Objective: Privacy-preserving data publishing addresses the problem of disclosing sensitive data when mining for useful information. Among existing privacy models, $\varepsilon$-differential privacy provides one of the strongest privacy guarantees and makes no assumptions about an adversary's background knowledge. All the existing solutions that ensure $\varepsilon$-differential privacy handle the problem of disclosing relational and set-valued data in a privacy-preserving manner separately. In this paper, we propose an algorithm that considers both relational and set-valued data in differentially private disclosure of healthcare data.

Material and methods: The proposed approach makes a simple yet fundamental switch in differentially private algorithm design: instead of listing all the possible records (i.e., contingency table) for noise addition, records are generalized before noise addition. The algorithm first generalizes the raw data in a probabilistic way, and then adds noise to guarantee $\varepsilon$-differential privacy.

Results: We showed that the disclosed data could be used effectively to build a decision tree induction classifier. Experimental results demonstrated that the proposed algorithm is scalable and performs better than the existing solutions for classification analysis.

Limitation: The resulting utility may degrade when the output domain size is very large, making it potentially inappropriate to generate synthetic data for large health databases.

Conclusion: Unlike existing techniques, the proposed algorithm allows the disclosure of health data containing both relational and set-valued data in a differentially private manner, and can retain essential information for discriminative analysis.
1 Introduction

With the wide deployment of Electronic Health Record (EHR) systems, health data have been collected at an unprecedented rate. The need for sharing health data among multiple parties has become evident in several applications\(^1\), such as decision support, policy development, and data mining. Meanwhile, major concerns have been raised about individual privacy in health data sharing. The current practice of privacy protection primarily relies on policies and guidelines, for example, the Health Insurance Portability and Accountability Act (HIPAA)\(^2\) in the United States. HIPAA defines two approaches to achieve de-identification: the first is Expert Determination, which requires that an expert certify that the re-identification risk inherent in the data is sufficiently low; the second is Safe Harbor, which requires the removal and suppression of a list of attributes\(^3\). Safe Harbor requires data disclosers to follow a checklist\(^4\) to remove specific information to de-identify the records.

However, there are numerous controversies on both sides of the privacy debate regarding these HIPAA privacy rules\(^5\). Some think that protections provided in the de-identified data are not enough\(^6\). Others contend that these privacy safeguards hamper biomedical research, and that observing them may preclude meaningful studies of medical data that depend on suppressed attributes, e.g., fine-grained epidemiology studies in areas with fewer than 20,000 residents or geriatric studies requiring detailed ages over 89\(^3\). There are concerns that privacy rules will erode the efficiencies that computerized health records may create, and in some cases, interfere with law enforcement\(^5\). Recently, the Institute of Medicine (IOM) Committee on Health Research and the Privacy of Health Information concluded that the privacy rules do not adequately safeguard privacy and also significantly impede high-quality research\(^7\). The result is that
patients’ health records are not well protected at the same time that researchers cannot effectively use them for discoveries. Technical efforts are highly encouraged to make published health data both privacy-preserving and useful.

Anonymizing health data is a challenging task due to inherent heterogeneity. Modern health data are typically composed of different types, for example relational data (e.g., demographics) and set-valued data (e.g., diagnostic codes and laboratory tests). In relational data (e.g., gender, age, BMI), records contain only one value for each attribute. On the other hand, set-valued data (e.g., diagnostic codes and lab tests) contain one or more values (cells) for each attribute. For example, the attribute-value \{1^*, 2^*\} of the diagnostic-code contains two separate cells: \{1^*\} and \{2^*\}. For many medical problems, different types of data need to be published simultaneously so that the correlation between different data types can be preserved. Such an emerging heterogeneous data-publishing scenario, however, is seldom addressed in the existing literature related to privacy technology. Current techniques primarily focus on a single type of data and therefore are unable to thwart privacy attacks caused by inferences involving different data types. In this article, we propose an algorithm to publish heterogeneous health data that can retain the essential information for supporting data mining tasks in a differentially private manner. The following real-life scenario further illustrates the privacy threats due to heterogeneous health data sharing.

