Geometry of Privacy and Utility

Bing-Rong Lin and Daniel Kifer

Abstract—One of the important challenges in statistical privacy is the design of algorithms that maximize a utility measure subject to restrictions imposed by privacy considerations. In this paper we examine large classes of privacy definitions and utility measures. We identify their geometric characteristics and some common properties of optimal privacy-preserving algorithms.

1 Introduction

Improvements in data collection technology have been accompanied by demonstrations of the importance of data-driven approaches to making business, policy, and social decisions. The need to use and share large data sets has also raised privacy concerns. Statistical privacy is a multi-disciplinary field that studies how to reveal useful information contained in these data sets while preventing inference about sensitive information (such as the record of a specific individual or a business secret).

As the study of "information" progresses, evolving ideas about privacy lead to new privacy definitions (i.e., restrictions on the behavior of data-processing algorithms to guarantee limits on adversarial inference) and new ways of measuring the quality of the outputs of privacy-preserving algorithms (i.e. utility).

As a consequence, the central optimization problem – designing algorithms that maximize utility subject to privacy constraints – keeps changing. Because of this changing landscape, it is important to identify optimization principles that remain invariant as privacy definitions and utility metrics change.

Even basic properties of optimal solutions can differ. For example, under some combinations of privacy definition/utility measure, if one is interested in a query answer then optimal privacy preserving algorithms should have as many possible output values as there are query answers. For other privacy definition/utility measure combinations, the optimal privacy-preserving algorithm must have strictly more possible outputs (contrary to a common intuition that the outputs should be in one-to-one correspondence with query answers).

Recent research about desirable properties of privacy definitions and utility measures has identified generic mathematical classes they can belong to. In this paper we discuss the geometry of these classes of privacy definitions and utility measures, and identify geometric properties possessed by the corresponding optimal privacy-preserving algorithms.

The goal of this paper is to present a new perspective on the central optimization problem in statistical privacy. We hope its main role is that of raising (rather than answering) additional interesting questions.

In Section 2, we introduce terminology and notation, including a convenient matrix view of randomized algorithms. In Section 3, we discuss *conic* privacy definitions – a large class of privacy definitions that subsumes many, but not all existing definitions. In Section 4, we show that for reasonable information-preserving utility measures, one can always find an optimal conic privacy-preserving algorithm with linearly independent conditional probability vectors (in particular, this implies the existence of optimal algorithms whose range and domain have the same size); this is not necessarily true for non-conic privacy definitions. In Section 5 we discuss geometric interpretations of a class of utility measures called *branching measures* and in Section 6 we discuss interactions between the geometries of privacy and utility.

2 Notation and Terminology

Let $\mathbb{I} = \{D_1, D_2, \dots\}$ be a countable collection of possible input datasets. Let $\mathbb{R}^{|\mathbb{I}|}_{\geq 0}$ be the set of vectors of dimension $|\mathbb{I}|$ with no negative components. Let $\vec{1}$ be the vector in $\mathbb{R}^{|\mathbb{I}|}_{\geq 0}$ whose components are all 1.

A *sanitizing algorithm* $\mathfrak{M}$ is a deterministic or randomized algorithm whose domain is $\mathbb{I}$ and whose range is countable.

For convenience, we represent a sanitizing algorithm $\mathfrak{M}$ as a matrix where the columns are indexed by $\mathbb{I}$, rows are indexed by the countable set $\text{range}(\mathfrak{M})$, and whose entries are $P(\mathfrak{M}(D) = \omega)$:

$$\begin{array}{c c} & \begin{array}{ccc} D_1 & D_2 & \dots \end{array} \\ \begin{array}{c} \omega_1 \\ \omega_2 \\ \omega_3 \\ \vdots \end{array} & \begin{pmatrix} P(\mathfrak{M}(D_1) = \omega_1) & P(\mathfrak{M}(D_2) = \omega_1) & \dots \\ P(\mathfrak{M}(D_1) = \omega_2) & P(\mathfrak{M}(D_2) = \omega_2) & \dots \\ P(\mathfrak{M}(D_1) = \omega_3) & P(\mathfrak{M}(D_2) = \omega_3) & \dots \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array}$$

We use the notation $P(\mathfrak{M}(\cdot) = \omega)$ to refer to the vector $\langle P(\mathfrak{M}(D_1) = \omega),\ P(\mathfrak{M}(D_2) = \omega), \dots \rangle$, which is the row of the matrix form of $\mathfrak{M}$ that is indexed by ω.

