Journal of Privacy and Confidentiality (2012) **4**, Number 1, 5–49

An Axiomatic View of Statistical Privacy and Utility

Daniel Kifer[*] and Bing-Rong Lin[†]

Abstract. "Privacy" and "utility" are words that frequently appear in the literature on statistical privacy. But what do these words really mean? In recent years, many problems with intuitive notions of privacy and utility have been uncovered. Thus more formal notions of privacy and utility, which are amenable to mathematical analysis, are needed. In this paper we present our initial work on an axiomatization of privacy and utility. We present two privacy axioms which describe how privacy is affected by both post-processing data and by randomly selecting a privacy mechanism. We present three axioms for utility measures which also describe how measured utility is affected by post-processing. Our analysis of these axioms yields new insights into the construction of privacy definitions and utility measures. In particular, we characterize the class of relaxations of differential privacy that can be obtained by changing constraints on probabilities; we show that the resulting constraints must be formed from concave functions. We also present several classes of utility metrics satisfying our axioms and explicitly show that measures of utility borrowed from statistics can lead to utility paradoxes when applied to statistical privacy. Finally, we show that the outputs of differentially private algorithms are best interpreted in terms of graphs or likelihood functions rather than query answers or synthetic data.

1 Introduction

Statistical privacy is the art of designing a privacy mechanism that transforms sensitive data into data that are simultaneously useful and non-sensitive. The sensitive data typically contain private information about individuals (e.g., income, medical history, search queries) or organizations (e.g., intranet network traces, customer records) and are usually collected by businesses (e.g., Netflix, AOL) or government agencies (e.g., U.S. Census Bureau).

Non-sensitive data produced by privacy mechanisms are highly desirable because they can be made available to the public without restrictions on access. Researchers will benefit from previously unavailable data—they could, for example, study socio-economic and business trends, develop new models, and design and evaluate new algorithms using such data.

All of this potential success hinges on two poorly-defined words: *privacy* and *utility*. What does it mean for a privacy mechanism to output a dataset that is non-sensitive?

[*] Department of Computer Science & Engineering, Penn State University, University Park, PA, mailto:dkifer@cse.psu.edu

[†] Department of Computer Science & Engineering, Penn State University, University Park, PA, mailto:blin@cse.psu.edu

What does it mean for a privacy mechanism to output a dataset that has high utility (i.e., is useful)? The literature is full of definitions of what privacy is and is not; it is also full of ways of assigning a numerical score to the utility of a dataset (for recent surveys, see [10, 22]).

However, current privacy definitions and utility measures are typically constructed on the basis of intuition, but intuition alone can lead us astray. Some spectacular privacy breaches (such as demonstrations involving AOL [3], GIC [44], and Netflix [35, 36] data) have occurred when such intuition was not followed by a thorough analysis. In other cases, subtle implicit assumptions created weaknesses that could be exploited to breach privacy [27, 46, 23, 30]. Similarly, the choice of a privacy mechanism based on some intuitively plausible measures of utility can result in a dataset that is not as useful as it could be [37, 24, 25]. For example, Ghosh et al. [24] have shown that if utility is measured by expected loss (in the Bayesian sense) then it is possible that a "suboptimal" privacy mechanism followed by a lossy postprocessing step can mimic an "optimal" privacy mechanism, thus casting doubts on the appropriateness of expected loss. In followup work, Gupte and Sundararajan [25] show a similar result for minimax loss.

Clearly, a unified theory of privacy and utility is needed to guide the development of privacy definitions, utility measures, and privacy mechanisms. We believe that the path to such a theory relies on an axiomatization of privacy and utility. That is, we must examine axioms for what privacy and utility should mean and then study the consequences of those axioms. Not all axioms will be appropriate for all applications. However, when new sensitive data need to be released, a data publisher can pick and choose whatever axioms are appropriate for the application at hand. The chosen axioms would specify a privacy definition and a utility measure. The data publisher can then select an appropriate privacy mechanism (satisfying the privacy definition and maximizing the utility measure) and then use it to generate useful non-sensitive data.

Our vision is that in the future, privacy definitions and utility measures can be constructed using axioms as building blocks. Privacy definitions will become easier to understand. The reason is that a small set of axioms is easier to study thoroughly than an enormous set of privacy definitions (that were not defined axiomatically) and so the strengths, weaknesses, and assumptions behind a privacy definition derived from axioms become more apparent. Privacy definitions built this way would also be more reliable. There would be a long list of axioms simple enough to understand. This list of axioms would serve as a list of privacy concerns and a data curator would choose the axioms appropriate for the given application. Similarly, for utility, axioms would reflect the information needs of intended recipients. Once the data curator chooses the appropriate axioms, the rest (privacy definition, utility measure, choice of privacy mechanism) follows mathematically. Thus, the chance of privacy breach due to erroneous human intuition is decreased (compared to selecting a privacy definition without such guidance); similarly the chance of releasing useless (but non-sensitive) data is also reduced.

This ultimate goal is still far away, but in this paper we present initial steps in this direction. We present two privacy axioms and three utility axioms and we study their consequences. The axioms are fairly simple but have non-trivial implications.

Our first main result answers questions about how differential privacy [14] can be relaxed. Differential privacy is a formal privacy definition that uses a set of predicates to restrict the output probabilities of a privacy mechanism. Relaxations of differential privacy are studied as a way of improving the utility of data that are output from privacy mechanisms (e.g., [15, 38, 32, 9]). These relaxations frequently change the predicates that differential privacy uses. In this paper we characterize the *class* of predicates that can be used (instead of just presenting one or two relaxed definitions). We show that these predicates must be constructed from concave functions and we provide a simple method for generating additional predicates. Visualizations of these predicates show that they intuitively make sense, but our results are based on axioms rather than potentially faulty intuition.

We consider the two privacy axioms to be universal (applicable to almost any privacy definition). They place mild restrictions on privacy definitions and require them to have a minimal level of consistency. It turns out that very few privacy definitions actually satisfy these axioms (which emphasizes the need for a principled, axiomatic approach). Most of the privacy definitions consistent with our axioms are variations of differential privacy. However, we give several examples of existing variations of differential privacy which are not consistent with these axioms.

Our second main contribution deals with utility. We present one universal axiom for utility (that all information-theoretic utility functions should satisfy) called *sufficiency* and then we present two additional axioms. We use the sufficiency axiom to characterize the types of differentially private mechanisms that are desirable in terms of utility. We then show explicitly that expected utility is not a suitable utility metric for privacy mechanisms because it does not satisfy the sufficiency axiom and therefore leads to utility paradoxes (i.e., expected utility can prefer a mechanism that provides absolutely no information). We then provide examples of utility metrics that do satisfy the sufficiency axiom, and then we use our additional axioms to completely characterize a class of utility metrics that we call *branching measures*.

The rest of this paper is organized as follows. We present our privacy axioms and some of their implications in Section 2, where we also characterize a class of relaxations of differential privacy and give examples of privacy definitions that do not satisfy the axioms. Utility is discussed in Section 3. The Axiom of Sufficiency is presented in Section 3.1 and examples of appropriate and inappropriate utility metrics, as well as additional axioms, are presented in Section 3.2. Using the Axiom of Sufficiency, we characterize the desirable differentially private mechanisms in Section 3.3. Proofs of our technical results can be found in the Appendix.

2 Reasoning About Privacy

In this section we demonstrate the benefits of studying privacy in an axiomatic way. We first present two privacy axioms which have been implicitly and explicitly accepted in much of the literature. We believe these privacy axioms to be universal—all privacy definitions should explicitly satisfy them. These two axioms are very mild and for this reason they do not define privacy by themselves. Nevertheless, we show that even these two simple axioms have interesting consequences—they can be used to characterize an entire class of relaxations of differential privacy. The reason we focus on differential privacy is that it explicitly satisfies these axioms, in contrast to many other privacy definitions that do not. To illustrate the axiomatic approach to privacy, we first present a generic form of differential privacy to which the axioms will be applied. We then formally present the axioms in Section 2.1 and explain how they can help characterize relaxations of differential privacy. We then give examples of relaxations of differential privacy in Section 2.2 and also explain how some existing privacy definitions do not satisfy these axioms.

The basic units to which we apply axioms are randomized (and deterministic) algorithms and privacy definitions. We first formalize both notions in an information-theoretic manner.

Definition 2.0.1. (Randomized Algorithm). *Given an input space $\mathbb{I}_{\mathcal{A}}$ and output space $\mathbb{O}_{\mathcal{A}}$, a* randomized *algorithm $\mathcal{A}$ is a conditional probability distribution $P_{\mathcal{A}}(O \mid i)$ where $i \in \mathbb{I}_{\mathcal{A}}$ and $O \subseteq \mathbb{O}_{\mathcal{A}}$.*[1] *Semantically, $P_{\mathcal{A}}(O \mid i)$ is the probability that $\mathcal{A}(i) \in O$.*

Note that deterministic algorithms are special cases of randomized algorithms, where all of the conditional probabilities are either 0 or 1.

In the literature, there are several distinct notions of where privacy comes from. In some cases, privacy is considered to be a property of the algorithm that generates sanitized data [14] and in other cases it is a property of the data that is being output [1]. We unify both approaches with the simple idea that in both cases, the goal is to find an algorithm that produces non-sensitive outputs from some sensitive input data. Thus we define a privacy definition to be the *set* of all algorithms that we trust to perform this process, and our axioms will therefore be statements about what properties such a set of algorithms should possess.

Definition 2.0.2. (Privacy definition, privacy mechanism). *Given an input space $\mathbb{I}$, a* privacy definition *is a set of randomized algorithms with common input space $\mathbb{I}$. We say that these randomized algorithms* satisfy *the privacy definition. A randomized algorithm that satisfies the privacy definition is called a* privacy mechanism.

It is important to note that each input $i \in \mathbb{I}$ corresponds to a possible **dataset** and *not* to a **tuple** in a dataset. The output space $\mathbb{O}_{\mathcal{A}}$ of each randomized algorithm $\mathcal{A}$ satisfying the privacy definition can be different (even though all the input spaces must be the same). An output $o \in \mathbb{O}_{\mathcal{A}}$ could be anything – a set of query answers, synthetic data, or some other object. Thus, this notion captures all possible processes that create sanitized (i.e., non-sensitive) data.

[1] More formally, we equip $\mathbb{I}_{\mathcal{A}}$ and $\mathbb{O}_{\mathcal{A}}$ with σ-algebras and require that $P_{\mathcal{A}}(O \mid i)$ be a regular conditional probability so that $P(\cdot \mid i)$ is a probability measure for each fixed i and $P(O \mid \cdot)$ is a measurable function of i for each measurable $O \subseteq \mathbb{O}_{\mathcal{A}}$.

In this paper we consider the scenario where a *data publisher* possesses sensitive information about individuals. The data publisher would like to release some version of this data without violating the privacy of those individuals. An *attacker* (or a class of attackers) will try to infer the sensitive information from the released data. The data publisher first selects a privacy definition that would defend against a certain class of attackers. Then the data publisher selects a special randomized algorithm known as a *privacy mechanism*, denoted by $\mathfrak{M}$, which satisfies the privacy definition. Finally, the data publisher applies the privacy mechanism $\mathfrak{M}$ to the sensitive data, and releases the output of $\mathfrak{M}$. We will refer to the output of $\mathfrak{M}$ as *sanitized data* to emphasize the fact that it should be safe to release to the public. Note that we will use the symbol $\mathfrak{M}$ to refer to any randomized algorithm that is a privacy mechanism and $\mathcal{A}$ to refer to a randomized algorithm in general.

The privacy axioms that we will discuss in Section 2.1 are not tied to any specific privacy definition. However, we will use those axioms to add insight to the definition known as differential privacy. Thus we discuss differential privacy next.

Definition 2.0.3. (Differential Privacy [14]). *Let $\mathbb{I}$ be a set of database instances and $\epsilon > 0$. A randomized algorithm $\mathfrak{M}$ with output space $\mathbb{O}_{\mathfrak{M}}$ satisfies ϵ-differential privacy if for all (measurable) $O \subseteq \mathbb{O}_{\mathfrak{M}}$ and for all pairs (i_1, i_2) of database instances that differ in one tuple, $P_{\mathfrak{M}}(O \mid i_1) \leq e^{\epsilon} P_{\mathfrak{M}}(O \mid i_2)$.*

The phrase "differ in one tuple" in Definition 2.0.3 can be interpreted in several ways. Dwork [14] interprets this to mean that i_1 and i_2 are neighbors if one can construct the dataset i_2 from i_1 by adding or removing one tuple. In earlier work, Dwork et al. [16] interpreted this to mean that one can construct i_2 from i_1 by changing the value of 1 tuple. Thus in the former interpretation, i_1 and i_2 have different numbers of tuples while in the latter interpretation the number of tuples is the same.

To avoid confusion, we will simply assume that there exists a *neighbor relation* $\mathcal{R}$ which is an irreflexive binary relation $\subseteq \mathbb{I} \times \mathbb{I}$. Thus $(i_1, i_2) \in \mathcal{R}$ if and only if i_1 and i_2 are neighbors (according to whatever interpretation is chosen). This is the first step to generalizing differential privacy. Note that the neighbor relation for differential privacy is symmetric; however, we do not require this symmetry in order to allow the inputs to be treated asymmetrically—we may want to treat a dataset in which Bob is healthy differently from a dataset where Bob has cancer simply because one dataset seems to have "less" sensitive information. Thus in the general case, on could choose a neighbor relation $\mathcal{R}$ such that $(i_1, i_2) \in \mathcal{R}$ but $(i_2, i_1) \notin \mathcal{R}$.

Our second generalization is to replace the condition $P_{\mathfrak{M}}(O \mid i_1) \leq e^{\epsilon} P_{\mathfrak{M}}(O \mid i_2)$ in Definition 2.0.3 with some other *privacy predicate* $\mathrm{q}(P_{\mathfrak{M}}(O \mid i_1), P_{\mathfrak{M}}(O \mid i_2))$. This leads to the following definition:

Definition 2.0.4. (q-Generic Differential Privacy). *Given an input space $\mathbb{I}$, a neighbor relation $\mathcal{R}$, and a privacy predicate $\mathrm{q}(\cdot, \cdot) : [0,1] \times [0,1] \to \{T, F\}$, a randomized algorithm $\mathfrak{M}$ with output space $\mathbb{O}_{\mathfrak{M}}$ satisfies generic differential privacy if for all (measurable) $O \subseteq \mathbb{O}_{\mathfrak{M}}$ and for all $(i_1, i_2) \in \mathcal{R}$ we must have $\mathrm{q}(P_{\mathfrak{M}}(O|i_1), P_{\mathfrak{M}}(O|i_2)) = T$.*

Now that we have a generic form of differential privacy, we can ask what kinds of predicates q are allowable. One such q has already been proposed. It is the condition that $P_{\mathfrak{M}}(O \mid i_1) \leq e^{\epsilon} P_{\mathfrak{M}}(O \mid i_2) + \delta$, for some small δ [15, 38]. The question is what other predicates can be used. For example, would the following make a valid privacy definition?

Definition 2.0.5. (Cosine Privacy.) *Given an input space $\mathbb{I}$ and a neighbor relation $\mathcal{R}$, a randomized algorithm $\mathfrak{M}$ with output space $\mathbb{O}_{\mathfrak{M}}$ satisfies cosine privacy if for all (measurable) $O \subseteq \mathbb{O}_{\mathfrak{M}}$ and for all $(i_1, i_2) \in \mathcal{R}$ we must have $\cos P_{\mathfrak{M}}(O|i_1) = P_{\mathfrak{M}}(O|i_2)$.*

To answer such questions, we need to identify properties we think a privacy definition should satisfy. In Section 2.1 we present two such properties which are simple enough for us to call them axioms. We consider them to be universal in the sense that almost all privacy definitions should satisfy them. They do not define privacy by themselves, but can be thought of as consistency conditions for privacy definitions.

Intuitively, cosine privacy (Definition 2.0.5) should be rejected. However, we will not need to rely on intuition alone, since we will show that the two axioms in Section 2.1 disallow cosine privacy.

2.1 Privacy Axioms

In this section we present two privacy axioms. These axioms are designed to enforce a certain internal consistency for privacy definitions. Our first axiom, Axiom 2.1.1 deals with the effects of postprocessing the sanitized data (this axiom has been observed to hold in differential privacy [14, 26, 2, 47], but we do not tie it to any specific privacy definition). Our second axiom, Axiom 2.1.2 deals with the effects of selecting a privacy mechanism at random.

First, we introduce some notation. Given two randomized algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$, their composition is denoted by $\mathcal{A}_1 \circ \mathcal{A}_2$ and is defined as the application of $\mathcal{A}_2$ followed by $\mathcal{A}_1$ (assuming the output space of $\mathcal{A}_2$ is contained in the input space of $\mathcal{A}_1$). If the randomness in $\mathcal{A}_1$ is independent of the randomness in $\mathcal{A}_2$ and is also independent of the input to $\mathcal{A}_2$, then the resulting conditional distribution $P_{\mathcal{A}_1 \circ \mathcal{A}_2}(Z|x)$ is $\int P_{\mathcal{A}_1}(Z|y) P_{\mathcal{A}_2}(y|x)\ dy$ (the integral is replaced by a summation for discrete output spaces).[2] For any two randomized algorithms $\mathcal{A}_1, \mathcal{A}_2$, we also say that $\mathcal{A}_1 = \mathcal{A}_2$ if the corresponding conditional probabilities are equal: $P_{\mathcal{A}_1}(\cdot \mid \cdot) = P_{\mathcal{A}_2}(\cdot \mid \cdot)$.

