PRIVACY ON A SUBSET OF DATASET VARIABLES

Pin Lin Tan
Department of Statistics
Singapore
TAN_Pin_Lin@singstat.gov.sg

ABSTRACT

Differential privacy (DP) is a mathematical definition of privacy that has many attractive properties. This includes allowing data providers to quantify the privacy risk of a collection of data releases from a dataset. As DP is defined for mechanisms acting on the dataset, the privacy guarantee of DP is at the level of the entire dataset and does not require the determination of which variables are sensitive or identifying. While this provides privacy guarantees that do not depend on the often-subjective classification of variables, this also leads to needing all outputs from the dataset to be noisy, even those that only involve variables that are not considered sensitive or identifying.

In this paper, we introduce privacy definitions that only protect a subset of the variables in a dataset, thus allowing statistics only involving unprotected variables to be released accurately. We show that the definitions are a generalisation and relaxation of DP and can be defined using the Pufferfish framework for privacy definitions, as well as provide an algorithm for generating privacy-protected count tables that are consistent with the counts that can be released accurately under the proposed privacy definition. We explore the properties of the definition in terms of composition and show how the definitions can be used to assign an epsilon value to each subset of variables. This contrasts with DP, which only assigns an epsilon value to the entire dataset. The epsilon value for a single variable is at most that of the entire dataset and can be smaller, reflecting that different variables in a dataset may be protected to different degrees in a data release. This may be useful for data providers who wish to protect some variables, for example more sensitive ones, more than others.

Keywords Differential Privacy · Privacy · Pufferfish

1 Introduction

Differential privacy (DP) [Dwork et al., 2006a] is a mathematical definition of privacy that has gained popularity in recent years due to its attractive properties. One of these properties is that it provides strong privacy guarantees that do not depend on assumptions about the information available to an adversary [Kasiviswanathan and Smith, 2014]. As such, there is no need to determine which variables of the data are identifying or sensitive. This is unlike other methods such as k-anonymity. Another attractive property of DP is that it allows data providers to quantify the privacy risk of a collection of data releases from a dataset.

However, the strong guarantees of DP come at a cost: all DP algorithm outputs are noisy. For data that has no disclosure risks, this is unnecessary information loss. There are also situations where the data provider may want to provide some data accurately. While a data provider can choose to protect some but not all of the data released from a dataset using DP, this leads to the collection of data releases not satisfying DP, making it difficult to characterise the privacy protection of the collection of data releases. This negates one of the key attractive properties of DP.

1.1 Contributions

In this paper, we:

1. Introduce a class of privacy definitions that only protect a subset of the variables in a dataset (Section 2). This allows statistics only involving unprotected variables to be released accurately while still satisfying a mathematical definition of privacy.
2. Show that the definitions are a generalisation and relaxation of DP and how the definitions can be used to assign an epsilon value to each subset of variables (Section 3). This allows data providers to understand the privacy risk for different subsets of their data, which may be useful for data providers who wish to protect some variables—for example more sensitive ones—more than others.
3. Show that the proposed definitions can be defined using the Pufferfish framework for privacy definitions (Section 4).
4. Provide an algorithm for generating count tables that satisfy the proposed definition while ensuring consistency with the counts that can be released accurately (Section 5).

2 Definitions

We first define the term *dataset* and related terms.

Definition 2.1 (Dataset)**.** Let $V = \{V_1, V_2, \dots, V_k\}$ be a set of sets and $X \subseteq \prod_{i=1}^{k} V_i$. A finite multiset D is said to be a *dataset over X and V* if each element of D is an element of X.

Each V_i is called a *dataset variable*, and V is called the *set of dataset variables*. Each element of D is called a *record*, X is called the *set of possible records*.

Remark 2.1.1. In the above definition, we allow for multisets so that datasets can contain more than one copy of a record. This is similar to the definition of databases in [Dwork et al., 2014], where databases are represented by their histograms.

Remark 2.1.2. A dataset as defined above can be thought of as a table, where each column is a dataset variable and each row is a record.