**Example 1** Consider the raw patient data in Table 1 (the attribute ID is just for the purposes of illustration). Each row in the table represents the information from a patient.
### Table 1: Raw patient data

<table>
<thead>
<tr>
<th>ID</th>
<th>Sex</th>
<th>Age</th>
<th>Diagnostic Code</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Male</td>
<td>34</td>
<td>11, 12, 21, 22</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>Female</td>
<td>65</td>
<td>12, 22</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>Male</td>
<td>38</td>
<td>12</td>
<td>N</td>
</tr>
<tr>
<td>4</td>
<td>Female</td>
<td>33</td>
<td>11, 12</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>Female</td>
<td>18</td>
<td>12</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>Male</td>
<td>37</td>
<td>11</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>Male</td>
<td>32</td>
<td>11, 12, 21, 22</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>Female</td>
<td>25</td>
<td>12, 21, 22</td>
<td>N</td>
</tr>
</tbody>
</table>

### Table 2: Differentially private disclosed data ($\epsilon = 1$, $h = 2$).

<table>
<thead>
<tr>
<th>Sex</th>
<th>Age</th>
<th>Diagnostic Code</th>
<th>Class</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>[18-65)</td>
<td>1*</td>
<td>Y</td>
<td>3</td>
</tr>
<tr>
<td>Male</td>
<td>[18-65)</td>
<td>1*</td>
<td>N</td>
<td>2</td>
</tr>
<tr>
<td>Female</td>
<td>[18-65)</td>
<td>1*</td>
<td>Y</td>
<td>1</td>
</tr>
<tr>
<td>Female</td>
<td>[18-65)</td>
<td>1*</td>
<td>N</td>
<td>3</td>
</tr>
<tr>
<td>Male</td>
<td>[18-65)</td>
<td>1*, 2*</td>
<td>Y</td>
<td>2</td>
</tr>
<tr>
<td>Male</td>
<td>[18-65)</td>
<td>1*, 2*</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>Female</td>
<td>[18-65)</td>
<td>1*, 2*</td>
<td>Y</td>
<td>2</td>
</tr>
<tr>
<td>Female</td>
<td>[18-65)</td>
<td>1*, 2*</td>
<td>N</td>
<td>4</td>
</tr>
</tbody>
</table>

### Figure 1: Taxonomy tree of attributes.

Attributes Sex, Age, and DiagnosticCode are categorical, numerical, and set-valued, respectively. Suppose that the data owner needs to release Table 1 for the purpose of classification analysis on the class attribute, which has two values, Y and N, indicating whether or not the patient is deceased. If a record in the table is too specific such that not many patients can match it, releasing the data may lead to the re-identification of a patient. For example, Loukides et al.\textsuperscript{11} demonstrated that for the International Classification of Disease (ICD) version 9 codes (or “diagnostic codes” for brevity), one source of set-valued data could be used by an adversary for linkage to patients’ identities. Needless to say, the knowledge of both relational and set-valued data about a victim
makes the privacy attack easier for an adversary. Suppose that the adversary knows that
the target patient is female and her diagnostic codes contain \{11\}. Then, record #4 can
be uniquely identified, since she is the only \textit{Female} with diagnostic codes \{11,12\} in the
raw data. Thus, identifying her record results in disclosure that she also has \{12\}. Note
that, we do not make any assumption about the adversary’s background knowledge. An
adversary may have partial or full information about the set-valued data and she can try
to use any background knowledge to identify the victim. A differentially-private
mechanism ensures that the probability of generating any output (released data) is almost
equally likely from all nearly identical input data sets, and therefore guarantees that
outputs are insensitive to any single individual's record. In other words, an individual's
privacy is not at risk because of inclusion in the disclosed data set.

To prevent such linking attacks, a number of partition-based privacy models have been
proposed [12, 13, 14, 15, 16]. However, recent research has indicated that these models
are vulnerable to various privacy attacks [18, 23, 24, 25] and provide insufficient privacy
protection. In this article, we employ \textit{differential privacy}\textsuperscript{26}, a privacy model that provides
provable privacy guarantees and that is, by definition, immune against all aforementioned
attacks. Differential privacy makes no assumption about an adversary's background
knowledge. A differentially-private mechanism ensures that the probability of any output
(released data) is almost equally likely from all nearly identical input data sets and thus
guarantees that all outputs are insensitive to any single individual's data. In other words,
an individual's privacy is not at risk because of inclusion in the disclosed data set.