We define the following operators:

Operator 2.1 **($\mathcal{A} \circ \mathfrak{M}$).** *When the domain of an (possibly randomized) algorithm $\mathcal{A}$ contains the range of $\mathfrak{M}$, then $\mathfrak{M}' \equiv \mathcal{A} \circ \mathfrak{M}$ is their composition: $\mathfrak{M}'(D) = \mathcal{A}(\mathfrak{M}(D))$.*

Operator 2.2 ($\mathfrak{M}_1 \oplus_p \mathfrak{M}_2$)**.** *When $\mathfrak{M}_1$ and $\mathfrak{M}_2$ have the same domain and $p \in [0,1]$, then $\mathfrak{M}' \equiv \mathfrak{M}_1 \oplus_p \mathfrak{M}_2$ is the algorithm that runs $\mathfrak{M}_1$ with probability p and $\mathfrak{M}_2$ with probability $1-p$ and reveals which algorithm was run.*

Operator 2.3 ($p\,\mathfrak{M}_1 + (1-p)\,\mathfrak{M}_2$)**.** *When $\mathfrak{M}_1$ and $\mathfrak{M}_2$ have the same domain and $p \in [0,1]$, then $\mathfrak{M}' \equiv p\,\mathfrak{M}_1 + (1-p)\,\mathfrak{M}_2$ is the algorithm that runs $\mathfrak{M}_1$ with probability p and $\mathfrak{M}_2$ with probability $1-p$.*

A *privacy definition* $\mathfrak{Priv}$ is a set of sanitizing algorithms with input domain $\mathbb{I}$. Intuitively, it is the set of algorithms trusted to process the sensitive input data without leaking too much sensitive information.

A *utility measure* $\mu_{\mathbb{I}}$ is a function that assigns a real number to sanitizing algorithms whose input domain is $\mathbb{I}$.

The sanitizing mechanism design problem is to (possibly approximately) solve the following optimization problem: $\underset{\mathfrak{M}\in\mathfrak{Priv}}{\operatorname{argmax}}\, \mu_{\mathbb{I}}(\mathfrak{M})$.

3 Conic Privacy Definitions

In this paper, we are investigating the sanitizing mechanism design problem over conic privacy definitions. This is a class of privacy definitions that includes differential privacy [1], pufferfish [2], and essentially all privacy definitions $\mathfrak{Priv}$ that satisfy several common-sense properties [3] and always (not just with high probability) bound information leakage to an attacker [3].

Definition 3.1 (Privacy Cone)**.** *A closed set $C \subseteq \mathbb{R}^{|\mathbb{I}|}_{\geq 0}$ is a privacy cone if it contains the vector $\vec{1}$ and is closed under vector addition and multiplication by scalars ≥ 0.*

Definition 3.2 (Conic Privacy Definition)**.** *A privacy definition $\mathfrak{Priv}$ is* conic *if there exists a privacy cone* C *such that $\mathfrak{M} \in \mathfrak{Priv}$ if and only if every row of the matrix form of $\mathfrak{M}$ belongs to* C *(i.e. $P(\mathfrak{M}(\cdot) = \omega) \in \mathrm{C}$ for all $\omega \in \mathrm{range}(\mathfrak{M})$)*

An example is differential privacy.

Definition 3.3 (Differential Privacy [1])**.** *$\mathfrak{M}$ belongs to the set of ϵ-differentially private algorithms if for every $\omega \in \mathrm{range}(\mathfrak{M})$ and pair of datasets D, D' that differ on the value of one record, $P(\mathfrak{M}(D) = \omega) \leq e^{\epsilon} P(\mathfrak{M}(D') = \omega)$.*

However, the following variant is not conic.