Axiom 2.1.1. (Transformation Invariance). *Let $\mathfrak{M}$ be a privacy mechanism for a particular privacy definition and let $\mathcal{A}$ be a randomized algorithm whose input space contains the output space of $\mathfrak{M}$ and whose randomness is independent of both the data and the randomness in $\mathfrak{M}$. Then $\mathfrak{M}' \equiv \mathcal{A} \circ \mathfrak{M}$ must also be a privacy mechanism satisfying that privacy definition.*

Essentially this axiom says that postprocessing sanitized data maintains privacy as long as the postprocessing algorithm does not use the sensitive information directly

[2] In measure-theoretic terms, $P_{\mathcal{A}_1 \circ \mathcal{A}_2} = \int P_{\mathcal{A}_1}(Z|y) P_{\mathcal{A}_2}(dy|x)$.

(i.e., sensitive information is only used indirectly via the sanitized data). This is an information theoretic axiom and does not place any computational restrictions on an attacker (who may be the one providing the postprocessing algorithm $\mathcal{A}$).

This has implications for encryption. If $\mathfrak{M}$ is an algorithm that encrypts the database and $\mathcal{A}$ is the corresponding decryption algorithm, then $\mathcal{A} \circ \mathfrak{M}$ is the identity algorithm (the output is the input) and by Axiom 2.1.1 the identity is a privacy mechanism if the encryption algorithm $\mathfrak{M}$ is (if we do not want our privacy definition to include the identity mechanism, then it should also not use encryption). In this sense, Axiom 2.1.1 is information-theoretic: there are no computational assumptions to worry about. In fact, when some loss of information is acceptable, then storing only a sanitized dataset can be a viable alternative to storing encrypted data—there is no need to worry about physically protecting encryption keys and no need to worry about other cryptographic nuances such as weak keys and potential weaknesses in encryption schemes and their implementations.

On the other hand, one argument for including computational complexity considerations in Axiom 2.1.1 is to allow a combination of data sanitization and secure multiparty computation [34]. We leave this extension to future work.

We shall also make use of the following axiom.

Axiom 2.1.2. (Convexity) *Let $\mathfrak{M}_1$ and $\mathfrak{M}_2$ be privacy mechanisms that satisfy a particular privacy definition (and such that the randomness in $\mathfrak{M}_1$ is independent of the randomness in $\mathfrak{M}_2$). For any $p \in [0,1]$, let $\mathfrak{M}_p$ be a randomized algorithm that on input i outputs $\mathfrak{M}_1(i)$ with probability p (independent of the data and the randomness in $\mathfrak{M}_1$ and $\mathfrak{M}_2$) and outputs $\mathfrak{M}_2(i)$ with probability $1-p$. Then $\mathfrak{M}_p$ is a privacy mechanism that satisfies the privacy definition.*

We can justify Axiom 2.1.2 (convexity) as follows. If $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are privacy mechanisms that satisfy a particular privacy definition, then intuitively either algorithm produces sanitized data that leaves sufficient uncertainty about the true input data. Thus we should be free to use either $\mathfrak{M}_1$ or $\mathfrak{M}_2$ to sanitize our data, as long as this choice does not depend on the true inputs. Taking this argument a step forward, we should be able to randomly choose (with known probabilities) which mechanism we apply to the input. If we do this then, given sanitized data, there are two sources of uncertainty: the uncertainty about which mechanism was used, and the uncertainty injected into the sanitized data by whatever mechanism was actually chosen ($\mathfrak{M}_1$ or $\mathfrak{M}_2$).

Using these two axioms, we can characterize the allowable predicates q that can be used in the generic version of differential privacy (Definition 2.0.4). This is summarized in the following two theorems. First, note that if we replace $\mathrm{q}(a,b)$ in Definition 2.0.4 with the predicate $\mathrm{q}(a,b) \wedge \mathrm{q}(1-a,1-b)$, the privacy definition remains unchanged because we require that $\mathrm{q}(P_{\mathfrak{M}}(O \mid i_1), P_{\mathfrak{M}}(O \mid i_2)) = T$ for all O (including $\overline{O}$, the complement of O) so that $\mathrm{q}(P_{\mathfrak{M}}(\overline{O} \mid i_1), P_{\mathfrak{M}}(\overline{O} \mid i_2)) = \mathrm{q}(1-P_{\mathfrak{M}}(O \mid i_1), 1-P_{\mathfrak{M}}(O \mid i_2)) = T$ must also hold.

Theorem 2.1.3. (Necessary Conditions). q-*Generic Differential Privacy satisfies Axioms 2.1.1 and 2.1.2 only if there exists a function $M : [0,1] \to [0,1]$ such that:*

1. $M(a) > b > 1 - M(1-a) \Rightarrow [\mathrm{q}(a,b) \wedge \mathrm{q}(1-a, 1-b)] = T$
2. $b > M(a)$ *or* $b < 1 - M(1-a) \Rightarrow [\mathrm{q}(a,b) \wedge \mathrm{q}(1-a, 1-b)] = F$
3. *M is concave*
4. *M is strictly increasing whenever $M(a) < 1$*
5. *M is continuous except possibly at* 0
6. $M(a) \geq a$

The proof of Theorem 2.1.3 can be found in Appendix A.

Theorem 2.1.4. (Sufficient Conditions). *If there exists a function M such that* q *satisfies the following conditions*

1. $M(a) \geq b \geq 1 - M(1-a) \Rightarrow \mathrm{q}(a,b) = T$
2. $b > M(a)$ *or* $b < 1 - M(1-a) \Rightarrow \mathrm{q}(a,b) = F$
3. *M is concave*
4. *M is strictly increasing whenever $M(a) < 1$*
5. *M is continuous except possibly at* 0
6. $M(a) \geq a$

then q-*Generic Differential privacy satisfies Axioms 2.1.1 and 2.1.2.*

The proof of Theorem 2.1.4 can be found in Appendix B.

The necessary and sufficient conditions given by Theorems 2.1.3 and 2.1.4 are almost identical.[3] The only difference appears in Item (1) of Theorem 2.1.3 and Item (1) of Theorem 2.1.4, where the ">" symbols are replaced with "≥" symbols. Thus the only difference is what happens when $M(a) = b$ or $1 - M(1-a) = b$. We can tighten the necessary conditions and relax the sufficient conditions with a more careful analysis (that considers points where M is not strictly concave) but this would complicate the statement of the theorems and so is omitted.

We illustrate Theorems 2.1.3 and 2.1.4 visually in Figure 1 (see also Figures 4, 5, 6, and 7 which are explained in Section 2.2). The meaning of this figure can be explained as follows. For a given privacy predicate q, suppose we are constructing a privacy mechanism $\mathfrak{M}$ for q-generic differential privacy. Once we have settled on a neighbor relation $\mathcal{R}$ and an output space $\mathbb{O}_{\mathfrak{M}}$ for $\mathfrak{M}$ then it is time to assign probabilities $P_{\mathfrak{M}}(O \mid i)$ for every $O \subseteq \mathbb{O}_{\mathfrak{M}}$ and every possible input $i \in \mathbb{I}$. If i_1 and i_2 are two datasets that

[3] Recall that we can replace $\mathrm{q}(a,b)$ with $\mathrm{q}'(a,b) \equiv \mathrm{q}(a,b) \wedge \mathrm{q}(1-a, 1-b)$ without changing the privacy definition; that is, q-generic differential privacy is exactly the same as q'-generic differential privacy.

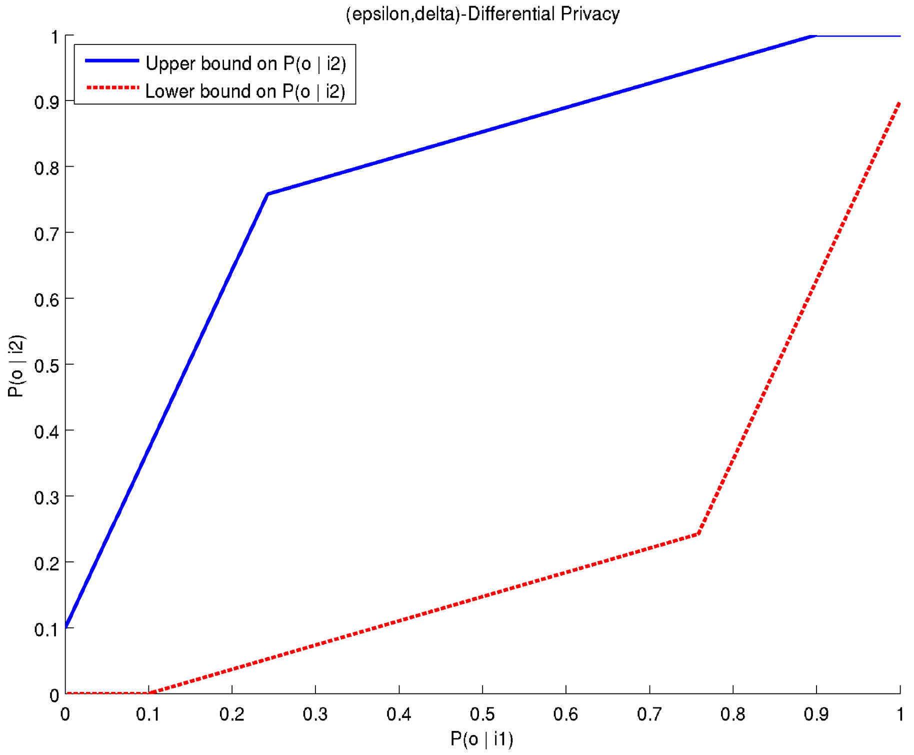

Figure 1: A plot of the functions $M(a)$ and $m(a) \equiv 1 - M(1-a)$ for (ϵ, δ)-differential privacy, with $\epsilon = 1$ and $\delta = 0.1$. For any fixed value of $P(O \mid i_1)$ on the x-axis, the allowable values of $P(O \mid i_2)$ lie between the curves $M(P(O \mid i_1))$ and $m(P(O \mid i_1))$. The curve defined by M is concave and lies above the line $y = x$.

are neighbors, then once we fix $P_{\mathfrak{M}}(O \mid i_1)$, the privacy predicate places restrictions on allowable values of $P_{\mathfrak{M}}(O \mid i_2)$. For example, i_1 may be a dataset in which Bob appears and has cancer and i_2 may be a dataset where Bob does not appear. The set O could be the statement "there are 42 cancer patients in the dataset". Thus, for example, if we decide that $\mathfrak{M}$ returns the statement "there are 42 cancer patients in the dataset" with probability 0.3 if the dataset happens to be i_1, then q places restrictions on the probability of making that statement if the dataset happens to be i_2. Intuitively, q should make this probability also close to 0.3 (so that the answer does not rely too much on Bob's presence in the dataset). If we accept Axioms 2.1.2 and 2.1.1 then q indeed does what our intuition suggests. It is forced to create an interval around 0.3 and the probability of answering "there are 42 cancer patients in the dataset" if the original data is i_2 must lie in this interval. Formally, once we fix $a \equiv P_{\mathfrak{M}}(O \mid i_1)$, the probability $b \equiv P_{\mathfrak{M}}(O \mid i_2)$ must lie between the interval with endpoints $m(a)$ and $M(a)$ (where $m(a) = 1 - M(1-a)$). As we vary a, the endpoints of this interval must vary in a nice way as well. If a increases, so do the endpoints. In fact, the curve specified by the upper endpoint, M, must be concave and the curve specified by the lower endpoint, m, must be convex. The corresponding curves for (ϵ, δ)-differential privacy [15, 38] are shown in Figure 1. The privacy predicate for (ϵ, δ)-differential privacy is $b \leq e^{\epsilon}a + \delta$. It is not hard to see that we get the same privacy definition by using the predicate $b \leq \min\{1,\ e^{\epsilon}a + \delta,\ 1 - e^{-\epsilon}(1 - a - \delta)\}$. In fact, for (ϵ, δ)-differential privacy, the upper bound is $M(a) = \min\{1,\ e^{\epsilon}a + \delta,\ 1 - e^{-\epsilon}(1 - a - \delta)\}$ and the lower bound is $m(a) = 1 - M(1-a)$. For ϵ-differential privacy, the upper bound function is $M(a) = \min(e^{\epsilon}a,\ 1 - e^{-\epsilon}(1-a))$, and the lower bound is $m(a) = 1 - M(1-a)$. Figure 1 plots these M and m curves for (ϵ, δ)-differential privacy. The x-axis is a value for $P_{\mathfrak{M}}(O \mid i_1)$ and the y-axis corresponds to a value for $P_{\mathfrak{M}}(O \mid i_2)$, which must lie between the M and m curves.

It is also interesting to note that the form of q required by Axioms 2.1.2 and 2.1.1 allows us to give a semantic interpretation to q-generic differential privacy. The formal statements are a bit complex (because of the generality of q-generic differential privacy) and can be found in [28, 29].

2.2 Applications

In this section we provide applications of Theorems 2.1.3 and 2.1.4. We show how Axioms 2.1.2 and 2.1.1 can be thought of as consistency requirements for a privacy definition and we give examples of accepted privacy definitions that do not meet these requirements. We then provide a general method for creating additional privacy predicates q for q-generic differential privacy that are consistent with our privacy axioms.

We begin by considering two privacy predicates. The first predicate is $\mathrm{q}_1(a, b) = T$ for all a, b (corresponding to $M(a) = 1$ and $m(a) = 0$) and the second is $\mathrm{q}_2(a, b) = T$ if and only if $b = \cos(a)$ (corresponding to cosine privacy, Definition 2.0.5). The first predicate, q_1, satisfies the necessary and sufficient conditions and so is consistent with our privacy axioms. This results in a lax privacy definition where the identity function (which simply outputs its input) satisfies q_1-generic differential privacy. Thus in some sense Axioms 2.1.2 and 2.1.1 are not very restrictive. However, they do rule out cosine privacy. To see why, suppose we have a privacy definition which allows us to release x, the number of people in New York and y, the number of people in New Jersey. We would expect this privacy definition to allow us to release $x + y$, the total number of people in New York and New Jersey instead of the individual counts x and y (this is the main idea behind Axiom 2.1.1). Indeed, it would be difficult to justify any privacy definition that allowed releasing x and y simultaneously while forbidding $x + y$ because such a privacy definition would not seem to be *internally consistent*. This is precisely the case with our straw-man cosine privacy.

Most privacy definitions that have appeared in the literature are not consistent with our privacy axioms (which further stresses the need for an axiomatic approach). One example of this is k-anonymity [43, 44] since it is very easy to take a k-anonymous table and post-process it so that each tuple has a distinct quasi-identifier. For example, one could replace each quasi-identifier attribute with a randomly generated number. This procedure would not add any privacy risks and, in fact, could only decrease the risk of a privacy breach, but the result would not be (explicitly) considered acceptable according to the definition of k-anonymity.

Most definitions that are consistent are variations of differential privacy. Here we cover some variations of differential privacy that are not consistent with the axioms. This includes probabilistic differential privacy [32] and (c, ϵ, δ)-privacy [9]; this fact was brought to our attention by Mironov (Ilya Mironov, personal communication, February 2010). Even though these privacy definitions are not consistent with our axioms, they do imply weaker definitions that are consistent variations of differential privacy (e.g., a mechanism satisfying probabilistic differential privacy [32] also satisfies (ϵ, δ)-differential privacy).[4]

These definitions follow a similar template: given two neighboring databases i_1, i_2 (under a suitable definition of neighbors), there is a set $O_{i_1,\text{bad}}$ of bad outputs for which no guarantees are provided and a set of good outputs $O_{i_1,\text{good}}$ for which a privacy predicate q holds; i.e., $\mathrm{q}(P_{\mathfrak{M}}(O \mid i_1), P_{\mathfrak{M}}(O \mid i_2)) = T$ for $O \subseteq O_{i_1,\text{good}}$ (usually this is the same predicate as in ϵ-differential privacy or (ϵ, δ)-differential privacy). These definitions require $P_{\mathfrak{M}}(O_{i_1,\text{bad}} \mid i_1) \leq \delta$. To show issues with consistency for these types of definitions, consider the following mechanism with two possible inputs (columns) that are neighbors and four possible outputs (rows):

	Input	
Output	0	1
A	0.89	0.79
B	0.1	0.2
C	0.01	0
D	0	0.01

[4] Actually, we should prefer the strongest *consistent* privacy definition implied by the inconsistent definition.

This mechanism satisfies $(\log 2)$-differential privacy with probability 0.99. The differential privacy constraints are not satisfied for outputs C and D, but they are produced with probability 0.01. Now consider a post-processing step which merges the outputs B and D into a single output labeled E. This results in the following mechanism:

	Input	
Output	0	1
A	0.89	0.79
C	0.01	0
E	0.1	0.21

Now $(\log 2)$-differential privacy is only satisfied with probability 0.79 because if the input is 1, there is a 0.21 probability of E being output, and the probabilities of generating E violate the constraint $P(\mathfrak{M}(0) = E) \leq 2P(\mathfrak{M}(1) = E)$ (which would be required of $(\log 2)$-differential privacy).

