that the method is not suitable for the large and sparse tables that are usually produced by agencies and stated a preference for other existing methods such as that of [25].

5.2.2 Alternative Mechanisms

Existing techniques that satisfy differential privacy rely almost exclusively on noise addition to query outputs. However, few alternatives that rely on other means in addition to noise have recently surfaced:

In [5] the authors introduced a new efficient mechanism to release set value data while satisfying $\epsilon$-differential privacy and $(\alpha, \delta)$-utility for counting queries (where the value of $\delta$ is a function of $\epsilon$ and $\alpha$). Set-value data consists of a set of records; each record consists of a set of items drawn from a universe of items, an example is transaction data. The authors introduced a probabilistic top down partitioning algorithm. The algorithm starts by creating a context-free taxonomy tree and by generalizing all items under one partition. The algorithm then, recursively, generates distinct sub-partitions based on the taxonomy tree. The sub partitions have more specific representations, thus splitting the records into more specific disjoint sets. The choice of which partition to split is based on the noisy partitions counts. Only (probabilistically) non-empty partitions are chosen, thus limiting the overall noise added and making this approach appealing when data is sparse. The authors experimentally evaluated their approach against the histogram approach presented earlier [11] and showed that it provides better utility for counting queries. However the authors did not justify their utility parameters, and did not study the interdependence between the utility and privacy parameters. Moreover, the mechanism in [11] suffers from high noise variance for set value data (being sparse and high dimensional) and that renders its results of limited usefulness7, and is therefore not a good standard for comparison.

In [39], the authors argue that contingency table release is not suitable for data with high dimension and large domains as the noise would be relatively large compared to the cell counts, and that leads to a total utility destruction. The authors then present a novel method that generalizes the contingency table then adds noise to the counts. The authors argue that the generalization step increases the cell counts and thus the counts become much larger than the noise added. The algorithm initially generalizes all records into one group, then iteratively implements a sequence of specializations. At each iteration, the algorithm probabilistically selects an attribute to specialize based on some score value (such as information gain). The algorithm terminates after a preset number of specializations. The authors performed a comparison between their algorithm and the top-down specialization approach of [27] that produces $k$-anonymous data. The comparison was done for a fixed value of $k = 5$, and for $\epsilon$ values 0.75, 1, 2, 3, and 4. They deduced that their algorithm performs slightly better for $\epsilon$ values 0.75 and 1, and noticeably better for the remaining $\epsilon$ values. However the authors did not justify their choice for these $\epsilon$ values, and why these particular values should be compared to a 5-anonymity. And although the authors state that a typical $\epsilon$ value is less than 1 -in fact the recommended $\epsilon$ value is 0.01, or 0.1 [26], [11] - they did not use these recommended values in their evaluation. Moreover, the authors did not explain how to decide on an optimal generalization that would produce counts that are high enough relative to the added noise given the privacy budget $\epsilon$.

Recently, the authors in [35] introduced a novel technique connecting $k$-anonymity to differential privacy. The technique tries to achieve $(\epsilon, \delta)$-differential privacy by adding a random sampling step followed by a “safe” $k$-anonymization. The authors show that when a random sampling step is applied to the data followed by a generalization scheme that does not depend on the database (like applying a fixed generalization scheme)
followed by the suppression of all tuples that occur less than \( k \) times, then the database satisfies \((\varepsilon, \delta)\)-differential privacy.

The problem with this technique is that the usage of a data independent generalization may result in poor utility, in [47] the authors argue that adding a random sampling step impacts the utility of the data as well.

Recently, several studies on the application of differential privacy have emerged. However these studies concentrate on particular kinds of data such as search logs [28], [34], on specific applications such as recommender systems [38], and record linkage [31], or on certain type of low sensitivity queries [23], [43], [45].

5.2.3 Discussion

In the non-interactive setting, researchers expect the data release to answer all their queries accurately. Moreover, they prefer non-synthetic data; in fact researchers and statisticians in several fields, such as the health sector, like to “look at the data”. Dwork mentions in [13] that “conversations with experts in this field involve pleas for a noisy table that will permit highly accurate answers to be derived for computations that are not specified at the outset”. So in general, the challenge is to release a non-synthetic dataset that can accurately answer any query, and the release mechanism needs to be efficient.