Definition 3.4 ((ϵ,δ)-Differential Privacy [4])**.** *$\mathfrak{M}$ satisfies (ϵ,δ)-differential privacy if for every set $S \subseteq \mathrm{range}(\mathfrak{M})$ and pair of datasets D, D' that differ on the value of one record, $P(\mathfrak{M}(D) \in S) \leq e^{\epsilon} P(\mathfrak{M}(D') \in S) + \delta$.*

4 Utility and Linear Independence

In this section we study properties of solutions to the sanitizing mechanism design equation $\mathfrak{M}^* = \underset{\mathfrak{M}\in\mathfrak{Priv}}{\operatorname{argmax}}\, \mu_{\mathbb{I}}(\mathfrak{M})$ when $\mathfrak{Priv}$ is a conic privacy definition. When $\mathbb{I}$ is finite, we show that for a large class of utility measures, we can restrict our attention to algorithms $\mathfrak{M}^*$ whose matrix form consists of linearly independent rows (hence, $\mathrm{range}(\mathfrak{M}^*) \leq |\,\mathbb{I}\,|$). We then show that this is not necessarily the case for non-conic privacy definitions (e.g., (ϵ,δ)-differential privacy).

We consider utility measures that satisfy the axioms of sufficiency, continuity, and quasi-convexity, which are defined as follows.

Axiom 4.1. (Sufficiency [5])**.** *$\mu_{\mathbb{I}}(\mathfrak{M}_1) \geq \mu_{\mathbb{I}}(\mathfrak{M}_2)$ whenever $\mathfrak{M}_2 = \mathcal{A} \circ \mathfrak{M}_1$ for some $\mathcal{A}$.*

The intuition behind sufficiency is that $\mathfrak{M}_1$ can be used to simulate $\mathfrak{M}_2$ (with the help of a post-processing algorithm $\mathcal{A}$). If $\mathfrak{M}_2$ is useful for some task, then $\mathfrak{M}_1$ can be used instead.

Axiom 4.2. (Continuity [5])**.** *$\mu_{\mathbb{I}}$ should be continuous with respect to the metric $\mathrm{d}^*_{\mathbb{I}}$, where $\mathrm{d}^*_{\mathbb{I}}(\mathfrak{M}_1, \mathfrak{M}_2)$ equals:*

$$\sup_{D\in\mathbb{I}} \sum_{\omega} \Big| P[\mathfrak{M}_1(D) = \omega] - P[\mathfrak{M}_2(D) = \omega] \Big|$$

Continuity states that small changes to the probabilistic behavior of an algorithm results in small changes to its utility.

Axiom 4.3. (Quasi-convexity [6])**.** *$\mu_{\mathbb{I}}(\mathfrak{M}_1 \oplus_p \mathfrak{M}_2) \leq \max\{\mu_{\mathbb{I}}(\mathfrak{M}_1), \mu_{\mathbb{I}}(\mathfrak{M}_2)\}$ for all $\mathfrak{M}_1, \mathfrak{M}_2$ and $p \in [0,1]$.*

The intuition behind quasi-convexity is that if we prefer $\mathfrak{M}_2$ over $\mathfrak{M}_1$, then we should also prefer $\mathfrak{M}_2$ over $\mathfrak{M} \equiv \mathfrak{M}_1 \oplus_p \mathfrak{M}_2$, since $\mathfrak{M}$ sometimes behaves like $\mathfrak{M}_2$ but otherwise behaves like the less preferred algorithm $\mathfrak{M}_1$.

We now arrive at the main result of this section.

Theorem 4.4. *Let $\mathbb{I}$ be finite, let $\mathfrak{Priv}$ be conic, and let $\mu_{\mathbb{I}}$ satisfy Axioms 4.1, 4.2, and 4.3. Then the problem $\operatorname{argmax}_{\mathfrak{M}\in\mathfrak{Priv}} \mu_{\mathbb{I}}(\mathfrak{M})$ has a solution $\mathfrak{M}^*$ whose matrix form consists of linearly independent rows.*

Proof: This proof is divided into three steps.