Using the above definition of a dataset, we introduce a new privacy definition below.

Definition 2.2 (ε-privacy on a subset of variables)**.** Let $V = \{V_1, V_2, \dots, V_k\}$ be a set of variables, $X \subseteq \prod_{i=1}^{k} V_i$ be a set of possible records, $U \subseteq V$, and $\varepsilon \geq 0$. Let Q denote the set of all datasets over X and V. A randomised algorithm $\mathcal{M} : Q \to R$ on is said to be *ε-private on U* (or *satisfy ε-privacy on U*) if for any two datasets D_1, D_2 such that there exists dataset D and $a_1, b_1 \in V_1, \dots, a_k, b_k \in V_k$ satisfying

$$D_1 = D \cup \{(a_1, , a_2, \dots, a_k)\}, D_2 = D \cup \{(b_1, , b_2, \dots, b_k)\}, \tag{1}$$

$$a_i = b_i \text{ for all } i \text{ such that } V_i \notin U, \tag{2}$$

we have for any $S \in R$,

$$P(\mathcal{M}(D_1) \in S) \leq e^{\varepsilon} P(\mathcal{M}(D_2) \in S).$$

We explore the properties of ε-privacy on U in the next section.

3 Properties and relation to differential privacy

The properties below follow easily from the definition of ε-privacy on U:

Proposition 3.1. *Let V be a set of variables. Then ε-privacy on V is equivalent to ε-indistinguishability [Dwork et al., 2006b], also known as bounded ε-differential privacy [Kifer and Machanavajjhala, 2011].*

Proposition 3.2. *Let V be a set of variables and $U_1 \subseteq U_2 \subseteq V$. If a randomised algorithm $\mathcal{M}$ is ε-private on U_2, then it is ε-private on U_1.*

Proposition 3.3. *Let $V = \{V_1, V_2, \dots, V_k\}$ be a set of variables, $X \subseteq \prod_{i=1}^{k} V_i$ be a set of possible records and $U \subseteq V$. Let $f : X \to \prod_{V_i \in V \setminus U} V_i$ be the function defined by*

$$f(a_1, a_2, \dots, a_k) = (a_{i_1}, a_{i_1}, \dots, a_{i_l}),$$

where $i_1 < i_2 < \cdots < i_l$ are all the elements of $\{1, 2, \dots, k\}$ such that $V_{i_1}, V_{i_2}, \dots, V_{i_l} \notin U$.

Let Q denote the set of all datasets over X and V. Then a randomised algorithm $\mathcal{M} : Q \to R$ is 0-private on U if and only if for any two datasets D_1, D_2 such that (taking in account multiplicities)

$$f(D_1) = f(D_2),$$

we have for any $S \in R$,

$$P(\mathcal{M}(D_1) \in S) = P(\mathcal{M}(D_2) \in S).$$

Remark 3.3.1. We can think of a randomised algorithm that is 0-private on U as one that is independent of the record values for the dataset variables in U. The function f can be thought of as removing the variables in U from the dataset and is itself a 0-private algorithm on U. Thus, the removal of direct identifiers from a dataset during the process of de-identification is 0-private on the set of direct identifiers.

Proposition 3.4 (Composition)**.** *Let V be a set of variables, $U \subseteq V$ and $\mathcal{M}_1$ and $\mathcal{M}_2$ be independent randomised algorithms such that $\mathcal{M}_1$ is ε_1-private on U and $\mathcal{M}_2$ is ε_2-private on U. Then the randomised algorithm $\mathcal{M}_{1,2}$ defined by $\mathcal{M}_{1,2}(D) = (\mathcal{M}_1(D), \mathcal{M}_2(D))$ is $(\varepsilon_1 + \varepsilon_2)$-private on U.*

For the rest of this paper, differential privacy refers to the bounded variant of differential privacy. Proposition 3.1 shows that ε-privacy on U is a generalisation of ε-differential privacy, with equivalence when $U = V$. Combining propositions 3.1 and 3.2, we get the following result that shows that ε-privacy on U is a relaxation of ε-differential privacy:

Proposition 3.5. *Let V be a set of variables and $U \subseteq V$. If a randomised algorithm $\mathcal{M}$ is ε-differentially private, then it is ε-private on U.*

The properties above can be used to assign an epsilon value to each subset of dataset variables of a dataset protected using differential privacy. We illustrate with an example below.