Thus requiring differential privacy constraints to hold with high probability is generally not a consistent approach[5] because a post-processing step that had no access to the input data suddenly results in a mechanism for which the privacy definition no longer holds. We conjecture that distributional privacy [6] may also have such an inconsistency.

2.2.1 Generating New Predicates

Now we provide a fairly general method for creating additional privacy predicates q for q-generic differential privacy that are consistent with our axioms.

Let $c(x)$ be a convex function such that $\kappa e^{-c(x)}$ is the density of a probability distribution (i.e., $\int_{-\infty}^{\infty} \kappa e^{-c(x)}\, dx = 1$). We allow $c(x)$ to equal ∞ for some values of x. Let $F(y)$ be the corresponding cumulative distribution function, and let $G(y) = 1 - F(y) = \int_y^\infty \kappa e^{-c(x)}\, dx$. For a fixed $t \geq 0$, we define $M(a) = G(G^{-1}(a) - t)$ and $m(a) = 1 - M(1-a)$. Note that G^{-1} exists on $(0, 1)$, by construction of G. We illustrate this definition in Figures 2 and 3.

We can think of a as the area under the right tail of the distribution $\kappa e^{-c(x)}$, so that $G^{-1}(a)$ is the left boundary of this region (Figure 2). We shift this boundary to the left by t units to place it at the point $G^{-1}(a) - t$, and then $G(G^{-1}(a) - t)$ is the area under this expanded tail region (Figure 3). We can perform a similar construction by using the left tail instead to get $M(a) = F(F^{-1}(a) + t)$, or we can combine the tails to get $M(a) = \min\left\{G(G^{-1}(a) - t),\ F(F^{-1}(a) + t)\right\}$.

The intuition behind choosing the tails of the distribution $\kappa e^{-c(x)}$ to define M is the following: If $c(x)$ is convex then the right tail has the following property. Out of all sets with probability a, when we shift the sets to the left (by subtracting t from each point), then the right tail experiences the largest gain in probability; thus we are considering the worst-case change in probability caused by a translation of length t to the left. If,

[5] We would like to thank Ilya Mironov for making this observation.

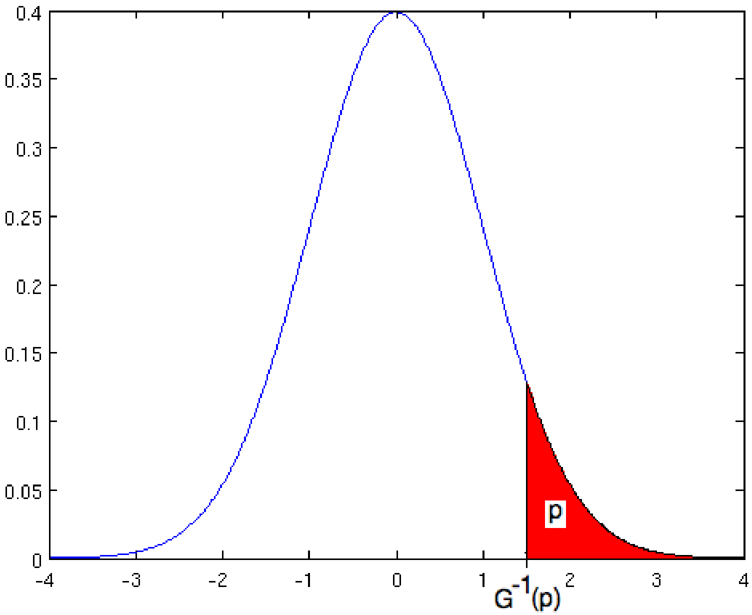

Figure 2: The red shaded region has area (probability) p and $G^{-1}(p)$ is the left boundary of this region.

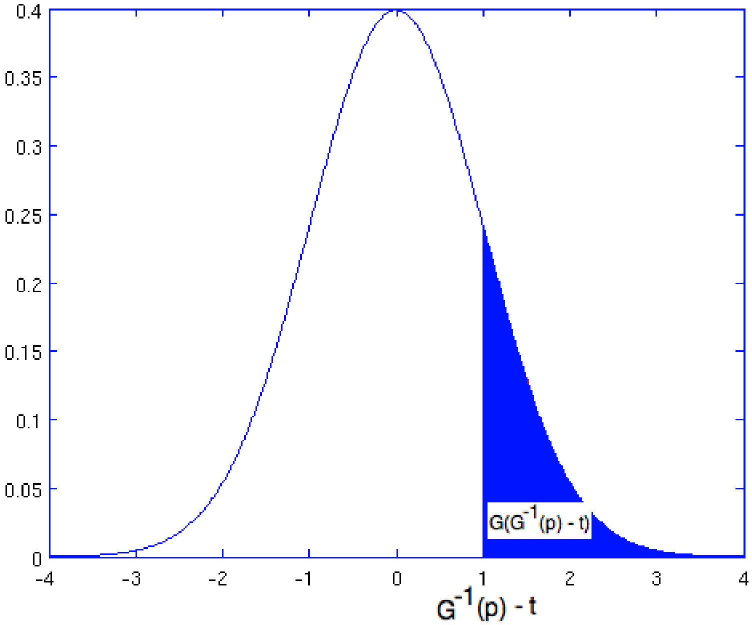

Figure 3: The left boundary from Figure 2 has been shifted to the left by t units and is now at the point $G^{-1}(p) - t$. The blue shaded area under the curve and to the right of this boundary has area (probability) $G(G^{-1}(p) - t)$.

instead, we use $M(a) = F(F^{-1}(a) + t)$ then we are considering the worst-case caused by translation to the right, and if we use $M(a) = \min\left\{G(G^{-1}(a) - t),\ F(F^{-1}(a) + t)\right\}$ then we are considering the worst-case caused by translation in either direction. This argument is formalized by the following theorem.

Theorem 2.2.1. *For any set $S \subset \mathbb{R}$, let S_{-t} denote the set $\{s - t \ :\ s \in S\}$. Let X be a random variable with density $\kappa e^{-c(x)}$, where $c(x)$ is convex. Consider the optimization problem*

$$\arg\max_{S \subseteq \mathbb{R}} P(X \in S_{-t}) - P(X \in S)$$

subject to $P(X \in S) = p$. This quantity is maximized by the set $\{r \ :\ r \geq G^{-1}(p)\}$.

A similar theorem holds for the left tail. The proof can be found in Appendix C.

The following theorem shows that this way of constructing M satisfies the sufficient conditions in Theorem 2.1.4 and hence the privacy axioms as well.

Theorem 2.2.2. *Given a random variable X with density $\kappa e^{-c(x)}$, where c is a convex function, let $G(x) = P(X \geq x)$. For a fixed $t > 0$, let $M(a) = G(G^{-1}(a) - t)$. Then M satisfies the following properties:*

- *M is concave*
- *M is strictly increasing whenever $M(a) < 1$*
- *M is continuous except possibly at 0*
- *$M(a) \geq a$*

The proof can be found in Appendix C.

We can use the technique justified by Theorem 2.2.2 to generate additional privacy predicates q for which q-generic differential privacy is consistent with our privacy axioms. If we set $c(x) = \frac{1}{2}x^2$, the corresponding distribution is a standard Gaussian. Setting $M(a) = G(G^{-1}(a) - 1)$ and $m(a) = 1 - M(1 - a)$ we get the predicate $\mathrm{q}(a, b) \equiv m(a) \leq b \leq M(a)$. The upper bound M and lower bound m functions are shown in Figure 4.

If, instead, we set $c(x) = |x|$, we get the Laplace distribution. The corresponding M and m functions are shown in Figure 2.2.1. This does not give the same M and m as in differential privacy. Differential privacy results from the two-step process we describe next.

If we set $c(x) = x$ if $x \geq 0$ and ∞ if $x < 0$, we get the exponential distribution. The corresponding M and m functions are shown in Figure 6.

We may also want to symmetrize the privacy predicate q. That is, we may want to have the condition $\mathrm{q}(a, b) \equiv m(a) \leq b \leq M(a) \wedge m(b) \leq a \leq M(b)$. This can be converted to a condition $\mathrm{q}(a, b) \equiv m'(a) \leq b \leq M'(a)$ with the observation that

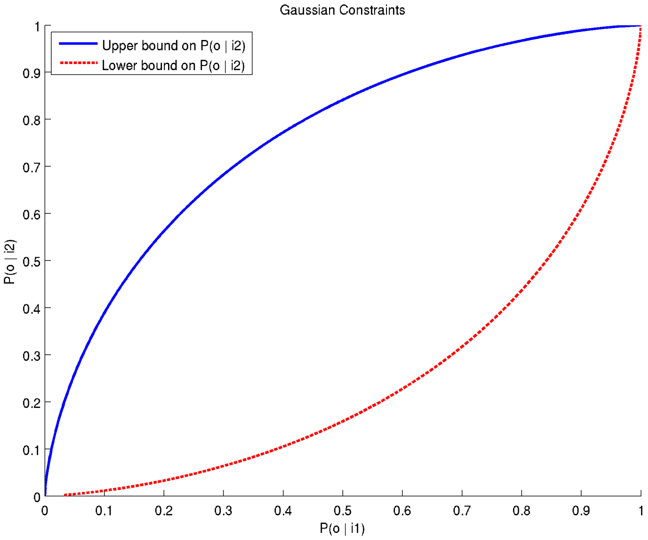

Figure 4: Upper and lower bound functions derived from a Gaussian distribution.

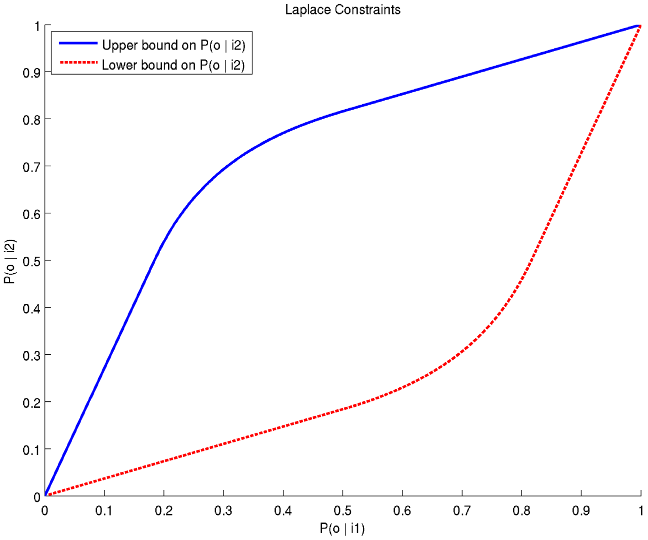

Figure 5: Upper and lower bound functions derived from a Laplace distribution.

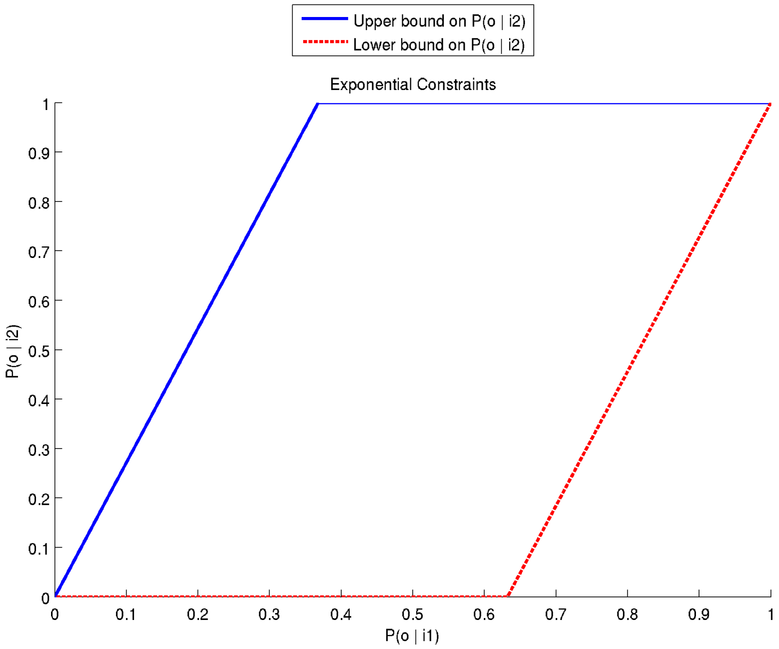

Figure 6: Upper and lower bound functions derived from the Exponential distribution.

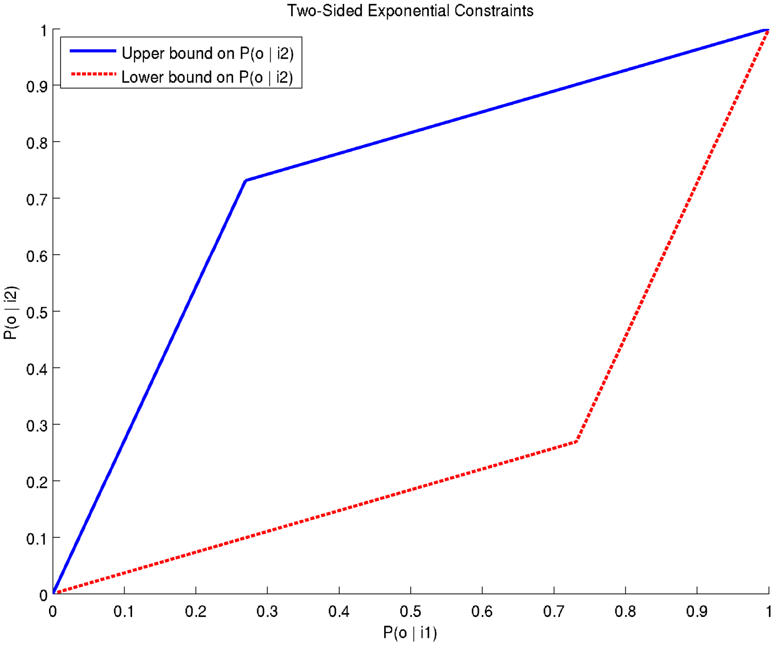

Figure 7: Symmetrized upper and lower bound functions derived from the Exponential distribution.

$a \leq M(b)$ if and only if $M^{-1}(a) \leq b$.[6] Thus we can then replace $\mathrm{q}(a,b) \equiv m(a) \leq b \leq M(a) \wedge m(b) \leq a \leq M(b)$ by the equivalent condition $\mathrm{q}(a,b) \equiv \max\{m(a), M^{-1}(a)\} \leq b \leq \min\{M(a), m^{-1}(a)\}$. When we symmetrize the privacy predicate derived from the exponential distribution, we get the upper and lower bound functions shown in Figure 7, which corresponds exactly to the upper and lower bound functions associated with differential privacy (Definition 2.0.3).

3 Reasoning About Utility

To really take advantage of privacy definitions (both new and old), we need to design privacy mechanisms that output the most useful data possible. For example, any mechanism whose output is independent of the input satisfies generic differential privacy. However, this is not a pleasing result since it seems that we can do "better". A common approach for "doing better" is to select a utility measure and to design a privacy mechanism that maximizes this utility measure (while preserving privacy guarantees). Utility measures are typically chosen arbitrarily or with the justification that they are used in decision-theoretic statistics.

Although intuitively this seems like a valid approach, recent results indicate otherwise. Ghosh et al. [24] have shown, in the case of differential privacy, that if a user asks a single count query, believes in a prior distribution over query answers, and provides a loss function from a suitably well-behaved class, then the following is true: There exists a privacy mechanism, called the *geometric mechanism* [24], such that any optimal mechanism (in the sense of minimizing expected loss) can be constructed from the geometric mechanism by a lossy postprocessing step (in general, the geometric mechanism is not considered optimal by the expected utility metric). This postprocessing step is a deterministic function that is not one-to-one and thus removes information. In a followup paper, Gupte and Sundararajan [25] show a similar result for minimax error.

Here we present a more concrete example of the paradoxical behavior of some utility measures. Suppose we are conducting a survey that asks the following question: "Have you attempted to commit a crime within the last 6 months?" Since this is a sensitive question, we may be concerned that respondents will not answer truthfully. We can adopt a variation of randomized response [45] to design the following mechanism. We instruct the respondent to answer truthfully with a certain probability and to lie otherwise. Thus for any particular respondent, we would not be sure of the respondent's true answer.

Let us suppose that we have decided upon a privacy definition which is satisfied by the mechanisms in Figures 8(a) and 8(b). In mechanism $\mathfrak{M}_1$ (Figure 8(a)), the respondent is instructed to lie with probability 1/3. In mechanism $\mathfrak{M}_2$ (Figure 8(b)), the user is instructed to always answer "yes".

[6] We can define M^{-1} because our axioms and Theorem 2.1.3 force M to be strictly increasing when $M(a) < 1$ and so M has a well-defined inverse except at 1; however, we can define $M^{-1}(1)$ as $\inf\{x : M(x) = 1\}$ except for the pathological function M where $M(0) \neq 1$ and $M(a) = 1$ for $a \neq 0$ since M must be continuous everywhere except possibly at 0.