Step 1: We first show that if a sanitizing algorithm $\mathfrak{M}$ has finite range then there exists a $\mathfrak{M}' \in \mathfrak{Priv}$ whose matrix representation consists of linearly independent rows and $\mu_{\mathbb{I}}(\mathfrak{M}') \geq \mu_{\mathbb{I}}(\mathfrak{M})$.

Without loss of generality, we may assume the matrix form of $\mathfrak{M}$ has no rows that are constant multiples of each other (if it does, we can merge those rows and the algorithm $\mathfrak{M}^{\dagger}$ that corresponds to the resulting matrix form has $\mu_{\mathbb{I}}(\mathfrak{M}^{\dagger}) = \mu_{\mathbb{I}}(\mathfrak{M})$ since $\mathfrak{M} = A_1 \circ \mathfrak{M}^{\dagger}$ for some A_1 and $\mathfrak{M}^{\dagger} = A_2 \circ \mathfrak{M}$ for some A_2).

If the matrix form of $\mathfrak{M}$ has full row rank then we are done (i.e. $\mathfrak{M}' = \mathfrak{M}$). Thus we need to consider $\mathfrak{M}$ with linearly dependent rows. Let $r_1, \ldots, r_m$ be the rows of the matrix form of $\mathfrak{M}$. Without loss of generality, assume the linear dependency is among the first $n+1$ rows (re-ordering rows as necessary):

$$c_1 r_1 + \ldots + c_L r_L = c_{L+1} r_{L+1} + \ldots + c_{n+1} r_{n+1},$$

where (1) the c_i are all non-negative, (2) $c_1 \leq c_2 \leq \ldots \leq c_L$, (3) $c_{L+1} \leq c_{L+2} \leq \ldots \leq c_{n+1}$, and (4) $c_L = 1$

(since the r_i have no negative components and all the c_i are non-negative, clearly there are non-zero terms on both sides of the equation, so we can rescale it so that $c_L = 1$). We construct algorithms A and B such that $\mathfrak{M} = pA + (1-p)B$ for some $p \in [0,1]$. Define

$$\begin{aligned}
a_k &= (1-c_k)r_k, \text{ when } k < L \\
a_L &= 0 \\
a_k &= (1+c_k)r_k, \text{ when } L < k \le (n+1) \\
a_k &= r_k, \text{ when } (n+1) < k \le m \\
b_k &= (1+\frac{c_k}{c_{n+1}})r_k, \text{ when } k \le L \\
b_k &= (1-\frac{c_k}{c_{n+1}})r_k, \text{ when } L < k \le n \\
b_{n+1} &= 0 \\
b_k &= r_k, \text{ when } (n+1) < k \le m
\end{aligned}$$

and set $P(A(\cdot) = \omega_i) = a_i$ and $P(B(\cdot) = \omega_i) = b_i$ for all i. Note that A never outputs ω_L and B never outputs ω_{n+1} so their matrix forms have one less row than $\mathfrak{M}$. Also, by construction, all of the a_i and b_i are vectors with no negative components. This, along with the fact that the sum of the a_i is the vector whose entries are all 1 (and same for b_i) means that A and B are indeed algorithms (all of the necessary conditional probabilities add up to 1). Since the rows of A and B are rescalings of the rows of $\mathfrak{M}$, we have $A, B \in \mathfrak{Priv}$.