1.1 Motivation

Existing algorithms that provide differential privacy guarantee are based on two
approaches: \textit{interactive} and \textit{non-interactive}. In an interactive framework, a data miner
can pose aggregate queries through a private mechanism, and a database owner answers these queries in response. Most of the proposed methods for ensuring differential privacy are based on an interactive framework [29,36,48,49,50]. In a non-interactive framework the database owner first anonymizes the raw data and then releases the anonymized version for public use. In this article, we adopt the non-interactive framework as it has a number of advantages for data mining [10]. Current techniques that adopt the non-interactive approach publish contingency table or marginals of the raw data [21,29,46,47]. The general structure of these approaches is to first derive a frequency matrix of the raw data over the database domain.

**Table 3:** Sample data table

<table>
<thead>
<tr>
<th>Job</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineer</td>
<td>34</td>
</tr>
<tr>
<td>Lawyer</td>
<td>50</td>
</tr>
<tr>
<td>Engineer</td>
<td>38</td>
</tr>
<tr>
<td>Lawyer</td>
<td>33</td>
</tr>
<tr>
<td>Dancer</td>
<td>20</td>
</tr>
<tr>
<td>Writer</td>
<td>37</td>
</tr>
<tr>
<td>Writer</td>
<td>32</td>
</tr>
<tr>
<td>Dancer</td>
<td>25</td>
</tr>
</tbody>
</table>

**Table 4:** Contingency table

<table>
<thead>
<tr>
<th>Job</th>
<th>Age</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineer</td>
<td>[18-40]</td>
<td>2</td>
</tr>
<tr>
<td>Engineer</td>
<td>[40-65]</td>
<td>0</td>
</tr>
<tr>
<td>Lawyer</td>
<td>[18-40]</td>
<td>1</td>
</tr>
<tr>
<td>Lawyer</td>
<td>[40-65]</td>
<td>1</td>
</tr>
<tr>
<td>Dancer</td>
<td>[18-40]</td>
<td>2</td>
</tr>
<tr>
<td>Dancer</td>
<td>[40-65]</td>
<td>0</td>
</tr>
<tr>
<td>Writer</td>
<td>[18-40]</td>
<td>2</td>
</tr>
<tr>
<td>Writer</td>
<td>[40-65]</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 5:** Generalized contingency table

<table>
<thead>
<tr>
<th>Job</th>
<th>Age</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professional</td>
<td>[18-40]</td>
<td>3</td>
</tr>
<tr>
<td>Professional</td>
<td>[40-65]</td>
<td>1</td>
</tr>
<tr>
<td>Artist</td>
<td>[18-40]</td>
<td>4</td>
</tr>
<tr>
<td>Artist</td>
<td>[40-65]</td>
<td>0</td>
</tr>
</tbody>
</table>
For example, Table 4 shows the contingency table of Table 3. After that, noise is added to each count to satisfy the privacy requirement. Finally, the noisy frequency matrix is published. However, this approach is not suitable for high-dimensional data with a large domain because when the added noise is relatively large compared to the count, the utility of the data is significantly destroyed. We also confirm this point in our experimental results (Section 4). Our proposed solution instead first probabilistically generates a generalized contingency table and then adds noise to the counts. For example, Table 5 is a generalized contingency table of Table 3. Thus the count of each partition is typically much larger than the added noise.

1.2 Contributions
We propose a novel technique for publishing heterogeneous health data that provides an $\varepsilon$-differential privacy guarantee. While protecting privacy is a critical element in data publishing, it is equally important to preserve the utility of the published data, since this is the primary reason for data release. Taking the decision tree induction classifier as an example, we show that our sanitization algorithm can be effectively tailored for preserving information in data mining tasks. The contributions of this article are:

1. To our knowledge, a differentially private data disclosure algorithm that simultaneously handles both relational and set-valued data has not been previously developed. The proposed differentially private data algorithm is based on a generalization technique and preserves information for classification analysis. Previous work\textsuperscript{27} suggests that deterministic generalization techniques cannot be used to achieve $\varepsilon$-differential privacy, as they depend heavily on the data to be disseminated. Yet, we show that differentially private data can be released through the addition of uncertainty in the generalization procedure.
2. The proposed algorithm can also handle numerical attributes. Unlike existing methods, it does not require the numerical attributes to be pre-discretized. The algorithm adaptively determines the split points for numerical attributes and partitions the data based on the workload, while guaranteeing $\varepsilon$-differential privacy. This is an essential requirement for getting accurate classification, as we show in Section 5. Moreover, the algorithm is computationally efficient.