Example 3.6. Suppose we release the following three count tables, each using an independent differentially private algorithm with $\varepsilon = 1$:

- Table A: Population by age and sex,
- Table B: Population by residential region and race, and
- Table C: Population by residential region and age.

Further assume that for each table, the differentially private algorithm used to generate the table is independent of variables not used in the table. For example, for Table A, we assume that the randomised algorithm used to generate it only depends on the age and sex values of the records in the dataset and is independent of the record values for all other variables. This would be the case if the table was generated using algorithms such as the geometric mechanism [Ghosh et al., 2009].

Then using Propositions 3.3, 3.4 and 3.5, we can assign an epsilon value to each dataset variable (or more accurately, each singleton subset of the set of dataset variables). For example, for the dataset variable age, we have by Proposition 3.5 that the algorithms used to generate Table A and Table C satisfy 1-privacy on {age}. By Proposition 3.3, we have that the algorithm used to generate Table B satisfies 0-privacy on {age}. Combining the above using the composition property, we have that the generation of all three tables satisfies 2-privacy on {age}.

A summary of the epsilon values for each variable and each release is provided in the table below:

Table 1: ε values for each release for privacy on each singleton subset of dataset variables

Release	{age}	{sex}	{residential region}	{race}	{other variable}
Table A	1	1	0	0	0
Table B	0	0	1	1	0
Table C	1	0	1	0	0
Tables A, B & C	2	1	2	1	0

Analyses of the epsilon value for subsets of a dataset's variables such as in the example above can be useful to data providers who wish to quantify the privacy risks for specific variables of their datasets.

4 Relation to Pufferfish privacy

In this section, we state the privacy definition under the Pufferfish framework that is equivalent to ε-privacy on a subset of variables.

To set up the definition using the Pufferfish framework, we first introduce the concept of *identified datasets*.

Definition 4.1 (identified dataset)**.** Let $\mathcal{H}$ be a set and D be a dataset. A set $I \subseteq \mathcal{H} \times D$ is said to be an *identified dataset of D* if it satisfies the following:

1. If $(h_1, d_1), (h_2, d_2)$ are distinct elements of I, then $h_1 \neq h_2$.
2. For each $d \in D$, $|\{h \in \mathcal{H} | (h, d) \in I\}| = m_D(d)$, the multiplicity of d in multiset D.

$\mathcal{H}$ is called the *population set*. If I is an identified dataset of D, we say that D is the *unidentified dataset* of I.

Remark 4.1.1. I can be thought of as a function that assigns each individual in the sample to a record in the dataset.

To set up the privacy definition under the Pufferfish framework where the dataset is a random variable, we let the sample space be the set of all possible identified datasets and denote it by $\mathcal{I}$. The dataset is then a random variable that maps each identified dataset to its unidentified dataset.

Proposition 4.2. *Let $V = \{V_1, V_2, \ldots, V_k\}$ be a set of variables, $X \subseteq \prod_{i=1}^{k} V_i$ be a set of possible records, $\mathcal{H}$ be a population set, $\mathcal{I}$ be a set of all possible datasets and $U \subseteq V$. Let*

$$\sigma(h, v_1, v_2, \ldots, v_k) = \{I \in \mathcal{I} | (h, v_1, v_2, \ldots, v_k) \in I\}, \tag{3}$$

$$\mathbb{S} = \{\sigma(h, v_1, v_2, \ldots, v_k) | h \in \mathcal{H}, v_i \in V_i \text{ for } i = 1, 2, \ldots, k, (v_1, v_2, \ldots, v_k) \in X\}, \tag{4}$$