It is also easy to verify that $\mathfrak{M} = \frac{1}{1+c_{n+1}}A + \frac{c_{n+1}}{1+c_{n+1}}B$ and so by Axiom 4.1 and then Axiom 4.3, we have $\mu_{\mathbb{I}}(\mathfrak{M}) \le \mu_{\mathbb{I}}(A \oplus_{\frac{1}{1+c_{n+1}}} B) \le \max\{\mu_{\mathbb{I}}(A), \mu_{\mathbb{I}}(B)\}$. Since the range of $\mathfrak{M}$ is finite, we repeatedly apply this procedure to either A or B until we obtain a matrix $\mathfrak{M}'$ with independent rows such that $\mu_{\mathbb{I}}(\mathfrak{M}') \ge \mu_{\mathbb{I}}(\mathfrak{M})$.
Step 2: If the range of $\mathfrak{M}$ is countably infinite, we use Axiom 4.2 and to obtain a $\mathfrak{M}^{(j)}$ with finite range and $\mu_{\mathbb{I}}(\mathfrak{M}^{(j)}) \ge \mu_{\mathbb{I}}(\mathfrak{M}) - 1/j$. We then use Step 1 to obtain $\mathfrak{M}^{(j\dagger)}$ whose range is at most $|\mathbb{I}|$ (because its rows are linearly independent) and $\mu_{\mathbb{I}}(\mathfrak{M}^{(j\dagger)}) \ge \mu_{\mathbb{I}}(\mathfrak{M}^{(j)})$. Standard compactness arguments now imply some subsequence of the $\mathfrak{M}^{(j\dagger)}$ converge to a $\mathfrak{M}'$ with at most $|\mathbb{I}|$ rows and $\mu_{\mathbb{I}}(\mathfrak{M}') \ge \mu_{\mathbb{I}}(\mathfrak{M})$. Since conic privacy definitions use closed cones, $\mathfrak{M}' \in \mathfrak{Priv}$ (also, by step 1, we can then get linearly independent rows).
Step 3: Let $\mathfrak{M}_1, \mathfrak{M}_2, \dots$ be a sequence of algorithms with linearly independent rows such that $\mu_{\mathbb{I}}(\mathfrak{M}_1) \le \mu_{\mathbb{I}}(\mathfrak{M}_2) \le \dots$. Standard compactness arguments and continuity of $\mu_{\mathbb{I}}$ imply that a subsequence converges to a $\mathfrak{M}' \in \mathfrak{Priv}$ with at most $|\mathbb{I}|$ rows. Combined with steps 1 and 2, this fact implies the existence of an optimal $\mathfrak{M}^* \in \mathfrak{Priv}$ having linearly independent rows. □

Now let us consider a non-conic privacy definition such as (ϵ, δ)-differential privacy (where $\epsilon \neq 0$). Let $\mathbb{I} = \{0,1\}$ and consider the utility function $\mu_{\mathbb{I}}{}^{L2}(\mathfrak{M}) = \sum\limits_{\omega \in \text{range}(\mathfrak{M})} \sqrt{P(\mathfrak{M}(1)=\omega)^2 + P(\mathfrak{M}(2)=\omega)^2}$. It is continuous and satisfies Axiom 4.3 because the L_2 norm is convex. As we will see in Section 5, it also satisfies Axiom 4.1. It is straightforward to show that for every algorithm $\mathfrak{M}$ whose matrix form has linearly independent rows (and hence $|\text{range}(\mathfrak{M})| \le 2$), there exists another $\mathfrak{M}'$ with 3 or more possible outputs and strictly higher utility.[1]

Aside from having linearly independent rows, we can also ensure that the rows of an optimal algorithm are points on the boundary of the privacy cone (i.e. the least private among the acceptable choices of $P(\mathfrak{M}(\cdot) = \omega)$) rather than, say, points in the interior of the privacy cone yet at the boundary of the unit hypercube caused by the constraint $P(\mathfrak{M}(D_i) = \omega) \le 1$.