	True Answer	
	Yes	No
Yes	2/3	1/3
No	1/3	2/3

(a) Privacy Mechanism $\mathfrak{M}_1$

	True Answer	
	Yes	No
Yes	1	1
No	0	0

(b) Privacy Mechanism $\mathfrak{M}_2$

Figure 8: Two privacy mechanisms for a survey.

We may believe that 75% of the population did indeed attempt to commit a crime (this serves as our prior for expected utility). We can define the loss to be 1 if the respondent lies and 0 otherwise. Now we want to choose the mechanism, $\mathfrak{M}_1$ (Figure 8(a)) or $\mathfrak{M}_2$ (Figure 8(b)), which has the smallest expected loss. The expected loss is $P(\text{Yes})P(\text{Response=No} \mid \text{Yes}) + P(\text{No})P(\text{Response=Yes} \mid \text{No})$. Thus the expected loss of $\mathfrak{M}_1$ is 1/3 and the expected loss of $\mathfrak{M}_2$ is 1/4.

Since $\mathfrak{M}_2$ has lower expected loss (and so higher expected utility), we would prefer $\mathfrak{M}_2$ according to this metric. However, we can simulate $\mathfrak{M}_2$ by first using $\mathfrak{M}_1$ and then mapping all responses to "Yes". In other words, there exists an algorithm $\mathcal{A}$ such that $\mathfrak{M}_2 = \mathcal{A} \circ \mathfrak{M}_1$. Since we can simulate $\mathfrak{M}_2$ from the output of $\mathfrak{M}_1$ (and not vice versa) it stands to reason that $\mathfrak{M}_1$ provides strictly more information than $\mathfrak{M}_2$ regardless of its expected loss.

Thus, expected utility seems like a poor choice of utility metric when choosing a privacy mechanism. In addition, optimizing a privacy mechanism $\mathfrak{M}$ for one specific task may also be a mistake—there could exist a privacy mechanism $\mathfrak{M}'$ such that $\mathfrak{M}(i) = \mathcal{A}(\mathfrak{M}'(i))$ for some randomized algorithm $\mathcal{A}$. Thus choosing $\mathfrak{M}'$ instead of the highly tuned $\mathfrak{M}$ would be preferable because $\mathfrak{M}'$ is clearly just as useful for the original task, but may also be useful for other tasks as well. In this sense $\mathfrak{M}'$ is sufficient for any task that requires $\mathfrak{M}$.

We formalize this notion of sufficiency with the sufficiency axiom in Section 3.1. We then present several measures of utility (one of which can be justified with additional axioms) and discuss whether or not they are appropriate for use in statistical privacy (Section 3.2). Finally, we characterize what optimal differentially private mechanisms should look like for finite input and output spaces.

3.1 The Sufficiency Axiom for Utility

Recall that a privacy mechanism $\mathfrak{M}$ is a randomized (or deterministic) algorithm with input space $\mathbb{I}_{\mathfrak{M}}$ and output space $\mathbb{O}_{\mathfrak{M}}$ and which satisfies some privacy definition. We represent $\mathfrak{M}$ as a conditional probability distribution $P_{\mathfrak{M}}(o \mid i)$ just as with any randomized algorithm. For any randomized algorithm $\mathcal{A}$, $\mathcal{A} \circ \mathfrak{M}$ denotes the composition of $\mathcal{A}$ and $\mathfrak{M}$, and is defined by first running $\mathfrak{M}$ and then running $\mathcal{A}$ on the output of

$$\begin{array}{cc}
 & \text{Inputs:} \\
 & \begin{array}{cccc} i_1 & i_2 & \dots & i_n \end{array} \\
\begin{array}{c} \text{Output } o_1 \\ \text{Output } o_2 \\ \vdots \\ \text{Output } o_m \end{array} & \begin{pmatrix} P(o_1 \mid i_1) & P(o_1 \mid i_2) & \dots & P(o_1 \mid i_n) \\ P(o_2 \mid i_1) & P(o_2 \mid i_2) & \dots & P(o_2 \mid i_n) \\ \vdots & \vdots & \vdots & \vdots \\ P(o_m \mid i_1) & P(o_m \mid i_2) & \dots & P(o_m \mid i_n) \end{pmatrix}
\end{array}$$

Figure 9: Matrix representation of a randomized algorithm $\mathcal{A}$ or mechanism $\mathfrak{M}$ with finite input and output spaces.

$\mathfrak{M}$.

When the input and output spaces are finite we can treat $\mathfrak{M}$ as a column stochastic matrix[7] $\{m_{j,k}\}$ whose (j,k) entry $m_{j,k}$ is equal to $P_{\mathfrak{M}}(j|k)$. Thus the rows correspond to elements of the output space and columns correspond to elements of the input space. We will abuse notation and use the symbol $\mathfrak{M}$ to refer to the matrix as well. In matrix form, the composition $\mathcal{A} \circ \mathfrak{M}$ is equivalent to $\mathcal{A}\,\mathfrak{M}$ (interpreted as matrix multiplication) when $\mathcal{A}$ and $\mathfrak{M}$ have randomness independent of each other.

Convention 3.1.1. (Matrix form of $\mathcal{A}$) *Given a randomized algorithm $\mathcal{A}$ with finite input and output space, we represent $\mathcal{A}$ as a matrix $\{m_{i,j}\}$ such that $m_{i,j} = P_{\mathcal{A}}(i \mid j)$. An example is shown in Figure 9.*

Mechanisms can be partially ordered by the sufficiency partial order, defined as follows:

Definition 3.1.2. (Sufficiency Partial Order). *Let S be the set of privacy mechanisms that satisfy a particular privacy definition. If $\mathfrak{M}_1 \in S$ and $\mathfrak{M}_2 \in S$ then we say that $\mathfrak{M}_2$ is sufficient for $\mathfrak{M}_1$, and denote this by $\mathfrak{M}_1 \preceq_{\mathrm{S}} \mathfrak{M}_2$, if there exists a randomized algorithm $\mathcal{A}$ such $\mathfrak{M}_1 = \mathcal{A} \circ \mathfrak{M}_2$ (or, more formally, $P_{\mathfrak{M}_1} = P_{\mathcal{A} \circ \mathfrak{M}_2}$). We call this partial order the* sufficiency partial order.

Thus if one can probabilistically simulate $\mathfrak{M}_1$ by postprocessing the output of $\mathfrak{M}_2$ with some randomized algorithm $\mathcal{A}$, then $\mathfrak{M}_2$ is sufficient for any task that requires $\mathfrak{M}_1$, and so $\mathfrak{M}_2$ is at least as preferable as $\mathfrak{M}_1$. This notion of sufficiency is equivalent to Bayesian sufficiency in comparing experiments and forecasters (see [12], Equation 3.5). It also leads to the following definition:

Definition 3.1.3. (Maximally Sufficient Mechanism). *Let S be the set of privacy mechanisms that satisfy a particular privacy definition. A privacy mechanism $\mathfrak{M} \in S$ is* maximally sufficient *if for every privacy-mechanism $\mathfrak{M}' \in S$ such that $\mathfrak{M} \preceq_{\mathrm{S}} \mathfrak{M}'$ it is also true that $\mathfrak{M}' \preceq_{\mathrm{S}} \mathfrak{M}$.*

[7] A matrix with nonnegative entries where each column sums up to 1.

Thus one cannot simulate a maximally sufficient mechanism with a non-invertible (i.e., lossy) post-processing of another privacy mechanism. Thus, for a given privacy definition, the corresponding set of maximally sufficient mechanisms is clearly desirable.

We first note that privacy axioms Axioms 2.1.1 and 2.1.2 are not strong enough to guarantee that a privacy definition must have maximally sufficient mechanisms. For example, if the privacy predicate in differential privacy (Definition 2.0.3) is replaced by $P_{\mathfrak{M}}(O \mid i_1) < e^{\epsilon} P_{\mathfrak{M}}(O \mid i_2)$ (using the strict inequality $<$ instead of $\leq$), then maximally sufficient mechanisms do not exist for the same reason that there is no largest number in the semi-open interval $[0, 1)$. Thus it would be interesting to see what natural privacy axioms are needed to enforce this condition.

Aside from existence of a nonempty set of maximally sufficient mechanisms, another property that is desirable but not guaranteed is a coverage condition: for every $\mathfrak{M}$ there exists a maximally sufficient $\mathfrak{M}^*$ such that $\mathfrak{M} \preceq_{\mathrm{S}} \mathfrak{M}^*$ (that is, every privacy mechanism can be realized as the postprocessing of the output of a maximally general mechanism). As with existence, it is important to find natural privacy axioms that can force the set of maximally sufficient mechanisms to have this coverage property.

Now we present our axiom of sufficiency for utility.

Axiom 3.1.4. (Axiom of sufficiency). *A measure μ of the utility of a privacy mechanism must respect the sufficiency partial order $\preceq_{\mathrm{S}}$. That is, $\mu(\mathfrak{M}) \geq \mu(\mathcal{A} \circ \mathfrak{M})$ for any randomized algorithm whose input space is the output space of $\mathfrak{M}$ (where $\mathcal{A}$ has no access to the input data and its randomness is independent from the randomness in $\mathfrak{M}$).*

Thus, if the set of maximally sufficient mechanisms had the desired coverage property, every valid measure of utility would be maximized by a maximally sufficient mechanism. We feel that this axiom is universal for any application. It eliminates measures, such as expected utility, that can generate the utility paradox discussed at the beginning of Section 3.

Note that this utility axiom, like the privacy axioms, is just a consistency condition for utility measures. Furthermore, it is information-theoretic and rules out computational notions of utility and privacy—in many cases it would consider an encrypted dataset to have more utility than a statistical model built from the data (since given a suitable encryption scheme and enough time the original database could be reconstructed).

3.2 Measures of Utility

In this section we examine some candidate measures of utility using the Axiom of Sufficiency (Axiom 3.1.4). For simplicity, we will assume that the input and output spaces are finite and thus we treat privacy mechanisms and randomized algorithms as column stochastic matrices, as discussed in Convention 3.1.1 and Figure 9. Thus any valid (according to the Axiom of Sufficiency) utility measure μ needs to satisfy the following property: $\mu(\mathcal{A}\,\mathfrak{M}) \leq \mu(\mathfrak{M})$ (where $\mathfrak{M}$ is a privacy mechanism and $\mathcal{A}$ is a randomized algorithm).

Example 3.2.1. (Negative Expected Loss). *Let L be a loss matrix where $L(j,k)$ is the loss we incur for outputting j when k is the true input. If $\mathfrak{M}$ is a privacy mechanism, we may want to minimize its expected loss, which is equivalent to maximizing $-Trace(L^T\,\mathfrak{M})$. The results of Ghosh et al. [24] imply that negative expected utility does not satisfy Axiom 3.1.4.*

Example 3.2.2. (Absolute value of Determinant). *If $\mathfrak{M} = \{m_{i,j}\}$ is represented as a square column stochastic matrix (see Convention 3.1.1) then it is natural to consider the utility measure $\mu(\mathfrak{M}) = |det(\mathfrak{M})|$. By the multiplicative property of the determinant, $|det(\mathcal{A}\,\mathfrak{M})| = |det(\mathcal{A})|\ |\det(\mathfrak{M})| \leq |\det(\mathfrak{M})|$ for a randomized algorithm $\mathcal{A}$ that is represented by a square matrix, since column stochastic matrices have determinants with absolute value ≤ 1. Geometrically, this measures the contractive properties of $\mathfrak{M}$ because $\mathfrak{M}$ maps the unit hypercube into another convex polytope whose area is $|det(\mathfrak{M})|$ [42]. This μ satisfies our utility criterion with the proviso that we are only considering privacy mechanisms whose output space is the same size as the input space. This is a restrictive assumption since we show in Section 3.3 that for differential privacy there are many maximally sufficient privacy mechanisms with much larger output spaces than input spaces (another weakness is that any mechanism with linearly dependent rows would have the smallest possible utility: 0).*

The absolute value of the determinant can be extended to non-square matrices by considering the area of the polytope determined by the columns of $\mathfrak{M}$. This area is $\sqrt{|\det(\mathfrak{M}^T\,\mathfrak{M})|}$ and is non-zero if the columns are linearly independent.[8] *This is equal to the absolute value of the determinant if $\mathfrak{M}$ is square. Unfortunately, this extension does not satisfy the axiom of sufficiency. One counterexample is:*

$$\mathfrak{M}_2 = \begin{bmatrix} 0 & 0.8 \\ 1 & 0.2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0.8 \\ 0.7 & 0 \\ 0.3 & 0.2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \mathfrak{M}_1$$

since $\sqrt{|\det(\mathfrak{M}_2^T\,\mathfrak{M}_2)|} = 0.8 > 0.6251 \approx \sqrt{|\det(\mathfrak{M}_1^T\,\mathfrak{M}_1)|}$.

Example 3.2.3. (Negative Dobrushin's Coefficient). *For a privacy mechanism $\mathfrak{M} = \{m_{i,j}\}$, define $\mu_{Dob}(\mathfrak{M}) = -\min_{j,k} \sum\limits_i \min(m_{i,j}, m_{i,k})$. This is the negative of Dobrushin's coefficient of ergodicity and is another useful measure of the contractive properties (in the geometric sense) of a column stochastic matrix [11]. In fact, $1 + \mu_{Dob}(\mathfrak{M})$ is equal to $\sup ||\,\mathfrak{M}\,x - \mathfrak{M}\,y||_1/||x-y||_1$ over all distinct pairs of vectors x, y with positive entries and unit L_1 norm (i.e., $||x||_1 = ||y||_1 = 1$) [11]. Thus it measures how much $\mathfrak{M}$ contracts the L_1 distance between vectors. In Appendix D we prove that $\mu_{Dob}(\mathcal{A}\,\mathfrak{M}) \leq \mu_{Dob}(\mathfrak{M})$. Optimal differentially private mechanisms under this metric are derived in [28, 29].*

Note that if $\sum\limits_i \min(m_{i,j}, m_{i,k})$ is close to 1, then the columns corresponding to inputs j and k are very similar (e.g., close in the L_1 norm), so that it is difficult to distinguish

[8] Without loss of generality, we assume that all rows consisting entirely of 0 entries are removed.

between them. If $\sum_i \min(m_{i,j}, m_{i,k})$ *is close to* 0 *then it is easy to distinguish between inputs* j *and* k*: if an output* o *has high probability for input* j *then it has low probability for input* k*. Thus the negative Dobrushin coefficient ensures that there exist two inputs that are easy to distinguish.*

Example 3.2.4. (Discrimination between all pairs of inputs). *Another possibility is to modify Example 3.2.3 to base a utility measure on the two inputs of a mechanism that are most difficult to distinguish between. For example, define* $\mu_{disc} = -\max \sum_i \min(m_{i,j}, m_{i,k})$*. Then the discussion in Example 3.2.3 shows that the utility measure* μ_{disc} *prefers mechanisms for which it is easy to distinguish between every pair of inputs. The proof that* $\mu_{disc}(\mathcal{A}\,\mathfrak{M}) \leq \mu_{disc}(\mathfrak{M})$ *is similar to the corresponding proof for* μ_{Dob}*. The details are discussed in Appendix D.*

While the Axiom of Sufficiency (Axiom 3.1.4) is universal in the sense that any utility metric used for statistical privacy must be consistent with it, other axioms would be chosen based on the needs of an application. Here we present two additional axioms. The first axiom, Axiom 3.2.5 requires continuity in our utility function. The second axiom, Axiom 3.2.6 states that a postprocessing algorithm that maps outputs o_1 and o_2 to the same point reduces the utility by an amount that only depends on $P(o_1 \mid \cdot)$ and $P(o_2 \mid \cdot)$. Both of these axioms have been well-studied in the literature on functional equations and information theory [20].

Axiom 3.2.5. (*Continuity*). *μ is continuous in the components of $\mathfrak{M}$ (when viewed as a matrix).*

Continuity is justified by the idea that a small change in the probabilities governing a mechanism should result in a small change in the usefulness of the information it provides—if we change one of the probabilities slightly, the new mechanism will, with high probability, behave exactly like the old mechanism.

Axiom 3.2.6. (Branching). *Given a finite input space* $\mathbb{I}$*, there exists a function* G *such that for any mechanism* $\mathfrak{M}$ *with input space* $\mathbb{I}$ *and finite output space* $\mathbb{O}_{\mathfrak{M}} = \{o_1, \ldots, o_n\}$,

$$\begin{aligned}&\mu\Big(P_{\mathfrak{M}}(o_1 \mid \cdot), \ldots, P_{\mathfrak{M}}(o_n \mid \cdot)\Big)\\ &\quad = \quad \mu\Big(P_{\mathfrak{M}}(o_1 \mid \cdot) + P_{\mathfrak{M}}(o_2 \mid \cdot), P_{\mathfrak{M}}(o_3 \mid \cdot), \ldots, P_{\mathfrak{M}}(o_n \mid \cdot)\Big) + G\Big(P_{\mathfrak{M}}(o_1 \mid \cdot), P_{\mathfrak{M}}(o_2 \mid \cdot)\Big).\end{aligned}$$

Note that the branching axiom relates the utility of a mechanism with n rows (when represented as a matrix) to the utility of a mechanism with $n-1$ rows. It deals with the idea that we lose information if we are not able to distinguish between two outputs. For example, consider a post-processing algorithm $\mathcal{A}$ that outputs its input with the exception that it outputs the statement "o_1 or o_2" whenever the input of $\mathcal{A}$ is o_1 or o_2. The combined mechanism $\mathcal{A} \circ \mathfrak{M}$ should have lower utility: clearly knowing that the output was either o_1 or o_2 is not as useful as knowing precisely which was the true output of a mechanism $\mathfrak{M}$. Thus merging outputs (i.e., reporting "o_1 or o_2" whenever the output is o_1 or o_2) reduces the amount of information. The branching axiom contains two distinct ideas. The first is that information loss from merging two outputs o_1, o_2 only depends on $P_{\mathfrak{M}}(o_1 \mid \cdot)$ and $P_{\mathfrak{M}}(o_2 \mid \cdot)$ and that none of the other outputs factor into the information loss (i.e., information loss should not depend on what didn't happen in either situation). The second idea, is that the loss of information is additive rather than, say, multiplicative. The idea that information should be additive seems rather arbitrary but it is just a matter of rescaling: had we initially chosen a multiplicative information decrease, for instance, we can convert it to an additive information decrease simply by taking logarithms.