$$\mathbb{S}_{pairs} = \{(\sigma(h, v_1, v_2, \ldots, v_k), \sigma(h, v_1', v_2', \ldots, v_k')) \in \mathbb{S} \times \mathbb{S} | v_j = v_j' \text{ for all } V_j \notin U\}, \tag{5}$$

and $\mathbb{D}$ be the set of all probability measures θ such that for any collection $\{\sigma(h_1, d_1), \sigma(h_2, d_2), \ldots, \sigma(h_n, d_n)\}$ of events such that $h_1, h_2, \ldots, h_n$ are distinct, we have

$$P_\theta \left(\bigcap_{i=1}^{n} \sigma(h_i, d_i) \right) = \prod_{i=1}^{n} P_\theta(\sigma(h_i, d_i)). \tag{6}$$

Then ε-PufferFish$(\mathbb{S}, \mathbb{S}_{\text{pairs}}, \mathbb{D})$ privacy is equivalent to ε-privacy on U.

The above theorem can be proved using a similar approach as the proof that differential privacy is a specific case of PufferFish privacy provided by Kifer and Machanavajjhala [2014].

Remark 4.2.1. Since ε-privacy on a subset of dataset variables is a privacy definition under the Pufferfish framework, by Theorem 5.1 of Kifer and Machanavajjhala [2014], it follows that ε-privacy on a subset of dataset variables satisfies tranformation invariance and convexity.

5 Algorithms

5.1 Motivating example

Suppose a university conducts a survey on a class of its graduates, and one of the questions in the survey asks respondents to indicate which income band they belong to. After conducting the survey, the university is interested to release the count of graduates by income band and degree programme:

Degree Programme	Total	Annual Income Band							
		<$30,000	$30,000 - $39,999	$40,000 - $49,999	$50,000 - $59,999	$60,000 - $79,999	$80,000 - $99,999	$100,000 - $119,999	$120,000 and above
Bachelor of Arts	231	5	10	47	68	54	33	12	2
Bachelor of Science	392	1	13	44	92	131	75	21	15
Bachelor of Engineering	777	2	25	78	218	237	153	46	18

Table 2: Possibly disclosive accurate counts by degree programme and annual income band

However, the university is concerned that releasing the accurate counts might lead to disclosure of the respondents' income and would like to use a formal privacy model to protect the data. Furthermore, the university does not consider the degree programme of a graduate as private information: the university had previously released the count of as well as name list of graduates for each degree programme for that graduating class. Thus, the university is interested to release the counts such that the total count for each degree programme (highlighted in Table 2 above) is accurate. For example, the university would like to release a table such as the one below:

Degree Programme	Total	Annual Income Band							
		<$30,000	$30,000 - $39,999	$40,000 - $49,999	$50,000 - $59,999	$60,000 - $79,999	$80,000 - $99,999	$100,000 - $119,999	$120,000 and above
Bachelor of Arts	231	4	12	47	66	55	37	5	5
Bachelor of Science	392	5	12	44	91	133	73	20	14
Bachelor of Engineering	777	4	20	74	218	238	155	52	16

Table 3: Noisy counts by degree programme and annual income band such that the total counts by degree programme are accurate

In the next subsection, we show how the privacy definition introduced in this paper can be used to protect data in such a situation, as well as an algorithm for generating privacy-protected count tables that are consistent with accurate counts such as Table 3.

5.2 Algorithm

Consider a data provider with a dataset D over X and $V = \{U_1, U_2, \ldots, U_k, V_1, V_2, \ldots, V_l\}$, where U_1 and V_1 are finite sets representing categorical variables and $X \subseteq U_1 \times \cdots \times U_k \times V_1 \times \cdots \times V_l$. Suppose that the data provider wants to release the count table of counts by U_1 and V_1 using a algorithm that is ε-private on $U = \{U_1, U_2, \ldots, U_k\}$. Since the accurate counts by V_1 can be generated without using the record values for the variables in U, by Proposition 3.3, these can be released accurately while still satisfying 0-privacy on U. The data provider is interested in releasing integer counts by U_1 and V_1 such that summing the counts over U_1 will give the accurate counts by V_1. We provide an algorithm for releasing such count tables below.