Theorem 4.5. *Let $\mathbb{I}$ be finite, let $\mathfrak{Priv}$ be conic with privacy cone C, and let $\mu_{\mathbb{I}}$ satisfy Axioms 4.1, 4.2, and 4.3. Then the problem* $\text{argmax}_{\mathfrak{M} \in \mathfrak{Priv}} \mu_{\mathbb{I}}(\mathfrak{M})$ *has a solution $\mathfrak{M}^*$ whose matrix form consists of linearly independent rows where each row comes from the boundary of C.*

Proof: Let $\mathfrak{M}^*$ be an algorithm with rows in C that maximizes $\mu_{\mathbb{I}}$. For each $\omega \in \text{range}(\mathfrak{M}^*)$, the vector $P(\mathfrak{M}^*(\cdot) = \omega)$ belongs to some finite portion of C (i.e. a subset of C containing all vectors with L_∞ norm less than some constant κ_ω. Thus, by Carathéodory's Theorem, $P(\mathfrak{M}^*(\cdot) = \omega)$ can be written as a convex combination $c_1\vec{x}_1 + \dots + c_r\vec{x}_r$ of $r \le |\mathbb{I}| + 1$ vectors from the boundary of C. We can modify $\mathfrak{M}^*$ so that instead of outputting ω (with probability vector $P(\mathfrak{M}^*(\cdot) = \omega)$), it produces new outputs $\omega^{(1)}, \dots \omega^{(r)}$ with probability vectors $P(\mathfrak{M}^*(\cdot) = \omega^{(i)}) = c_i\vec{x}_i$. Performing this modification for all $\omega \in \text{range}(\mathfrak{M}^*)$ for which $P(\mathfrak{M}^*(\cdot) = \omega)$ is in the interior of C results in an algorithm $\mathfrak{M}^\dagger$ whose rows all belong to the boundary of C and clearly there exists an $\mathcal{A}$ such that $\mathfrak{M}^* = \mathcal{A} \circ \mathfrak{M}^\dagger$ so that $\mu_{\mathbb{I}}(\mathfrak{M}^\dagger) \ge \mu_{\mathbb{I}}(\mathfrak{M}^*)$. Now we apply Theorem 4.4 to obtain from $\mathfrak{M}^\dagger$ a new algorithm $\mathfrak{M}$ whose rows are linearly independent vectors. These vectors also belong to the boundary of C because they are formed by taking scalar multiples and limits of subsequences of rows of $\mathfrak{M}^\dagger$. □

5 Geometric View of Utility

In this section we provide a geometric view of a large class of utility measures. We consider utility measures that satisfy Axioms 4.1, 4.2, and the following branching axiom (it turns out that quasi-convexity is a consequence of these three axioms).

Axiom 5.1. (Branching [5]). *An information preservation measure $\mu_{\mathbb{I}}$ should satisfy the relation*

$$\mu_{\mathbb{I}}(\mathfrak{M}) = \mu_{\mathbb{I}}\left(\mathfrak{M}^\dagger\right) + H\left(\vec{P}[\mathfrak{M}(\cdot) = \omega_1],\ \vec{P}[\mathfrak{M}(\cdot) = \omega_2]\right)$$

[1] The main idea is that if $\epsilon \neq 0$ and $\mathfrak{M}$ has two possible outputs then there exists some output $\omega \in \text{range}(\mathfrak{M})$ such that $0 < P(\mathfrak{M}(2) = \omega) \le P(\mathfrak{M}(1) = \omega)$ and $P(\mathfrak{M}(1) = \omega) \le e^\epsilon P(\mathfrak{M}(2) = \omega) + \delta$. This vector $\vec{z} \equiv P(\mathfrak{M}(\cdot) = \omega)$ can then be broken into two vectors $\vec{x}$ and $\vec{y}$, with $\vec{x} + \vec{y} = \vec{z}$ such that replacing $\vec{z}$ in the matrix representation of an algorithm with $\vec{x}$ and $\vec{y}$ will result in a new algorithm that still satisfies the privacy constraints but has strictly higher utility.

for some function H, where

- *ω_1 and ω_2 are two elements in* $\text{range}(\mathfrak{M})$.
- $\text{range}(\mathfrak{M}^\dagger) = \{\omega^*\} \cup \text{range}(\mathfrak{M})$ *and $\mathfrak{M}^\dagger$ behaves exactly like $\mathfrak{M}$ except that $\mathfrak{M}^\dagger$ outputs ω^* whenever $\mathfrak{M}$ would have output ω_1 or ω_2.*