However, no differentially private interactive mechanism can achieve all of the above:

- In [4], [12], and [30] the release can be used for a set of predicate queries that should be known in advance. The data released is synthetic, and contrary to the intuition on non-interactive data, the utility of the data suffers with the size of the query set.
- In [11], [3], the release can be used for a set of count queries, and the utility suffers when databases are large and sparse [25].
- In several other mechanisms, usability is only guaranteed for particular data types along with particular set of queries such as count queries on set value data [5].

In fact Result 1 indicates that no efficient algorithm can output a noisy table with acceptable utility to random queries. This result clearly states that the general applicability of differential privacy will always be limited to a class of queries. Current research presented big advances in the mechanism efficiency and in the size of the query set with utilisable outputs. However, applications remain limited to queries with low sensitivity such as count and predicate queries.

5.3 Limitations of Differential Privacy

In line with the previous discussions, the literature has concentrated on the application of differential privacy on count data. In fact, the options available in differential privacy limit the user to a number of queries with low sensitivity (\( \Delta f \leq 1 \) regardless of the database \( D \)). Low sensitivity means that the output of these queries is not “severely affected” by the removal or addition of one record.

What about other queries? Recent articles [40] and [42] raise serious concerns about the utility of differentially private algorithms when applied to numeric data. They presented several examples on the application of Laplace noise for numeric data and showed that the level of noise added can be so large making the responses useless. In fact the level of noise is directly affected by the value of \( \Delta f \) and is independent of the actual database. So when the data is skewed, this will result in a large \( \Delta f \) value and subsequently a large noise variance (compared to the actual variance of the database). The authors concluded that “Like Dalenius definition of privacy before it, differential privacy is an interesting concept, but of little value in practice” [40].

The limited available experiments on differentially private mechanisms suggest that in general - for numerical data - very low privacy is required to have acceptable utility [10][2]. In [8], the author suggests that the current formulation of differential privacy focuses on achieving privacy rather than on preserving utility, and, as mentioned before, that is due to the fact that differential privacy seeks to protect even extreme values in the database domain (through \( \Delta f \) ).

Another limitation was mentioned by Muralidhar and Sarathy [40][42]; the authors stressed the difficulty in calculating queries’ sensitivity in unbounded domains. That raises a question about not only the sensitivity calculation but also about differential privacy verification as well. These two questions have been neglected in the literature, as illustrations are limited to queries with low and domain-independent sensitivities.

Another limitation is deciding on a value for \( \varepsilon \). In fact, the degree of distortion to the query output has not been adequately covered in the differential privacy literature. The literature on the privacy parameter \( \varepsilon \) is mostly theoretical, and there exists no experimental evaluations to guide the user on choosing the right \( \varepsilon \) value, or to help the user understand what effect changing \( \varepsilon \) will have on the common notions of data utility. In [46], the author complains that their research team is running against the hurdle of choosing a good value for \( \varepsilon \) and quantifying the “indistinguishability” language in the differential privacy guarantee.

6. DIFFERENTIAL PRIVACY AND HEALTH DATA

The disclosure of health data has a number of characteristics that need to be considered in any practical mechanism used to preserve privacy. These characteristics have to do with current practices and data sets, and the introduction of any new mechanism for privacy protective data disclosure or analysis would have to address these issues before it can be adopted widely. The considerations below are driven by our experiences creating health data sets for secondary purposes over the last seven years, some of which have been documented, as well as empirical studies of health data sharing practices and challenges [14][15][16][17][18][19][20][21].

Health data contains categorical data (e.g., diagnosis codes, procedure codes, drugs dispensed, laboratory tests ordered, and geographical information about the patient and the provider), as well as numeric data (e.g., age in years, length of stay in hospital, and time since last visit). Therefore both types of variables need to be addressed. As noted earlier, the addition of Laplace noise to numeric data can distort the values significantly.

Users of health data are accustomed to data publishing – which is where the data is disclosed to the end user. There are multiple reasons. Health data is often messy, with data errors and sometimes unexpected distributions. Analysts need to look at the data to determine the appropriate transformations to apply, compute appropriate indices from the original data, and extract the appropriate cohort of patients for the analysis. This is easiest to do when one has access to the data directly. Furthermore, biostatisticians and epidemiologists will often have a suite of analysis methods and tools that are commonly used, that they have
used for many years, that they understand, for which they can interpret the results correctly, and for which they have code in languages such as SAS that they use. From a behavior change perspective, it would be challenging to convince data analysts to abandon their current methods and SAS code, which they may have been using for decades, in favor of an interactive system that is less understood. Therefore, at least in the short term, a non-interactive mechanism would be most suitable for this community.