3. It is well acknowledged that $\varepsilon$-differential privacy provides a strong privacy guarantee. However, the utility of data disclosed by differentially private algorithms has received much less study. Does an interactive approach offer better data mining results than a non-interactive approach? Do differentially private data disclosure provides less utility than disclosure based on $k$-anonymous data? Experimental results demonstrate that our algorithm outperforms the recently proposed differentially-private interactive algorithm for building a classifier and the top-down specialization (TDS) approach that publishes $k$-anonymous data for classification analysis.

This article is organized as follows. Section 2 provides an overview of the generalization technique and presents the problem statement. Our anonymization algorithm is explained in Section 3. In Section 4, we experimentally evaluate the performance of our solution, and we summarize our main findings in Section 5.

2 Preliminaries

In this section, we introduce the notion of generalization in the context of data publishing, followed by a problem statement.
2.1 Generalization

Let \( D = \{r_1, ..., r_n\} \) be a multiset of records, where each record \( r_i \) represents the information of an individual with \( d \) attributes \( \mathcal{A} = \{A_1, ..., A_d\} \). We represent the data set \( D \) in a tabular form and use the terms “data set” and “data table” interchangeably. We assume that each attribute \( A_i \) has a finite domain, denoted by \( \Omega(A_i) \). The domain of \( D \) is defined as \( \Omega(D) = \Omega(A_1) \times \ldots \times \Omega(A_d) \). To generalize a data set \( D \), we replace the value of an attribute with a more general value. The exact general value is determined according to the attribute partition.

**Definition 2.1 (Partition)** The partitions \( P(A_i) \) of a numerical attribute are the intervals \( \langle l_1, l_2, ..., l_k \rangle \) in \( \Omega(A_i) \) such that \( \bigcup_{j=1}^{k} l_j = \Omega(A_i) \). For categorical and set-valued attributes, partitions are defined by a set of nodes from the taxonomy tree such that it covers the whole tree, and each leaf node belongs to exactly one partition.

For example, \( \text{Any Sex} \) is the general value of \( \text{Female} \) according to the taxonomy tree of \( \text{Sex} \) in Figure 1. Similarly, \( \text{age 23} \) and \( \text{diagnostic code 11} \) can be represented by the interval \( [18 - 40] \) and the code \( 1^* \), respectively. For numerical attributes, these intervals are determined adaptively from the entire data set.

**Definition 2.2 (Generalization)** Generalization is defined by a function \( \Phi = \{\phi_1, \phi_2, ..., \phi_d\} \), where \( \phi_i: v \mapsto p \) maps each value \( v \in \Omega(A_i) \) to a \( p \in P(A_i) \).

Clearly, given a data set \( D \) over a set of attributes \( \mathcal{A} = \{A_1, ..., A_d\} \), many alternative generalization functions are feasible. Each generalization function partitions the attribute domains differently. To satisfy \( \varepsilon \)-differential privacy, the algorithm must determine a
generalization function that is insensitive to the underlying data. More formally, for any two data sets $D$ and $D'$, where $|D \Delta D'| = 1$ (i.e., they differ on at most one record), the algorithm must ensure that the ratio of $Pr[Ag(D) = \Phi]$ and $Pr[Ag(D') = \Phi]$ is bounded, where $Ag(\cdot)$ is a randomized algorithm (see Appendix A1). One naive solution satisfying $\varepsilon$-differential privacy is to have a fixed generalization function, irrespective of the input data set (i.e., by definition zero-differential private but useless). However, a proper choice of a generalization function is very crucial since the data mining result varies significantly for different choices of partitioning. In Section 4, we present an efficient algorithm for determining an adaptive partitioning technique for classification analysis while guaranteeing $\varepsilon$-differential privacy. Appendix A1 presents an overview of $\varepsilon$-differential privacy and the core mechanisms to achieve $\varepsilon$-differential privacy.