Kifer and Lin [5] showed that if $\mathbb{I}$ is finite then a utility measure satisfies Axioms 4.1, 4.2, and 5.1 if and only if it has the form:

$$\mu_{\mathbb{I}}(\mathfrak{M}) \quad = \sum_{\omega \in \text{range}(\mathfrak{M})} f(P(\mathfrak{M}(\cdot) = \omega)) \qquad (1)$$

for some function f where $f(\vec{x}+\vec{y}) \leq f(\vec{x}) + f(\vec{y})$ and $f(c\vec{x}) = cf(\vec{x})$ for all vectors $\vec{x}, \vec{y} \in \mathbb{R}^{|\mathbb{I}|}_{\geq 0}$ and all $c \geq 0$. Since this implies that f is convex, quasi-convexity of $\mu_{\mathbb{I}}$ (Axiom 4.3) follows.

Based on Equation 1, one would like to think of f as "the amount of information per output" of $\mathfrak{M}$. However, the f in Equation 1 may be negative and f may not be minimized by the vector $\vec{1}$ (if $P(\mathfrak{M}(\cdot) = \omega) = \vec{1}$ then this output ω provides no information about the input to $\mathfrak{M}$ and so has no utility). This drawback can be fixed as follows.

Since f is convex over $\mathbb{R}^{|\mathbb{I}|}_{\geq 0}$, let $\vec{v}$ be a subgradient of f at the vector $\vec{1}$. Define $g(\vec{x}) = f(\vec{x}) - \vec{v} \cdot \vec{x}$. By definition of subgradient of a convex function, $g(\vec{x}) \geq g(\vec{1})$ for all vectors $\vec{x} \in \mathbb{R}^{|\mathbb{I}|}_{\geq 0}$. Note also that $cg(\vec{1}) = g(c\vec{1})$ for all $c \geq 0$ (a property g inherits from f). Combining these last two facts, we get $cg(\vec{1}) \geq g(\vec{1})$ for all $c \geq 0$ and hence $g(\vec{1}) = 0$. Furthermore,

$$\begin{aligned} \mu_{\mathbb{I}}(\mathfrak{M}) &= \sum_{\omega \in \text{range}(\mathfrak{M})} g(P(\mathfrak{M}(\cdot) = \omega)) + \vec{v} \cdot P(\mathfrak{M}(\cdot) = \omega) \\ &= \vec{v} \cdot \vec{1} + \sum_{\omega \in \text{range}(\mathfrak{M})} g(P(\mathfrak{M}(\cdot) = \omega)) \end{aligned}$$

To summarize, if $\mathbb{I}$ is finite, a utility measure satisfies Axioms 4.1, 4.2, and 5.1 if and only if it is equal, up to an additive constant, to the summation $\sum_{\omega \in \text{range}(\mathfrak{M})} g(P(\mathfrak{M}(\cdot) = \omega))$ for some g such that:

(i) g is continuous over $\mathbb{R}^{|\mathbb{I}|}_{\geq 0}$
(ii) $g(\vec{x}) \geq 0$ for all $\vec{x} \in \mathbb{R}^{|\mathbb{I}|}_{\geq 0}$
(iii) $g(\vec{1}) = 0$
(iv) $g(c\vec{x}) = cg(\vec{x})$ for all $c \geq 0$ and $\vec{x} \in \mathbb{R}^{|\mathbb{I}|}_{\geq 0}$
(v) $g(\vec{x}+\vec{y}) \leq g(\vec{x}) + g(\vec{y})$ for all $\vec{x}, \vec{y} \in \mathbb{R}^{|\mathbb{I}|}_{\geq 0}$

Thus g behaves like a seminorm over $\mathbb{R}^{|\mathbb{I}|}_{\geq 0}$, but in general, its domain cannot be extended to $\mathbb{R}^{|\mathbb{I}|}$ while maintaining the seminorm properties.[2]