The two axioms of Branching and Continuity, combined with the Axiom of Sufficiency (Axiom 3.1.4), completely characterize the following measure of utility.

Definition 3.2.7. (Branching Measures). *A utility measure μ is a* branching measure *if $\mu(\vec{x}_1, \dots \vec{x}_n) = \sum\limits_{i=1}^{n} F(\vec{x}_i)$ for some continuous convex function F whose domain is the set of vectors with nonnegative components and such that $F(c\vec{y}) = cF(\vec{y})$ for every $c \geq 0$.*

Thus if a mechanism $\mathfrak{M}$ has finite output space $\mathbb{O}_{\mathfrak{M}} = \{o_1, \dots, o_n\}$, a branching measure μ assigns $\mathfrak{M}$ the utility score:

$$\mu(\mathfrak{M}) \equiv \mu\Big(P_{\mathfrak{M}}(o_1 \mid \cdot), \dots, P_{\mathfrak{M}}(o_n \mid \cdot)\Big) = \sum_{j=1}^{n} F(P_{\mathfrak{M}}(o_j \mid \cdot)).$$

The following theorem links the branching, continuity, and sufficiency axioms to the definition of branching measures.

Theorem 3.2.8. *A utility metric μ is a branching measure if and only if it satisfies Axioms 3.1.4, 3.2.5, and 3.2.6.*

The proof can be found in Appendix F.

3.3 Maximally Sufficient Differentially Private Mechanisms

In this section we characterize maximally sufficient differentially private algorithms. Our main result is Theorem 3.3.2, which characterizes what maximally sufficient mechanisms with finite input and output spaces look like (an extension to infinite output spaces is discussed in Appendix E; an extension to infinite input spaces requires additional topological and measure-theoretic assumptions and is left for future work).

Recall that differential privacy has an input space $\mathbb{I}$, a symmetric neighbor relation $\mathcal{R}$, and the requirement that a mechanism $\mathfrak{M}$ (with output space $\mathbb{O}_{\mathfrak{M}}$) must satisfy the condition that for any (measurable) $O \subseteq \mathbb{O}_{\mathfrak{M}}$ and $(i_1, i_2) \in \mathcal{R}$ we must have $P_{\mathfrak{M}}(O \mid i_1) \leq e^{\epsilon} P_{\mathfrak{M}}(O \mid i_2)$ and $P_{\mathfrak{M}}(O \mid i_2) \leq e^{\epsilon} P_{\mathfrak{M}}(O \mid i_1)$.

If the output and input spaces are finite, then for each output $o \in \mathbb{O}_{\mathfrak{M}}$ we can create a graph whose nodes are the inputs and whose edges are determined by pairs of

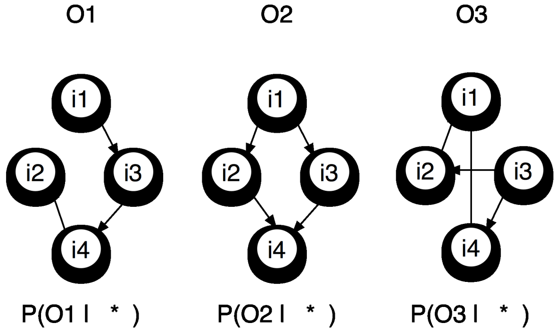

Figure 10: A set of row graphs for a differentially private mechanism $\mathfrak{M}$. From the graph we can see that $P_{\mathfrak{M}}(o_1 \mid i_1) = e^\epsilon P_{\mathfrak{M}}(o_1 \mid i_3)$, $P_{\mathfrak{M}}(o_1 \mid i_3) = e^\epsilon P_{\mathfrak{M}}(o_1 \mid i_4)$, $P_{\mathfrak{M}}(o_1 \mid i_4) = e^\epsilon P_{\mathfrak{M}}(o_1 \mid i_2)$, etc.

neighbors for which the inequality constraints hold with equality. This is formalized in Definition 3.3.1.

Definition 3.3.1. (Row graphs). *For a differentially private mechanism $\mathfrak{M}$ with finite output space and input space, the* row graphs *of $\mathfrak{M}$ are a set of graphs, one for each $o \in \mathbb{O}_{\mathfrak{M}}$. The graph associated with output o has $\mathbb{I}$ as the set of nodes, and for any $i_1, i_2 \in \mathbb{I}$, there is a directed edge (i_i, i_2), if $(i_1, i_2) \in \mathcal{R}$ and $P_{\mathfrak{M}}(O \mid i_1) = e^\epsilon P_{\mathfrak{M}}(O \mid i_2)$.*

An example of a set of row graphs is shown in Figure 10. These are row graphs from a mechanism with 4 possible inputs and (at least) 3 possible outputs. From these row graphs, we can see that $P_{\mathfrak{M}}(o_1 \mid i_1) = e^\epsilon P_{\mathfrak{M}}(o_1 \mid i_3)$, $P_{\mathfrak{M}}(o_1 \mid i_3) = e^\epsilon P_{\mathfrak{M}}(o_1 \mid i_4)$, $P_{\mathfrak{M}}(o_1 \mid i_4) = e^\epsilon P_{\mathfrak{M}}(o_1 \mid i_2)$. Notice that for each output o, the corresponding row graph is connected (when viewed as an undirected graph). This is not a requirement for differentially private mechanisms, but Theorem 3.3.2 shows that the axiom of sufficiency forces this to hold for maximally sufficient differentially private mechanisms (this is another case where an intuitive idea can be justified by axioms).

Theorem 3.3.2. *For a given $\epsilon > 0$, finite input space $\mathbb{I}$, and symmetric neighbor relation $\mathcal{R}$ (it must be symmetric for differential privacy), let S be the set of all ϵ-differentially private mechanisms (with input space $\mathbb{I}$). Let S_{con} be the subset of S consisting of all mechanisms $\mathfrak{M}$ with finite output spaces and such that each row graph is connected (when viewed as an undirected graph). Then S_{con} is precisely the set of maximally sufficient differentially private mechanisms with finite output spaces.*

The proof can be found in Appendix E, which also discusses extensions to infinite output spaces.

One consequence of Theorem 3.3.2 is that we can identify each output with its row graph. For any two distinct outputs o_1 and o_2 with the same row graph, it is easy to see that $f_1(\cdot) \equiv P_{\mathfrak{M}}(o_1 \mid \cdot)$ and $f_2(\cdot) \equiv P_{\mathfrak{M}}(o_2 \mid \cdot)$ are proportional to each other and so o_1 and o_2 can be merged into a single point without any loss of information. Thus we can now define a common output space for all differentially private mechanisms with finite input space[9] to be the set of all possible row graphs on the inputs.

This means that the outputs of a differentially private mechanism are not best viewed as query answers or synthetic data—they are best viewed as row graphs, which correspond to a restricted set of likelihood functions $P(o_j \mid \cdot)$. To statisticians this is a pleasing result, since according to the *likelihood principle* [8], the likelihood function is all we need for statistical inference about the inputs from the outputs.

On the other hand, this result is less pleasing to end-users, who use statistical software whose input is data, not likelihood functions. Thus, in addition to maximally sufficient mechanisms, we need to develop additional tools to shoe-horn this output space into a format that can be digested by off-the-shelf statistical software.

One final point to mention is that previous work on differential privacy has focused on mechanisms with output spaces that were at most the size of the input space or were equivalent (according to the sufficiency partial order $\preceq_S$) to such mechanisms. Since the number of possible row graphs is larger than the cardinality of the input space, many maximally sufficient mechanisms could have been missed this way. Thus the existence of parts of Theorem 3.3.2 are obvious after the fact, but surprisingly not a priori.

4 Related Work

Our efforts at axiomatizing privacy and utility are motivated by corresponding efforts in mathematical philosophy and probabilistic inductive logic (e.g., [7, 39]) where the goal is to model the reasoning of a rational agent.

For surveys on statistical privacy, see [10, 22, 1].

The need for a better understanding of privacy definitions and privacy mechanisms was underscored by the work of Dinur and Nissim [13] (and later by Dwork et al. [17]) that showed that some intuitive methods for preserving privacy actually did not preserve privacy according to essentially any privacy definition. This work was followed by a line of research that led to differential privacy [5, 14, 16, 15, 38, 19]. Note that there have been some attempts to weaken the definition of differential privacy (e.g., [15, 38, 32, 31]) as its stringent guarantees are not always considered necessary (especially when data utility can be increased).

What sets differential privacy apart from most privacy definitions is the strength of its guarantees and the ability to formally investigate its properties. In particular, Rastogi et al. provide a connection between privacy and utility guarantees [41] as well as a connection to another definition known as adversarial privacy [40], which was also studied by Evfimievski et al. [21] in the context of query auditing.

Utility of sanitized data has also been studied. Of particular relevance are the following: McSherry and Talwar [33] have presented a general recipe for taking a "quality

[9] The restriction to finite input spaces is fundamental, but the requirement of a finite output space can be eliminated, see Appendix E.

function" and turning it into a privacy mechanism for differential privacy. Although this recipe does not come with optimality guarantees in terms of the quality function, it has been used successfully in other work [6]. Dwork et al. [18] provided a link between utility and computational complexity for differential privacy. Recent work by Ghosh et al. [24] and Gupte and Sundararajan [25] has shown that optimizing for commonly accepted utility metrics (expected utility and minimax error) is not always the correct goal since the output of a suboptimal mechanism (according to the utility metric) may sometimes be post-processed (in a lossy way) to mimic the output of an "optimal" mechanism.

5 Conclusions

In this paper we presented five axioms for privacy and utility and showed how they can guide the development of privacy definitions, utility measures, and privacy mechanisms. We feel this is just the beginning of a unified theory of privacy. Additional consequences of the Axiom of Sufficiency for utility need to be explored, especially in the context of generic differential privacy. We also plan to explore axioms concerning prior beliefs that an attacker may possess. Currently many privacy definitions have not been expressed formally enough to apply an axiomatic approach. We feel this makes them into privacy goals rather privacy definitions and additional work is needed to formalize them so that they can be analyzed under the same mathematical lens as differential privacy.

6 Acknowledgments

We would like to thank Adam Smith from Penn State University for helpful discussions. We would like to thank Ilya Mironov for his observations of variations of differential privacy that are inconsistent with our privacy axioms (see Section 2.2). We thank Stephen Fienberg for pointing out connections between Bayesian sufficiency and our sufficiency axiom (which we earlier called the generality axiom). We also thank the anonymous reviewers for many helpful comments, for the insights about the relationship between the axiom of sufficiency and computational notions of privacy, and for the utility measure discussed in Example 3.2.4. This material is based upon work supported by the National Science Foundation under Grant No. 1054389.

Appendix A: Characterizing Generic Differential Privacy (necessary conditions)

In this section we prove necessary conditions on generic differential privacy (Definition 2.0.4) for satisfying Axioms 2.1.1 and 2.1.2. The main result is Theorem 0.0.6 which is a slightly stronger version of Theorem 2.1.3.

Fix i_1 and $i_2 \in \mathbb{I}$ such that $(i_1, i_2) \in \mathcal{R}$. Recall that if $\mathfrak{M}$ is a privacy mechanism for q-generic differential privacy (Definition 2.0.4), then by considering any $O \subseteq \mathbb{O}$ and its complement, we must have $\mathrm{q}(a, b) = T$ and $\mathrm{q}(1-a, 1-b) = T$ (where $a \equiv P_{\mathfrak{M}}(O|i_1)$, and $b \equiv P_{\mathfrak{M}}(O|i_2)$). Thus we can replace $\mathrm{q}(a, b)$ with $\mathrm{q}'(a, b) \equiv \mathrm{q}(a, b) \wedge \mathrm{q}(1-a, 1-b)$ without changing the privacy definition. Our results will thus characterize what q' should look like for a given predicate q to be usable in Definition 2.0.4. The following two results are useful because they will allow us to convert the predicate q' into real-valued functions.

Proposition 0.0.3. *Suppose there exists a mechanism $\mathfrak{M}$ that satisfies* q*-generic differential privacy. Axiom 2.1.1 implies:*

- $\mathrm{q}'(a, a) = T$ *for all* $a \in [0, 1]$.
- *If* $\mathrm{q}'(a, b) = T$ *and* $a \leq 1/2$ *then* $\mathrm{q}'(a, b') = T$ *for all* b' *between* b *and* $\frac{(1-b)a}{1-a}$.
- *If* $\mathrm{q}'(a, b) = T$ *and* $a \geq 1/2$ *then* $\mathrm{q}'(a, b') = T$ *for all* b' *between* b *and* $1 - \frac{b(1-a)}{a}$.

Proof. First note that the existence of a privacy mechanism $\mathfrak{M}$ implies that $\mathrm{q}'(1, 1) = T$ (and therefore $\mathrm{q}'(0, 0) = T$) since by Axiom 2.1.1, $\mathcal{A}_1 \circ \mathfrak{M}$ must satisfy privacy whenever $\mathcal{A}_1$ returns the same value o with probability 1 for any input. Now consider $\mathcal{A}_2$ which, on input o outputs o_1 with probability c and o_2 with probability $1 - c$. Then $P_{\mathcal{A}_2 \circ \mathcal{A}_1 \circ \mathfrak{M}}(o_1 \mid i) = c$ for any input i. Since $\mathcal{A}_2 \circ \mathcal{A}_1 \circ \mathfrak{M}$ must satisfy generic differential privacy (by Axiom 2.1.1), we must have $\mathrm{q}'(c, c) = T$ for $c \in [0, 1]$.

Now create an output space with two points: o_1 and o_2. Define $\mathfrak{M}$ as a randomized algorithm that (1) on input i_1 outputs o_1 with probability a and o_2 with probability $1 - a$; and (2) on input i_2 outputs o_1 with probability b and o_2 with probability $1 - b$. Clearly $\mathfrak{M}$ satisfies q'-generic differential privacy.

Consider the class of randomized algorithms $\mathcal{A}_{c,d}$ indexed by $c, d \in [0, 1]$ such that (1) on input o_1, $\mathcal{A}$ outputs o_1 with probability c and o_2 with probability $1 - c$; and (2) on input o_2, $\mathcal{A}$ outputs o_1 with probability d and o_2 with probability $1 - d$.

For the case where $a \leq 1/2$, set $d = (1 - c)a/(1 - a)$. Then as c increases continuously from 0 to 1, d decreases continuously from $a/(1 - a)$ to 0. At the same time $P_{\mathcal{A}_{c,d} \circ \mathfrak{M}}(o_1 \mid i_1) = a$ while $P_{\mathcal{A}_{c,d} \circ \mathfrak{M}}(o_1 \mid i_2)$ ranges continuously from $(1 - b)a/(1 - a)$ to b. Axiom 2.1.1 then implies that $\mathcal{A}_{c,d} \circ \mathfrak{M}$ satisfies q'-generic differential privacy and so $\mathrm{q}'(a, b') = T$ for all b' between b and $(1 - b)a/(1 - a)$.

For the case where $a \geq 1/2$, we apply our previous result to $1 - a$ and $1 - b$. Thus $q(1-a, 1-b') = T$ for all $1-b'$ between $1-b$ and $b(1-a)/a$ (and thus all b' between b and

$1-b(1-a)/a)$. Axiom 2.1.1 then implies that $\mathcal{A}_{c,d}\circ\mathfrak{M}$ satisfies q'-generic differential privacy and so $q(a,b')=q(1-(1-a),1-(1-b'))=T$ for all b' between b and $1-b(1-a)/a$. □

The significance of Proposition 0.0.3 is that it allows us to show that for each a, the set of b values that make $\mathrm{q}'(a,b)=T$ is actually an interval. We prove this in Proposition 0.0.4.