The non-interactive mechanism that allows the computation of statistics without publishing the data may be suitable in some circumstances. For example, in the context of public health, ongoing surveillance often relies on the computation of a well defined (and known a priori) set of statistics at regular intervals. Similarly, performance or safety reporting (e.g., the number of eligible patients that received appropriate screening and the rate of surgical site infections) involves the computation of well defined statistics. Therefore, for surveillance and reporting purposes differentially private statistics would be congruent with the process. In such cases the primary consideration would be data utility of the differentially private statistics.

Beyond such applications, with well defined analytical needs, many other data uses would require actual data publishing to meet the needs of the analyst community.

Another important consideration is the law. The healthcare sector often has specific privacy laws in many jurisdictions. Current health privacy statutes in the US, Canada, and Europe do not specify the acceptable risk and often use the “reasonableness” standard. In practice, one relies on precedent to justify the risk thresholds that are used. For currently used privacy models, such as $k$-anonymity, there is a significant amount of precedent for different values of $k$. Data custodians have been releasing data for more than two decades, including health data. During that period guidelines, policies, court cases, and regulatory orders have come out which define what can be considered acceptable levels of risk. A data custodian, if confronted in a court case or by a regulator for example, can point to these precedents to justify their decisions. In the case of differential privacy, important parameters such as $\varepsilon$ have no intrinsic meaning and there are few existing precedents of actual health data releases to justify the choice of any value. A data custodian needs to consider how they would justify their choice in a dispute, and it is much easier to do so under current models, such as $k$-anonymity, and risky (financially and reputationally) under differential privacy.

Many fields in health data sets are correlated or have natural constraints. For example, one treatment would often precede another, or certain drugs given in combination. There are correlations among drugs, and diagnoses, and between lab results and diagnoses. Distortions to the data that produce results that do not make sense erode the trust of the data analysts in the data and act as barriers to the acceptability of the techniques used to protect the privacy of the data. For example, if the distorted data shows two drugs that are known to interact in a way that can be damaging to a patient’s health, a drug that would never be prescribed with a particular treatment appear for the same patient, or a dose that does not make sense for a patient, then the analysts will cease to trust the data. In practice this has created challenges for introducing mechanisms that add noise to data because it is not possible to guarantee that nonsense data cannot be output.

Reference deployments of differential privacy in practice are also important. A powerful argument in convincing data analysts to use a data set that has been transformed in some way to deal with privacy concerns is to show them actual examples where such data has produced useful and valid results. To our knowledge, thus far there have been limited real world disclosures of differentially private health data, and consequently few examples of useful and valid analytical results.

While not often explicitly considered when designing new mechanisms to protect data, convincing the public that stewardship of their data is being conducted in a responsible way is becoming a necessary objective. For instance, patients and providers have expressed concerns about the disclosure and use of health information, and there is evidence that patients adopt privacy protective behaviors when they have concerns about how their own information is being used or disclosed, especially among vulnerable patient groups [22]. In practice this means there is an on-going need to explain in non-specialist terms the parameters of the privacy mechanisms used and how much protection they really provide. In the context of differential privacy, it is quite challenging to explain to a patient the meaning of the $\varepsilon$ value used to disclose or provide analytical access to their data, for example. It is necessary to relate these parameters to more common notions to allow easier communication to the public.

While the above observations are limited by our experiences, we believe they represent real challenges that the differential privacy model and mechanisms need to address to ensure wider acceptability and adoption within the health domain.

7. CONCLUSIONS

In this paper we have provided a general overview of the state-of-the-art in differential privacy and outlined some of the limitations of the model and the various mechanisms that have been proposed to implement it. We have further highlighted some practical limitations to the use of differential privacy for the disclosure of health information today. At the same time, the highlighted limitations identify elements of future research programs that could potentially lead to more adoption of differential privacy in healthcare.

The healthcare community still needs to disclose data today. Therefore, until the theoretical and practical limitations, as well as the healthcare specific considerations have been addressed, it would be prudent for data custodians to continue using current methods for anonymizing their data.

8. ACKNOWLEDGMENTS

Our sincere thanks to Marina Sokolova, Elizabeth Jonker, and Luk Arbuckle for their helpful and insightful comments on earlier versions of this paper.

9. REFERENCES


[46] Yin M.