Let $V_1 = \{v_1, v_2, \ldots, v_m\}$. For each $i \in \{1, 2, \ldots, m\}$, let $U_{1,i} = \{u_{i,1}, u_{i,2}, \ldots, u_{i,n_i}\} = \{u \in U_1 | \exists (a_1, \ldots, a_k, b_1, \ldots, b_l) \in X \text{ s.t. } a_1 = u, b_1 = v_i\}$. Denote the count of records $(a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_l) \in D$ such that $b_1 = v_i, a_1 = u_{i,j}$ by $f(i, j)$.

For any integer $N \geq 2$, define $M(N)$ to be the set of $\binom{N}{2} \times N$ matrices satisfying the following:

1. Every row contain one entry with value 1, one entry with value -1, and $N - 2$ entries with value 0, and
2. No two rows of the matrix have their nonzero entries in the same pair of columns.

With the above definitions, we define the Algorithm 1 and summarise the key properties of Algorithm 1 in Proposition 5.1.

Algorithm 1

Require: $\varepsilon > 0$, f, m, $n_1, n_2, \ldots, n_m$
  for $i = 1, 2, \ldots, m$ **do**
    $[g(i,1), g(i,2), \ldots, g(i,n_i)] \leftarrow [f(i,1), f(i,2), \ldots, f(i,n_i)]$
    if $n_i > 2$ **then**
      $M \leftarrow$ a matrix belonging to $M(n_i)$
      $[b_1, b_2, \ldots, b_{\binom{n_i}{2}}] \leftarrow \binom{n_i}{2}$ independent samples from the two-sided geometric distribution $P(Z = z) = \frac{1-e^{\varepsilon/2}}{1+e^{\varepsilon/2}} e^{-\varepsilon|z|/2}$
      $[g(i,1), g(i,2), \ldots, g(i,n_i)] \leftarrow [g(i,1), g(i,2), \ldots, g(i,n_i)] + [b_1, b_2, \ldots, b_{\binom{n_i}{2}}]M$
    end if
  end for
  return g

Proposition 5.1. *Algorithm 1 satisfies the following:*

1. *ε-privacy on U,*

2. *for all* $i \in \{1, 2, \ldots, m\}$, $\sum_{j=1}^{n_i} g(i,j) = \sum_{j=1}^{n_i} f(i,j)$,
3. *for all* i, j, $g(i,j) \in \mathbb{Z}$, *and*
4. *for all* i, j, $E(g(i,j)) = f(i,j)$.

Note that the above algorithm does not ensure nonnegativity. Further optimisation would be required if the output contains negative values and the data provider does not wish to release negative count estimates.

6 Conclusion

In this paper we introduce a class of privacy definitions that protects a subset of variables in a dataset. We show its relation to DP and Pufferfish privacy, and how it can be used to assign a privacy risk ε to each variable of a dataset. We also provide an algorithm for generating count tables that satisfy the proposed definition while ensuring consistency with the counts that can be released accurately. These results can be useful to data providers that want to release some statistics accurately while still satisfying a mathematical definition of privacy, or wish to understand the privacy risk for different variables in their dataset.

References

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006a.

Shiva P Kasiviswanathan and Adam Smith. On the'semantics' of differential privacy: A bayesian formulation. *Journal of Privacy and Confidentiality*, 6(1), 2014.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006b.

Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 193–204, 2011.

Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 351–360, 2009.

Daniel Kifer and Ashwin Machanavajjhala. Pufferfish: A framework for mathematical privacy definitions. *ACM Trans. Database Syst.*, 39(1), jan 2014. ISSN 0362-5915. doi:10.1145/2514689. URL `https://doi.org/10.1145/2514689`.