Proposition 0.0.4. *If there exists a mechanism $\mathfrak{M}$ satisfying q'-generic differential privacy then Axiom 2.1.1 implies that there exist functions $M_{\mathrm{q}'}$ and $m_{\mathrm{q}'}$ such that for all $a\in[0,1]$, $\mathrm{q}'(a,b)=T$ when $m_{\mathrm{q}'}(a)<b<M_{\mathrm{q}'}(a)$ and $\mathrm{q}'(a,b)=F$ whenever $b<m_{\mathrm{q}'}(a)$ or $b>M_{\mathrm{q}'}(a)$.*

Proof. By Proposition 0.0.3, for each a there is a b value such that $\mathrm{q}'(a,b)=T$. Now, note that if $a\leq 1/2$ and $b\leq a$ then $(1-b)a/(1-a)\geq a$, and if $b\geq a$ then $(1-b)a/(1-a)\leq a$. Similarly, if $a\geq 1/2$ and $b\leq a$ then $1-\frac{b(1-a)}{a}\geq a$, and if $b\geq a$ then $1-\frac{b(1-a)}{a}\leq a$.

Fix an $a\in[0,1]$. For each b, Proposition 0.0.3 gives an interval $[low(b),high(b)]$ which contains both b and a such that $\mathrm{q}'(a,b')=T$ whenever $b'\in[low(b),high(b)]$. Thus for all b where $\mathrm{q}'(a,b)=T$, the corresponding intervals overlap. Thus $\bigcup\limits_{b:\ \mathrm{q}'(a,b)=T}[low(b),high(b)]$ is an interval such that $\mathrm{q}'(a,b)=T$ if and only if b belongs to this interval. The proof is completed by defining

$$m_{\mathrm{q}'}(a)=\inf\bigcup_{b:\ \mathrm{q}'(a,b)=T}[low(b),high(b)]$$

and

$$M_{\mathrm{q}'}(a)=\sup\bigcup_{b:\ \mathrm{q}'(a,b)=T}[low(b),high(b)].$$

□

Thus when the input pair (i_1,i_2) is in our privacy relation $\mathcal{R}$, then given $P_{\mathfrak{M}}(o|i_1)$ there is an interval of allowable values for $P_{\mathfrak{M}}(o|i_2)$. However, the endpoints of the interval may or may not be allowable values. Keeping track of which endpoints are allowable and which are not will greatly complicate the presentation of our ideas, and so we introduce the following proposition which will help simplify things.

Proposition 0.0.5. *Let $\mathfrak{M}$ be a privacy mechanism satisfying q'-generic differential privacy and Axiom 2.1.1. Let $M_{\mathrm{q}'}$ and $m_{\mathrm{q}'}$ be the functions associated with q' by Proposition 0.0.4. Let ${\mathrm{q}'}^*$ be a predicate such that ${\mathrm{q}'}^*(a,b)=T$ if $b=M_{\mathrm{q}'}(a)$ or $b=m_{\mathrm{q}'}(a)$ and let ${\mathrm{q}'}^*(a,b)=\mathrm{q}'(a,b)$ otherwise. Then $\mathfrak{M}$ satisfies ${\mathrm{q}'}^*$-generic differential privacy and $M_{\mathrm{q}'^*}=M_{\mathrm{q}'}$ and $m_{\mathrm{q}'^*}=m_{\mathrm{q}'}$.*

Proof. The fact that $\mathfrak{M}$ satisfies ${\mathrm{q}'}^*$-generic differential privacy follows directly from the definition of generic differential privacy. The rest of the statements follow from the continuity of the $low(b)$ and $high(b)$ functions introduced in the proof of Proposition 0.0.4. □

Thus when studying the properties of $m_{\mathrm{q}'}$ and $M_{\mathrm{q}'}$ only, we can assume without loss of generality that $\mathrm{q}'(a,b) = T$ if and only if $m_{\mathrm{q}'} \leq b \leq M_{\mathrm{q}'}(a)$. The addition of Axiom 2.1.2 now ensures that the $M_{\mathrm{q}'}$ and $m_{\mathrm{q}'}$ functions have nice properties.

Theorem 0.0.6. *For* q*-generic differential privacy with neighbor relation* $\mathcal{R}$ *and privacy predicate* q*, define* $\mathrm{q}'(a,b) \equiv \mathrm{q}(a,b) \wedge \mathrm{q}(1-a, 1-b)$*. If there exists a privacy mechanism* $\mathfrak{M}$ *(with output space* $\mathbb{O}_{\mathfrak{M}}$*) satisfying* q*-generic differential privacy then:*

(i) Axiom 2.1.1 implies that there exist functions M *and* m *such that for any* $O \subseteq \mathbb{O}_{\mathfrak{M}}$*:*

$$\begin{aligned} M(a) > b > m(a) \quad &\Rightarrow \quad \mathrm{q}'(a,b) = T \\ b > M(a) \ \textit{or}\ b < m(a) \quad &\Rightarrow \quad \mathrm{q}'(a,b) = F \end{aligned}$$

where $a = P_{\mathfrak{M}}(O \mid i_1)$ *and* $b = P_{\mathfrak{M}}(O \mid i_2)$*.*

(ii) Axiom 2.1.1 implies

$$1 \geq M(a) \geq a \geq m(a) \geq 0.$$

(iii) Axiom 2.1.1 implies

$$M(a) \geq m(a) = 1 - M(1-a).$$

(iv) Axioms 2.1.1 and 2.1.2 imply M *is concave and* m *is convex.*

(v) Axiom 2.1.1 implies M *is nondecreasing and is strictly increasing at any point* a *where* $M(a) < 1$*.* m *is nonincreasing and is strictly decreasing at any point* a *where* $m(a) > 0$*.*

(vi) Axioms 2.1.1 and 2.1.2 imply M *is continuous except possibly at* $a = 0$ *and* m *is continuous except possibly at* $a = 1$*.*

Proof. Fix two points $i_1, i_2 \in \mathbb{I}$ from the input space of $\mathfrak{M}$ such that $(i_1, i_2) \in \mathcal{R}$. Item (i) is just Proposition 0.0.4. Item (ii) follows easily from the fact that $\mathrm{q}'(a,a) = T$ (Proposition 0.0.3). We next prove Item (iii) by using Proposition 0.0.5; we have $m(a) \leq b \leq M(a) \Leftrightarrow \mathrm{q}'(a,b) = T \Leftrightarrow \mathrm{q}'(1-a, 1-b) = T \Leftrightarrow m(1-a) \leq 1-b \leq M(1-a)$ so that $M(a)$ is the maximum allowable value of b if and only if $m(1-a)$ is the minimum allowable value of $1-b$. Item (iii) now follows.

To prove item (iv), consider $a_1 \neq a_2$. Again we invoke Proposition 0.0.5: let $\mathfrak{M}_1$ be the privacy mechanism such that $P_{\mathfrak{M}_1}(o_1 \mid i_1) = a_1$, $P_{\mathfrak{M}_1}(o_2 \mid i_1) = 1 - a_1$, $P_{\mathfrak{M}_1}(o_1 \mid i_2) = M(a_1)$ and $P_{\mathfrak{M}_1}(o_2 \mid i_2) = 1 - M(a_1)$. Similarly, let $\mathfrak{M}_2$ be the privacy mechanism such that $P_{\mathfrak{M}_2}(o_1 \mid i_1) = a_2$, $P_{\mathfrak{M}_2}(o_2 \mid i_1) = 1 - a_2$, $P_{\mathfrak{M}_2}(o_1 \mid i_2) = M(a_2)$ and $P_{\mathfrak{M}_2}(o_2 \mid i_2) = 1 - M(a_2)$. It is easy to see that $\mathfrak{M}_1$ and $\mathfrak{M}_2$ satisfy q'-generic differential privacy. Now choose a $c \in [0,1]$ and define $\mathfrak{M}_c$ to be the mechanism that runs $\mathfrak{M}_1$ with probability c and $\mathfrak{M}_2$ with probability $1-c$. Axiom 2.1.2 implies that $\mathfrak{M}_c$ satisfies q$'$-generic differential privacy. Now, $P_{\mathfrak{M}_c}(o_1 \mid i_1) = ca_1 + (1-c)a_2$ and $P_{\mathfrak{M}_c}(o_1 \mid i_2) = cM(a_1) + (1-c)M(a_2)$. Proposition 0.0.4 and the fact that $\mathfrak{M}_c$ satisfies q$'$-generic differential privacy then implies $M(ca_1 + (1-c)a_2) \geq cM(a_1) + (1-c)M(a_2)$ and so M is concave. The convexity of m then follows from Item (iii).

To prove item (v), choose a such that $M(a) < 1$ and define the mechanism $\mathfrak{M}$ such that $P_{\mathfrak{M}}(o_1 \mid i_1) = a$, $P_{\mathfrak{M}}(o_2 \mid i_1) = 1-a$, $P_{\mathfrak{M}}(o_1 \mid i_2) = M(a)$, and $P_{\mathfrak{M}}(o_2 \mid i_2) = 1-M(a)$ (again we are invoking Proposition 0.0.5). For $0 < c < 1$, define the randomized algorithm $\mathcal{A}_c$ such that $P_{\mathcal{A}_c}(o_1 \mid o_1) = 1$, $P(o_1 \mid o_2) = c$, and $P(o_2 \mid o_2) = 1-c$. Then by Axiom 2.1.1 $\mathcal{A}_c \circ \mathfrak{M}$ satisfies q$'$-generic differential privacy. Now, $P_{\mathcal{A}_c \circ \mathfrak{M}}(o_1 \mid i_1) = a + c(1-a) > a$ while $M(a + c(1-a)) \geq P_{\mathcal{A}_c \circ \mathfrak{M}}(o_1 \mid i_2) = M(a) + c(1-M(a)) > M(a)$. Thus M is strictly increasing at any point a where $M(a) < 1$. If $M(a) = 1$ but $a < 1$ then $P_{\mathcal{A}_c \circ \mathfrak{M}}(o_1 \mid i_1) = a + c(1-a) > a$ and $M(a + c(1-a)) \geq P_{\mathcal{A}_c \circ \mathfrak{M}}(o_1 \mid i_2) = M(a) + c(1-M(a)) = 1$, and so $M(a + c(1-a)) = 1$ and therefore M is nondecreasing. The corresponding result for m follows from Item (iii).

We now prove Item (vi). Since M is concave (as a result of Axioms 2.1.1 and 2.1.2), a basic continuity result from convexity theory [4] states that M is continuous on the open interval $(0,1)$ (i.e., the relative interior of its domain). Continuity at $a = 1$ follows from the fact that M is nondecreasing and so any discontinuity at 1 would be a jump discontinuity with $M(1) > \epsilon + M(a)$ for some $\epsilon > 0$ and all $a < 1$. This contradicts the fact that M is concave. The corresponding result for m follows from Item (iii). □

Appendix B: Characterizing Generic Differential Privacy (sufficient conditions)

In this section we prove a slightly stronger version of Theorem 2.1.4.

Theorem 0.0.7. *Let $\mathcal{R}$ be a neighbor relation. Let M and m be functions with the following properties:*

(i) $m(a) = 1 - M(1-a)$.

(ii) M *is concave (and* m *is convex).*

(iii) M *is continuous on* $(0,1]$ *(and* m *is continuous on* $[0,1)$*).*

(iv) $M(0) \geq 0$ *and* $M(1) = 1$ *(*$m(0) = 0$ *and* $m(1) \leq 1$*).*

Define $\mathrm{q}(a,b) = T$ *if and only if* $m(a) \leq b \leq M(a)$*. Then* q*-generic differential privacy (Definition 2.0.4) satisfies Axioms 2.1.1 and 2.1.2.*

Proof. Note that Items (ii) and (iv) and the fact that M is bounded by 1 ensures that M is strictly increasing except where it equals 1. Let $\mathfrak{M}$, $\mathfrak{M}_1$, $\mathfrak{M}_2$ be privacy mechanisms satisfying q-generic differential privacy with the same input space $\mathbb{I}$ (the existence of such mechanisms is implied by the concavity and nonnegativity of M, along with $M(1) = 1$, since then any $\mathfrak{M}$ whose output is independent of the input satisfies q-generic differential privacy). Fix two points $i_1, i_2 \in \mathbb{I}$ from the input space of $\mathfrak{M}$ such that $(i_1, i_2) \in \mathcal{R}$.

Implication of Axiom 2.1.1: Transformation Invariance. Choose a randomized algorithm $\mathcal{A}$ (whose input space is the output space of $\mathfrak{M}$) and consider an arbitrary measurable subset S of the output space of $\mathcal{A}$. Let μ_1 be the probability measure $P_{\mathfrak{M}}(\cdot \mid i_1)$ and let μ_2 be the probability measure $P_{\mathfrak{M}}(\cdot \mid i_2)$. Let h_S be the measurable function $P_{\mathcal{A}}(S \mid \cdot)$ and note that $0 \le h_S \le 1$. Let $a = P_{\mathcal{A}\,\circ\,\mathfrak{M}}(S|i_1)$ and $b = P_{\mathcal{A}\,\circ\,\mathfrak{M}}(S|i_2)$. Note that $a = \int h_S(x)\; d\mu_1(x)$ and $b = \int h_s(x)\; d\mu_2(x)$. Our goal is to prove $m(a) \le b \le M(a)$. For any measurable subset X of the output space of $\mathfrak{M}$, we will use the notation I_X to denote the indicator function which is 1 on $x \in X$ and 0 otherwise.

Step 1 Suppose $h_S(x) = I_X(x)$ for some measurable subset X of the output space of $\mathfrak{M}$. Then $a = P_{\mathcal{A}\,\circ\,\mathfrak{M}}(S|i_1) = \int h_S(x)\; d\mu_1(x) = \mu_1(X) = P_{\mathfrak{M}}(X|i_1)$ and similarly $b = P_{\mathfrak{M}}(X|i_2)$, and so since $\mathfrak{M}$ satisfies abstract differential privacy, $m(a) \le b \le M(a)$. On the other hand, if $h_S(x) \equiv 0$ then $P_{\mathcal{A}\,\circ\,\mathfrak{M}}(S|i_1) = 0$ and $P_{\mathcal{A}\,\circ\,\mathfrak{M}}(S|i_2) = 0$. Item (iv) now implies $m(P_{\mathcal{A}\,\circ\,\mathfrak{M}}(S \mid i_1)) \le P_{\mathcal{A}\,\circ\,\mathfrak{M}}(S \mid i_2) \le M(P_{\mathcal{A}\,\circ\,\mathfrak{M}}(S \mid i_1))$.

Step 2. We will now prove the theorem for the case when $h_S(x)$ is a *simple function*, i.e., $h_S(x) = \sum\limits_{j=1}^{n} c_j I_{X_j}(x)$ where the X_j are pairwise disjoint measurable subsets of the output space of $\mathfrak{M}$ and the $c_j \in [0, 1]$. Without loss of generality, assume $c_n \le \cdots \le c_1$ and for notational convenience, define $c_{n+1} = 0$. In this case,

$$\begin{aligned} a &= P_{\mathcal{A}\,\circ\,\mathfrak{M}}(S|i_1) = \int h_S(x)\; d\mu_1(x) = \sum_{j=1}^{n} c_j \mu_1(X_j), \\ b &= P_{\mathcal{A}\,\circ\,\mathfrak{M}}(S|i_2) = \int h_S(x)\; d\mu_2(x) = \sum_{j=1}^{n} c_j \mu_2(X_j). \end{aligned}$$

We can rewrite a and b as follows:

$$a = c_1 \sum_{j=1}^{n} \frac{c_j - c_{j+1}}{c_1} \mu_1\left(\bigcup_{\ell=1}^{j} X_\ell\right), \tag{1}$$

$$b = c_1 \sum_{j=1}^{n} \frac{c_j - c_{j+1}}{c_1} \mu_2\left(\bigcup_{\ell=1}^{j} X_\ell\right) \tag{2}$$

and note that the factors $\frac{c_j - c_{j+1}}{c_1}$ are nonnegative, sum up to 1, and therefore define a convex combination. From Step 1 we have for all j:

$$m\left(\mu_1\left(\bigcup_{\ell=1}^{j} X_\ell\right)\right) \le \mu_2\left(\left(\bigcup_{\ell=1}^{j} X_\ell\right)\right) \le M\left(\mu_1\left(\bigcup_{\ell=1}^{j} X_\ell\right)\right).$$

Thus

$$\begin{aligned}
m(a) &= m\left(c_1\sum_{j=1}^{n}\frac{c_j-c_{j+1}}{c_1}\mu_1\left(\bigcup_{\ell=1}^{j}X_\ell\right)\right)\\
&\leq c_1 m\left(\sum_{j=1}^{n}\frac{c_j-c_{j+1}}{c_1}\mu_1\left(\bigcup_{\ell=1}^{j}X_\ell\right)\right)\\
&\ \text{(since } m \text{ is convex and } m(0)=0\text{)}\\
&\leq c_1\sum_{j=1}^{n}\frac{c_j-c_{j+1}}{c_1}m\left(\mu_1\left(\bigcup_{\ell=1}^{j}X_\ell\right)\right)\\
&\ \text{(by convexity of } m\text{)}\\
&\leq c_1\sum_{j=1}^{n}\frac{c_j-c_{j+1}}{c_1}\mu_2\left(\bigcup_{\ell=1}^{j}X_\ell\right)\\
&\ \text{by Step 1}\\
&= b \quad \text{(by Equation 2).}
\end{aligned}$$

$$\begin{aligned}
M(a) &= M\left(c_1\sum_{j=1}^{n}\frac{c_j-c_{j+1}}{c_1}\mu_1\left(\bigcup_{\ell=1}^{j}X_\ell\right)\right)\\
&\geq c_1 M\left(\sum_{j=1}^{n}\frac{c_j-c_{j+1}}{c_1}\mu_1\left(\bigcup_{\ell=1}^{j}X_\ell\right)\right)\\
&\ \text{(since } M \text{ is concave and } M(0)\geq 0\text{)}\\
&\geq c_1\sum_{j=1}^{n}\frac{c_j-c_{j+1}}{c_1}M\left(\mu_1\left(\bigcup_{\ell=1}^{j}X_\ell\right)\right)\\
&\geq c_1\sum_{j=1}^{n}\frac{c_j-c_{j+1}}{c_1}\mu_2\left(\bigcup_{\ell=1}^{j}X_\ell\right)\\
&= b.
\end{aligned}$$

Step 3. We now prove the theorem for arbitrary measurable $h_S(x)$. By Theorem 1.17 in [42], there exists a sequence $h_S^{(1)}, h_S^{(2)}, \dots$ of simple functions such that for all x, $0 \leq h_S^{(1)}(x) \leq h_S^{(2)}(x) \leq \cdots \leq h_S(x)$ and $\lim_{n\to\infty} h_S^{(n)}(x) \to h_S(x)$. By the Lebesgue Monotone Convergence Theorem [42],

$$\begin{aligned}
\lim_{n\to\infty}\int h_S^{(n)}(x)\ d\mu_1(x) &\to \int h_S(x)\ d\mu_1(x) = P_{\mathcal{A}\circ\mathfrak{M}}(S|i_1),\\
\lim_{n\to\infty}\int h_S^{(n)}(x)\ d\mu_2(x) &\to \int h_S(x)\ d\mu_2(x) = P_{\mathcal{A}\circ\mathfrak{M}}(S|i_2).
\end{aligned}$$

From Step 2 we have: $m\left(\int h_S^{(n)}(x)\ d\mu_1(x)\right) \leq \int h_S^{(n)}(x)\ d\mu_2(x) \leq M\left(\int h_S^{(n)}(x)\ d\mu_1(x)\right)$. The continuity of M (except at 0) then implies that $M\left(\int h_S(x)\ d\mu_1(x)\right) \geq \int h_S(x)\ d\mu_2(x)$, except possibly in the case when $\int h_S(x)\ d\mu_1(x) = 0$. However, $\int h_S(x)\ d\mu_1(x) = 0$ implies that $h_S(x) \equiv 0$ except possibly on a set X with $\mu_1(X) = 0$ (since $h_S(x)$ cannot be negative); this case is covered by Step 1.