Let $\mathcal{G} = \left\{\vec{x} \ : \ \vec{x} \in \mathbb{R}^{|\mathbb{I}|}_{\geq 0},\ g(\vec{x}) \leq 1\right\}$. It is easy to check that $\mathcal{G}$ is a utility envelope, defined as:

Definition 5.2 (Utility Envelope). *We say a set $\mathcal{G} \subseteq \mathbb{R}^{|\mathbb{I}|}_{\geq 0}$ is a* utility envelope *if it is a closed convex set containing a relatively open ball $\left\{\vec{x} \in \mathbb{R}^{|\mathbb{I}|}_{\geq 0} \ : \ ||\vec{x}||_2 < \delta\right\}$ (for some $\delta > 0$) and all vectors of the form $c\vec{1}$ for $c \geq 0$.*

From a utility envelope $\mathcal{G}$, one can reconstruct a g with properties (i), (ii), (iii), (iv), (v) mentioned above as follows: $g(\vec{x}) = \inf\{\lambda > 0 \mid \vec{x}/\lambda \in \mathcal{G}\}$.

The privacy cone and utility envelope give us geometric interpretations of privacy and utility, which we explore next.

6 Privacy/Utility Tradeoff Geometry

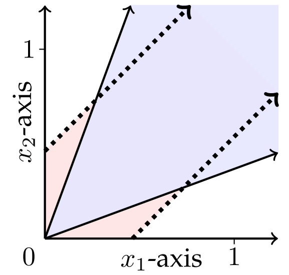

Fig. 1. Privacy cone (blue) intersecting a scaled utility envelope (between dotted lines).

A branching utility measure $\mu_{\mathbb{I}}$ assigns a utility score to each output of $\mathfrak{M}$ (see Equation 1) and $\mu_{\mathbb{I}}(\mathfrak{M})$ is the sum of those utility scores.

For a branching measure $\mu_{\mathbb{I}}$, let U be the utility envelope. For a conic privacy definition $\mathfrak{Priv}$ let C be the privacy cone. Based on the results of Theorem 4.5, the process of choosing a mechanism $\mathfrak{M} \in \mathfrak{Priv}$ that maximizes $\mu_{\mathbb{I}}$ can be thought of as the process of selecting constants $c_1, c_2, \ldots, c_r$ (where each c_i corresponds to the amount of utility provided by an output ω_i) and then choosing $P(\mathfrak{M}(\cdot) = \omega_i)$ as an $|\mathbb{I}|$-dimensional point in the intersection of the boundaries of C and c_iU (the utility envelope scaled by c_i), as shown in Figure 1.

Acknowledgments

This material is based on work supported by NSF grant number 1228669.

References

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis." in *TCC*, 2006.
[2] D. Kifer and A. Machanavajjhala, "A rigorous and customizable framework for privacy," in *PODS*, 2012.
[3] B.-R. Lin and D. Kifer, "Reasoning about privacy using axioms," in *Signals, Systems and Computers (ASILOMAR)*, 2012.
[4] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *STOC*, 2007.
[5] D. Kifer and B.-R. Lin, "An axiomatic view of statistical privacy and utility," *J. of Privacy and Confidentiality*, vol. 4, no. 1, 2012.
[6] B.-R. Lin and D. Kifer, "Information measures in statistical privacy and data processing applications," Penn State University, Tech. Rep., 2013.

[2] Any extension must deal with the fact $g(-\vec{1}) = |-1|g(\vec{1}) = 0$ and hence $g(\vec{x}+\vec{1}) \leq g(\vec{x})$ and $g(\vec{x}-\vec{1}) \leq g(\vec{x}) + g(-\vec{1}) = g(\vec{x})$ which together imply that $g(\vec{x}+c\vec{1}) = g(\vec{x})$ for all c. However, one frequently stipulates conditions such as the probability vector $P(\mathfrak{M}(\cdot) = \omega_1) \equiv (0.5, 0)$ provides strictly more information about the inputs than $P(\mathfrak{M}(\cdot) = \omega_1) \equiv (0.6, 0.1) = (0.5, 0) + 0.1(1, 1)$