Similarly, the continuity of m (except at 1) then implies that $m\left(\int h_S(x)\ d\mu_1(x)\right) \leq \int h_S(x)\ d\mu_2(x)$, except possibly in the case when $\int h_S(x)\ d\mu_1(x) = 1$. However, since $h_S(x) \leq 1$ then $\int h_S(x)\ d\mu_1(x) = 1$ implies that $h_S(x) = I_X(x)$ for some measurable set X and so this case is also covered by Step 1.

Implication of Axiom 2.1.2: Convexity. Now consider privacy mechanisms $\mathfrak{M}_1$ and $\mathfrak{M}_2$ with the same input space. Choose $c \in [0,1]$ and define $\mathfrak{M}_c$ as the randomized algorithm that on input i returns $\mathfrak{M}_1(i)$ with probability c and $\mathfrak{M}_2(i)$ with probability $1-c$ (independently of the input). Let S be an arbitrary measurable subset of the union of the output spaces of $\mathfrak{M}_1$ and $\mathfrak{M}_2$. Thus $m(P_{\mathfrak{M}_1}(S \mid i_1)) \leq P_{\mathfrak{M}_1}(S \mid i_2) \leq M(P_{\mathfrak{M}_1}(S \mid i_1))$ and $m(P_{\mathfrak{M}_2}(S \mid i_1)) \leq P_{\mathfrak{M}_2}(S \mid i_2) \leq M(P_{\mathfrak{M}_2}(S \mid i_1))$. Now, $P_{\mathfrak{M}_p}(S \mid i_1) = pP_{\mathfrak{M}_1}(S \mid i_1) + (1-p)P_{\mathfrak{M}_2}(S \mid i_1)$ and $P_{\mathfrak{M}_p}(S \mid i_2) = pP_{\mathfrak{M}_1}(S \mid i_2) + (1-p)P_{\mathfrak{M}_2}(S \mid i_2)$. By the convexity of m and concavity of M, we have

$$\begin{aligned}
m(P_{\mathfrak{M}_p}(S \mid i_1)) &= m(pP_{\mathfrak{M}_1}(S \mid i_1) + (1-p)P_{\mathfrak{M}_2}(S \mid i_1)) \\
&\leq pm(P_{\mathfrak{M}_1}(S \mid i_1)) + (1-p)m(P_{\mathfrak{M}_2}(S \mid i_1)) \\
&\leq pP_{\mathfrak{M}_1}(S \mid i_2) + (1-p)P_{\mathfrak{M}_2}(S \mid i_2) \\
&= P_{\mathfrak{M}_p}(S \mid i_2). \\
M(P_{\mathfrak{M}_p}(S \mid i_1)) &\geq pM(P_{\mathfrak{M}_1}(S \mid i_1)) + (1-p)M(P_{\mathfrak{M}_2}(S \mid i_1)) \\
&\geq pP_{\mathfrak{M}_1}(S \mid i_2) + (1-p)P_{\mathfrak{M}_2}(S \mid i_2) \\
&= P_{\mathfrak{M}_p}(S \mid i_2).
\end{aligned}$$

□

Appendix C: Generating Predicates for Generic Differential Privacy

In this appendix, we restate and prove Theorems 2.2.1 and 2.2.2.

Theorem. *(2.2.1). For any set $S \subset \mathbb{R}$, let S_{-t} denote the set $\{s-t \ : \ s \in S\}$. Let X be a random variable with density $\kappa e^{-c(x)}$, where $c(x)$ is convex. Consider the optimization problem*

$$\arg\max_{S \subseteq \mathbb{R}} P(X \in S_{-t}) - P(X \in S)$$

subject to $P(X \in S) = p$. This quantity is maximized by the set $\{r \ : \ r \geq G^{-1}(p)\}$.

Proof. In this proof we use the fact that convexity of a function f is equivalent to the statement that for $x > y$, $\frac{f(x)-f(y)}{x-y}$ is nondecreasing as a function of x and is also nondecreasing as a function of y (as long as $y < x$).

Let $T = \{r \ : \ r \geq G^{-1}(p)\}$ and let S be any other set such that $P(X \in S) = p$ and $P(T \Delta S) > 0$ (where Δ is the symmetric difference). Then there exists a $q > 0$, a set $V \subseteq T \setminus S$, and a set $W \subseteq S \setminus T$ such that $P(V) = P(W) = q \leq p$. By definition of T, for every $v \in V$ and $w \in W$, $v \geq G^{-1}(a) \geq w$ which, by convexity of c, implies $c(v) - c(v-t) \geq c(w) - c(w-t)$ and therefore $\exp(c(v) - c(v-t)) \geq \exp(c(w) - c(w-t))$. Choose a ϕ such that $\exp(c(v) - c(v-t)) \geq \phi \geq \exp(c(w) - c(w-t))$ for all $v \in V$ and $w \in W$. Now,

$$\begin{aligned} P(V_{-t}) &= \int_V e^{-c(x-t)}\ dx = \int_V e^{c(x)-c(x-t)} e^{-c(x)}\ dx \\ &\geq \int_V \phi e^{-c(x)}\ dx = \phi q \\ &= \int_W \phi e^{-c(x)}\ dx \geq \int_W e^{c(x)-c(x-t)} e^{-c(x)}\ dx \\ &= \int_W e^{-c(x-t)}\ dx = P(W_{-t}). \end{aligned}$$

which means that to maximize our objective, we need to remove W from S and insert V instead. Thus any such set S cannot be the optimal solution. The fact that T is the optimal solution follows from a limiting argument using the fact that any measurable S can be approximated arbitrarily closely from above by a finite union of intervals (which can be converted to a single interval covering the right tail using the previous argument finitely many times). □

Theorem. *(2.2.2) Given a random variable X with density $\kappa e^{-c(x)}$, where c is a convex function, let $G(x) = P(X \geq x)$. For a fixed $t > 0$, let $M(a) = G(G^{-1}(a) - t)$. Then M satisfies the following properties:*

- *M is concave.*
- *M is strictly increasing whenever $M(a) < 1$.*
- *M is continuous except possibly at 0.*
- *$M(a) \geq a$.*

Proof. The fact that $M(a) \geq a$ is obvious (e.g., see Figures 2 and 3). By continuity of the density function, it is also clear that $\lim_{a \to 0} M(a) = 0$ and $\lim_{a \to 1} M(a) = 1$, so if $M(a)$ is concave then it must also be strictly increasing whenever $M(a) < 1$. Thus we need to show that $M(a)$ is concave.

In this proof we use the fact that convexity of a function f is equivalent to the statement that for $x > y$, $\frac{f(x)-f(y)}{x-y}$ is increasing as a function of x and is also increasing as a function of y (as long as $y < x$).

Note that $\frac{dG}{dx} = -\kappa e^{-c(x)}$, and so G^{-1} is differentiable on $(0,1)$ and $\frac{dG^{-1}(a)}{da} = \frac{-1}{\kappa e^{-c\left((G^{-1}(a)\right)}}$. Using these equalities,

$$\begin{aligned}
M(a) &= \int_{G^{-1}(a)-t}^{\infty} \kappa e^{-c(x)}\ dx = 1 - \int_{-\infty}^{G^{-1}(a)-t} \kappa e^{-c(x)}\ dx \\
\frac{dM(a)}{da} &= -\kappa e^{-c\left(G^{-1}(a)-t\right)} \frac{dG^{-1}(a)}{da} = \frac{\kappa e^{-c\left(G^{-1}(a)-t\right)}}{\kappa e^{-c(G^{-1}(a))}} \\
&= \exp\left(-\left[c\left(G^{-1}(a)-t\right) - c\left(G^{-1}(a)\right)\right]\right).
\end{aligned}$$

Thus as a increases, $\left[c\left(G^{-1}(a)-t\right) - c\left(G^{-1}(a)\right)\right]$ also increases (by convexity of c) and so the derivative of M decreases. This is equivalent to M being concave. □

Appendix D: Validity of Dobrushin's Coefficient of Ergodicity

Here we prove that the negative Dobrushin coefficient of ergodicity, defined as $\mu_{Dob}(\mathfrak{M}) = -\min_{j,k} \sum\limits_i \min(m_{i,j}, m_{i,k})$, satisfies the relation $\mu_{Dob}(\mathcal{A}\,\mathfrak{M}) \leq \mu_{Dob}(\mathfrak{M})$.

Proof. Let $\mathfrak{M}$ be a privacy mechanism with finite input and output spaces. We represent $\mathfrak{M}$ as a column-stochastic matrix representation $\{m_{i,j}\}$ (see Figure 9). Let $\mathcal{A}$ be a randomized algorithm with column-stochastic matrix representation $\{p_{i,j}\}$, with appropriate dimensions so that the product $\mathcal{A}\,\mathfrak{M}$ makes sense.

Below, we will use the fact that min is concave and $c\min(x_1, x_2) = \min(cx_1, cx_2)$ for $c \geq 0$ from which it follows that $\min(\sum_{i=1}^r p_i x_i) \geq \sum_{i=1}^r p_i \min(x_i)$ when $p_i \geq 0$ for all i.

$$\begin{aligned}
\sum_i \min(m_{i,j}, m_{i,k}) &= \sum_i \sum_\ell p_{\ell,i} \min\left(m_{i,j}, m_{i,k}\right) \\
&= \sum_\ell \sum_i p_{\ell,i} \min\left(m_{i,j}, m_{i,k}\right) \\
&\leq \sum_\ell \min\left(\sum_i m_{i,j} p_{\ell,i}, \sum_i m_{i,k} p_{\ell,i}\right) \\
&= \sum_\ell \min\left(m'_{\ell,j}, m'_{\ell,k}\right)
\end{aligned}$$

where $\{m'_{\ell,j}\}$ is the matrix representation of $\mathcal{A}\,\mathfrak{M}$. Thus it follows that

$$\min_{j,k} \sum_i \min(m_{i,j}, m_{i,k}) \leq \min_{j,k} \sum_\ell \min\left(m'_{\ell,j}, m'_{\ell,k}\right)$$

and so $\mu_{Dob}(\mathfrak{M}) \geq \mu_{Dob}(\mathcal{A}\,\mathfrak{M})$.

□

Note that the same proof implies that

$$\max_{j,k}\sum_i \min(m_{i,j}, m_{i,k}) \leq \max_{j,k}\sum_\ell \min\left(m'_{\ell,j}, m'_{\ell,k}\right),$$

and so $\mu_{disc}(\mathfrak{M}) \geq \mu_{disc}(\mathcal{A}\mathfrak{M})$.

Appendix E: Maximally Sufficient Differentially Private Mechanisms with Finite Input Spaces

Theorem. (3.3.2) *For a given $\epsilon > 0$, finite input space $\mathbb{I}$, and symmetric neighbor relation $\mathcal{R}$ (it must be symmetric for differential privacy), let S be the set of all ϵ-differentially private mechanisms (with input space $\mathbb{I}$). Let S_{con} be the subset of S consisting of all mechanisms $\mathfrak{M}$ with finite output spaces and such that each row graph is connected (when viewed as an undirected graph). Then S_{con} is precisely the set of maximally sufficient differentially private mechanisms with finite output spaces.*

Proof. When viewed as a column stochastic matrix, no maximally sufficient mechanism can have an entry equal to 1 (the constraints for differential privacy would then imply that an entire row consists of entries equal to 1, meaning that the output of such a mechanism is constant). Such a mechanism is clearly not in S_{con} (the row containing all 1 entries is not connected).

We first show that mechanisms with finite output spaces excluded from S_{con} cannot be maximally sufficient. Let $\mathfrak{M}$ be a mechanism and let o be an output such that the corresponding row graph is not connected. This row can be decomposed into two disjoint components C_1 and C_2 such that there are no edges between them. Let

$$\begin{aligned}\rho_1 &= \max\{e^\delta : \exists s \in C_1, t \in C_2, P(o|s) = e^\delta P(o|t), (s,t) \in \mathcal{R}\}, &(3)\\ \rho_2 &= \max\{e^\delta : \exists s \in C_1, t \in C_2, P(o|t) = e^\delta P(o|s), (s,t) \in \mathcal{R}\}, &(4)\end{aligned}$$

and note that $0 < \rho_1 < e^\epsilon$ and $0 < \rho_2 < e^\epsilon$ (if either ρ_1 or ρ_2 were 0 then the whole row of $\mathfrak{M}$ would consist entirely of 0's and all constraints would be tight). Define

$$\begin{aligned}a &= \frac{(e^\epsilon/\rho_2) - 1}{(e^\epsilon/\rho_1)(e^\epsilon/\rho_2) - 1},\\ b &= \frac{(e^\epsilon/\rho_1) - 1}{(e^\epsilon/\rho_1)(e^\epsilon/\rho_2) - 1}.\end{aligned}$$

We form a new output space $\mathbb{O}' = \mathbb{O}\setminus\{o\} \uplus \{o_1, o_2\}$ (where $\uplus$ denotes disjoint union) by splitting o into two outputs, o_1 and o_2, and define mechanism $\mathfrak{M}'$ with output space

$\mathbb{O}'$ such that

$$P_{\mathfrak{M}'}(o' \mid s) \quad = \quad \begin{cases} P_{\mathfrak{M}}(o|s) \times ae^{\epsilon}/\rho_1 & \text{if } o' = o_1 \ \wedge \ s \in C_1 \\ P_{\mathfrak{M}}(o|s) \times a & \text{if } o' = o_1 \ \wedge \ s \in C_2 \\ P_{\mathfrak{M}}(o|s) \times b & \text{if } o' = o_2 \ \wedge \ s \in C_1 \\ P_{\mathfrak{M}}(o|s) \times be^{\epsilon}/\rho_2 & \text{if } o' = o_2 \ \wedge \ s \in C_2 \\ P_{\mathfrak{M}}(o' \mid s) & \text{if } o' \in \mathbb{O} \cap \mathbb{O}' \end{cases} .$$

Note that all of these are proper probabilities since a, b, ae^{ϵ}/ρ_1, and be^{ϵ}/ρ_2 are nonnegative and less than 1 since $e^{\epsilon} > \rho_1$ and $e^{\epsilon} > \rho_2$. Clearly we also must have for each fixed s, $\sum_{o' \in \mathbb{O}} P_{\mathfrak{M}'}(o' \mid s) = 1$.

Let $\mathcal{A}$ be a randomized algorithm such that $\mathcal{A}(o') = o'$ if $o' \in \mathbb{O} \cap \mathbb{O}'$ and $\mathcal{A}(o') = o$ if $o' \in \{o_1, o_2\}$. We claim that:

- $\mathfrak{M} \preceq_{\mathrm{S}} \mathfrak{M}'$. Proof: clearly $\mathfrak{M} = \mathcal{A} \circ \mathfrak{M}'$.
- $\mathfrak{M}'$ satisfies differential privacy: for any $(s,t) \in \mathcal{R}$, if $s,t \in C_1$ then clearly $\frac{P_{\mathfrak{M}'}(o_i|s)}{P_{\mathfrak{M}'}(o_i|t)} = \frac{P_{\mathfrak{M}}(o|s)}{P_{\mathfrak{M}}(o|t)}$ for $i = 1,2$ (and similarly for $s,t \in C_2$). If $s \in C_1$ and $t \in C_2$, then since the row corresponding to o cannot have 0 entries:

  $$\frac{P_{\mathfrak{M}'}(o_1|s)}{P_{\mathfrak{M}'}(o_1|t)} \quad = \quad P_{\mathfrak{M}}(o|s)/P_{\mathfrak{M}}(o|t) \times e^{\epsilon}/\rho_1 \leq e^{\epsilon} \qquad (5)$$

  since $P_{\mathfrak{M}}(o|s)/P_{\mathfrak{M}}(o|t) \leq \rho_1$. We reach a similar conclusion for $s \in C_2$ and $t \in C_1$. The results for o_2 are similar.
- The row graph of o with respect to $\mathfrak{M}$ is a proper subgraph of the row graphs of o_1 and o_2 with respect to $\mathfrak{M}'$: since there is no edge between C_1 and C_2 in the row graph of o with respect to $\mathfrak{M}$, then the previous argument shows that any edge present in the row graph for o is also present in the row graphs for o_1 and o_2. Also note that in Equation 5, equality is achieved for the s and t that achieve the maximum in Equation 3. Thus the row graph for o_1 has an additional edge. Similarly, the row graph for o_2 has an additional edge.
- The randomized algorithm $\mathcal{A}$ defined above is not reversible, so $\mathfrak{M}'$ is sufficient for $\mathfrak{M}$ but not vice versa (the proof is obvious).

Repeating this procedure finitely many times (the number is at most the number of spanning trees in the privacy relation $\mathcal{R}$ when viewed as a graph), we get a privacy mechanism that belongs to S_{con}. Thus there can be no maximally sufficient differentially private mechanism with finite output space that does not belong to S_{con}.

To show that every $\mathfrak{M} \in S_{con}$ is maximally sufficient, first note that if any two rows of $\mathfrak{M}$ are proportional to each other (which can only happen if the corresponding row graphs are the same), we can form a mechanism $\mathfrak{M}_2$ which is the same as $\mathfrak{M}$, except that those two rows are replaced by one row containing their sum. It is easy to see that

$\mathfrak{M} \preceq_{\mathrm{S}} \mathfrak{M}'$ and $\mathfrak{M}' \preceq_{\mathrm{S}} \mathfrak{M}$. Thus for this part of the proof it is enough to assume that no two rows of $\mathfrak{M}$ are proportional to each other and no two row graphs are the same.

Now, suppose there exists a privacy mechanism $\mathfrak{M}'$ with output space $\mathbb{O}'$ and a randomized algorithm $\mathcal{A}$ such that $P_{\mathcal{A} \circ \mathfrak{M}'} = P_{\mathfrak{M}}$ for some $\mathfrak{M} \in S_{con}$ with output space $\mathbb{O}$. For any $o \in \mathbb{O}$, define $\mathcal{A}^{-}(o) \equiv \{o' \in \mathbb{O}' \; : \; P_{\mathcal{A}}(o \mid o') > 0\}$ (it is a poor man's inverse). It is easy to see that every measurable $O' \subseteq \mathcal{A}^{-}(o)$, and any $(i_1, i_2) \in \mathcal{R}$, $\frac{P_{\mathfrak{M}}(o|i_1)}{P_{\mathfrak{M}}(o|i_2)} = \frac{P_{\mathfrak{M}'}(O'|i_1)}{P_{\mathfrak{M}'}(O'|i_2)}$ whenever the denominator of the right hand side is nonzero (in which case the numerator must also be positive by the differential privacy requirements). This is because the tightness constraints in the row graph for o determine (up to a constant factor) all the probabilities $P_{\mathfrak{M}}(o \mid \cdot)$, and no positive combination of non-tight constraints can yield a tight constraint for differential privacy. This implies that the conditional probability $P_{\mathfrak{M}'}(O' \mid \mathcal{A}^{-}(o), i)$ is independent of the input i.

Since no other row in $\mathfrak{M}$ has the same row graph as o (without loss of generality) and all other row graphs are connected and therefore each represent a maximal set of tight constraints in differential privacy, we see that $P_{\mathcal{A}}(o \mid o') = 1$ for all $o' \in \mathcal{A}^{-}(o)$. This implies that $P_{\mathfrak{M}}(o \mid i) = P_{\mathfrak{M}'}(\mathcal{A}^{-}(o) \mid i)$ for any $i \in \mathbb{I}$.

Thus we can define a randomized algorithm $\mathcal{A}_2$ that for any $o \in \mathbb{O}$ and $O' \subseteq \mathcal{A}^{-}(o)$, we have $P_{\mathcal{A}_2}(O' \mid o) = P_{\mathfrak{M}'}(O' \mid \mathcal{A}^{-}(o), i)$ (for any $i \in \mathbb{I}$, since this quantity does not depend on i). Using the fact that $P_{\mathfrak{M}}(o \mid i) = P_{\mathfrak{M}'}(\mathcal{A}^{-}(o) \mid i)$ for any $i \in \mathbb{I}$, it is each to check that $P_{\mathfrak{M}'} = P_{\mathcal{A}_2 \circ \mathfrak{M}}$ and so $\mathfrak{M}' \preceq_{\mathrm{S}} \mathfrak{M}$. Thus we have shown that every $\mathfrak{M} \in S_{con}$ is maximally sufficient.

□

This result can be extended to mechanisms with finite input spaces but infinite output spaces. For a finite input space $\mathbb{I} = \{i_1, \ldots, i_n\}$ we need to consider the Radon-Nikodyn derivative [42] of the measure $P(\cdot \mid i_j)$ with respect to the base measure $\frac{1}{n} \sum\limits_{k=1}^{n} P(\cdot \mid i_k)$ and use a similar proof technique. Since there are only finitely many row graphs, it is then also straightforward to show that every mechanism $\mathfrak{M}_1$ with finite input space but infinite output space is equivalent to a mechanism $\mathfrak{M}_2$ with finite output space in the sense that $\mathfrak{M}_2 = \mathcal{A} \circ \mathfrak{M}_1$ and $\mathfrak{M}_1 = \mathcal{A}' \circ \mathfrak{M}_2$ for some $\mathcal{A}$ and $\mathcal{A}'$ (where equality is interpreted as equality between the induced conditional probability distributions of inputs and outputs).

Appendix F: Characterizing Branching Measures

Theorem. (3.2.8). *A utility metric μ is a branching measure if and only if it satisfies Axioms 3.1.4, 3.2.5 and 3.2.6.*

Proof. Clearly branching measures satisfy 3.2.5, and 3.2.6. By Theorem 2.3.1 of [20], a function satisfying Axioms 3.1.4 (which makes the function symmetric in its inputs) and 3.2.6 must have the form $\sum\limits_{j=1}^{n} F(P_{\mathfrak{M}}(o_j \mid \cdot))$. Thus the only thing left to prove is that $F(c\vec{x}) = cF(\vec{x})$ and F is convex (over vectors with nonnegative components) if and only if μ satisfies the axiom of sufficiency.

Since we are dealing with mechanisms with finite input and output spaces, postprocessing the output of $\mathfrak{M}$ by $\mathcal{A}$ is equivalent to using the algorithm $\mathcal{A} \circ \mathfrak{M} = \mathcal{A}\,\mathfrak{M}$ (in matrix notation). Matrix multiplication can be reduced to a chain of two basic operations. The first operation takes a row $\vec{x}$ and replaces it with two copies $p\vec{x}$ and $(1-p)\vec{x}$ (where $1 \geq p \geq 0$). The second operation takes two rows $\vec{x}$ and $\vec{y}$ and replaces them by $\vec{x} + \vec{y}$.

The first operation takes a matrix $\mathfrak{M}$ with rows $P_{\mathfrak{M}}(o_1 \mid \cdot), \dots, P_{\mathfrak{M}}(o_j \mid \cdot), \dots P_{\mathfrak{M}}(o_n \mid \cdot)$ and creates a matrix $\mathfrak{M}'$ with rows $P_{\mathfrak{M}'}(o_1 \mid \cdot)$, $\dots$, $P_{\mathfrak{M}'}(o_{j-1} \mid \cdot)$, $pP_{\mathfrak{M}'}(o'_j \mid \cdot)$, $(1-p)P_{\mathfrak{M}'}(o''_j \mid \cdot)$, $P_{\mathfrak{M}'}(o_{j+1} \mid \cdot)$, $\dots$, $P_{\mathfrak{M}'}(o_n \mid \cdot)$ (by splitting row j). Since we can merge the two new rows to recover $\mathfrak{M}$, then $\mathfrak{M}$ and $\mathfrak{M}'$ are sufficient for each other and have the same utility. This means that $F(\vec{x}) = F(p\vec{x}) + F((1-p)\vec{x})$ for all $\vec{x}$ and $1 \geq p \geq 0$.

The second operation takes a matrix $\mathfrak{M}$ with rows $P_{\mathfrak{M}}(o_1 \mid \cdot)$, $\dots$, $P_{\mathfrak{M}}(o_i \mid \cdot)$, $\dots$, $P_{\mathfrak{M}}(o_j \mid \cdot)$, $\dots$, $P_{\mathfrak{M}}(o_n \mid \cdot)$ and creates a matrix $\mathfrak{M}'$ with rows $P_{\mathfrak{M}}(o_1 \mid \cdot)$, $\dots$, $P_{\mathfrak{M}}(o_{i-1} \mid \cdot)$, $\left[P_{\mathfrak{M}}(o_i \mid \cdot) + P_{\mathfrak{M}}(o_j \mid \cdot)\right]$, $P_{\mathfrak{M}}(o_{i+1} \mid \cdot)$, $\dots$, $P_{\mathfrak{M}}(o_{j-1} \mid \cdot)$, $P_{\mathfrak{M}}(o_{j+1} \mid \cdot)$, $\dots$, $P_{\mathfrak{M}}(o_n \mid \cdot)$ (by merging rows i and j). This implies that $F(\vec{x}) + F(\vec{y}) \geq F(\vec{x} + \vec{y})$.

Thus all we need to show is that if F is continuous then $F(\vec{x} + \vec{y}) \leq F(\vec{x}) + F(\vec{y})$ and $F(\vec{x}) = F(p\vec{x}) + F((1-p)\vec{x})$ if and only if F is convex and $F(p\vec{x}) = pF(\vec{x})$. The "if" direction is obvious, so now we prove the "only if" direction.

It is clear that $F(\vec{x}) = F(p\vec{x}) + F((1-p)\vec{x})$ implies $F(\vec{x}) = kF(\vec{x}/k)$ for any positive integer k and therefore $F(\frac{p}{q}\vec{x}) = \frac{p}{q}F(\vec{x})$ for positive integers p, q as long as $\vec{x}$ and $(p/q)\vec{x}$ do not have components outside the interval $[0,1]$. By continuity, $F(p\vec{x}) = pF(\vec{x})$. Then from $F(\vec{x} + \vec{y}) \leq F(\vec{x}) + F(\vec{y})$ we get $F(p\vec{x} + (1-p)\vec{y}) \leq pF(\vec{x}) + (1-p)F(\vec{y})$. □

References

[1] Adam, N. and Worthmann, J. (1989). Security-control methods for statistical databases. *ACM Computing Surveys*, 21(4): 515–556.

[2] Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., and Talwar, K. (2007). Privacy, accuracy, and consistency too: A holistic solution to contingency table release. In *PODS*.

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

[4] Bertsekas, D. P., Nedic, A., and Ozdaglar, A. E. (2003). *Convex Analysis and Optimization*. Athena Scientific.

[5] Blum, A., Dwork, C., McSherry, F., and Nissim, K. (2005). Practical privacy: the sulq framework. In *PODS*, 128–138.

[6] Blum, A., Ligett, K., and Roth, A. (2008). A learning theory approach to non-interactive database privacy. In *STOC*, 609–618.

[7] Carnap, R. and Jeffrey, R. C. (eds.) (1971). *Studies in Inductive Logic and Probability*, volume I. University of California Press.

[8] Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Duxbury, 2nd edition.

[9] Chaudhuri, K. and Mishra, N. (2006). When random sampling preserves privacy. In *Proceedings of the International Cryptology Conference*.

[10] Chen, B.-C., Kifer, D., LeFevre, K., and Machanavajjhala, A. (2009). Privacy-preserving data publishing. *Foundations and Trends in Databases*, 2(1-2): 1–167.

[11] Cohen, J. E., Derriennic, Y., and Zbaganu, G. (1993). Majorization, monotonicity of relative entropy and stochastic matrices. *Contemporary Mathematics*, 149.

[12] DeGroot, M. H. and Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *The Statistician*, 32.

[13] Dinur, I. and Nissim, K. (2003). Revealing information while preserving privacy. In *PODS*.

[14] Dwork, C. (2006). Differential privacy. In *ICALP*, volume 4051 of *Lecture Notes in Computer Science*, 1–12.

[15] Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006). Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, 486–503.

[16] Dwork, C., Mcsherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, 265–284.

[17] Dwork, C., McSherry, F., and Talwar, K. (2007). The price of privacy and the limits of lp decoding. In *STOC*, 85–94.

[18] Dwork, C., Naor, M., Reingold, O., N.Rothblum, G., and Vadhan, S. (2009). On the complexity of differentially private data release: Efficient algorithms and hardness results. In *STOC*, 381–390.

[19] Dwork, C. and Nissim, N. (2004). Privacy-preserving datamining on vertically partitioned databases. In *CRYPTO*.

[20] Ebanks, B. R., Sahoo, P. K., and Sander, W. (1997). *Characterizations of Information measures*. World Scientific Publishing Co.

[21] Evfimievski, A., Fagin, R., and Woodruff, D. P. (2008). Epistemic privacy. In *PODS*.

[22] Fung, B., Wang, K., Chen, R., and Yu, P. (2010). Privacy-preserving data publishing: A survey on recent developments. *ACM Computing Surveys*, 42(4).

[23] Ganta, S. R., Kasiviswanathan, S. P., and Smith, A. (2008). Composition attacks and auxiliary information in data privacy. In *KDD*.

[24] Ghosh, A., Roughgarden, T., and Sundararajan, M. (2009). Universally utility-maximizing privacy mechanisms. In *STOC*, 351–360.

[25] Gupte, M. and Sundararajan, M. (2010). Universally optimal privacy mechanisms for minimax agents. In *PODS*.

[26] Hay, M., Rastogi, V., Miklau, G., and Suciu, D. (2010). Boosting the accuracy of differentially-private histograms through consistency. In *VLDB*.

[27] Kifer, D. (2009). Attacks on privacy and de finetti's theorem. In *SIGMOD*.

[28] Kifer, D. and Lin, B.-R. (2010). Towards an axiomatization of statistical privacy and utility. In *PODS*.

[29] — (2010). Towards an axiomatization of statistical privacy and utility. Technical Report CSE-10-002, Penn State University.

[30] Kumar, R., Novak, J., Pang, B., and Tomkins, A. (2007). On anonymizing query logs via token-based hashing. In *WWW*.

[31] Machanavajjhala, A., Gehrke, J., and Götz, M. (2009). Data publishing against realistic adversaries. *VLDB*.

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

[33] McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In *FOCS*, 94–103.

[34] Mironov, I., Pandey, O., Reingold, O., and Vadhan, S. (2009). Computational differential privacy. In *Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*.

[35] Narayanan, A. and Shmatikov, V. (2006). How to break anonymity of the netflix prize dataset.
URL `http://www.citebase.org/abstract?id=oai:arXiv.org:cs/06\% 10105`

[36] — (2008). Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy (SP)*.

[37] Nergiz, M. E. and Clifton, C. (2007). Thoughts on k-anonymization. *Data & Knowledge Engineering*, 63(3): 622–645.

[38] Nissim, K., Raskhodnikova, S., and Smith, A. (2007). Smooth sensitivity and sampling in private data analysis. In *STOC*, 75–84.

[39] Nix, C. J. and Paris, J. B. (2006). A continuum of inductive methods arising from a generalized principle of instantial relevance. *Journal of Philosophical Logic*, 35(1): 83–115.

[40] Rastogi, V., Hay, M., Miklau, G., and Suciu, D. (2009). Relationship privacy: Output perturbation for queries with joins. In *PODS*, 107–116.

[41] Rastogi, V., Suciu, D., and Hong, S. (2007). The boundary between privacy and utility in data publishing. In *VLDB*, 531–542.

[42] Rudin, W. (1987). *Real & Complex Analysis*. McGraw-Hill, 3rd edition.

[43] Samarati, P. (2001). Protecting respondents' identities in microdata release. *TKDE*, 13(6).

[44] Sweeney, L. (2002). k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5): 557–570.

[45] Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*.

[46] Wong, R., Fu, A., Wang, K., and Pei, J. (2007). Minimality attack in privacy preserving data publishing. In *VLDB*.

[47] Xiao, X., Wang, G., and Gehrke, J. (2010). Differential privacy via wavelet transforms. In *ICDE*.