### 2.2 Problem Statement

Suppose a data owner wants to release a de-identified data table $D'(A_{1^{pr}}, ..., A_{d^{pr}}, A^{cls})$, where the symbols $A_{\cdot^{pr}}$ and $A^{cls}$ correspond to predictor attributes and the class attribute, respectively, for release to the public for classification analysis. The attributes in $D$ are classified into three categories: (1) An identifier $A_{i}$ attribute that explicitly identifies an individual, such as SSN, and Name. These attributes are removed before releasing the data as per the HIPAA Privacy Rule. (2) A class attribute $A^{cls}$ that contains the class value; the goal of the data miner is to build a classifier to accurately predict the value of this attribute. (3) A set of $d$ predictor attributes $\mathcal{A}^{pr} = \{A_{1^{pr}}, ..., A_{d^{pr}}\}$, whose values are used to predict the binary label of the class attribute.

We require the class attribute to be categorical, and the predictor attribute can be
categorical, numerical or set-valued. Further, we assume that for each categorical or
set-valued attribute $A_i^{pr}$, a taxonomy tree is provided. The taxonomy tree of an
attribute $A_i^{pr}$ specifies the hierarchy among the values. Our problem statement can
be written as: given a data table $D$ and the privacy parameter $\varepsilon$, our objective is to
generate a de-identified data table $D^*$ such that $D^*$ (1) satisfies $\varepsilon$-differential privacy,
and (2) preserves as much information as possible for classification analysis.

3 The Algorithm

In this section, we present an overview of our Differentially-private algorithm based on
Generalization ($\text{DiffGen}$). We elaborate the key steps, and prove that the algorithm is
$\varepsilon$-differentially private in Appendix A2. In addition, we present the implementation details
and analyze the complexity of the algorithm in Appendix A3.

\textbf{Algorithms 1 DiffGen}

\textbf{Input:} Raw data set $D$, privacy budget $\varepsilon$, and number of specializations $h$
\textbf{Output:} Generalized data set $D^*$

1: Initialize every value in $D$ to the topmost value (see Figure 2 for details);
2: Initialize $\text{Cut}_i$ to include the topmost value (see Figure 2 for details);
3: Set a privacy budget for specification of predictors $\varepsilon' \leftarrow \varepsilon/(2(|A_n^{pr}| + 2h))$;
4: Determine the split value for each $v_n \in \text{Cut}_i$ with probability $\propto \exp(\varepsilon'/2\Delta u(D,v_n))$;
5: Compute the score for each candidate $v \in \text{Cut}_i$ (see Appendix A2 for details);
6: \textbf{for} $i = 1$ to $h$ \textbf{do}
7: \hspace{1em} Select $v \in \text{Cut}_i$ with probability $\propto \exp(\varepsilon'/2\Delta u(D,v))$;
8: \hspace{1em} Specialize $v$ on $D$ and update $\cup \text{Cut}_i$;
9: \hspace{1em} Determine the split value for each new $v_n \in \text{Cut}_i$ with probability $\propto \exp(\varepsilon'/2\Delta u(D,v_n))$;
10: \hspace{1em} Update score for $v \in \text{Cut}_i$;
11: \textbf{end for}
12: \textbf{return} each group with count $(C + \text{Lap}(2/\varepsilon))$, where $\text{Lap}(\cdot)$ denote the
probability density function of Laplacian distribution.
Algorithm 1 first generalizes the predictor attributes $\mathcal{A}^{pr}$ and divides the raw data into several equivalence groups, where all the records within a group have the same attribute values. Then, the algorithm publishes the noisy counts of the groups. The general idea is to sanitize the raw data by a sequence of specializations, starting from the topmost general state as shown in Figure 2. A specialization, written as $v \rightarrow \text{child}(v)$, where $\text{child}(v)$ denotes the set of child values of $v$, replaces the parent value $v$ with a child value. The specialization process can be viewed as pushing the “cut” of each taxonomy tree downwards. A cut of the taxonomy tree for an attribute $A_i^{pr}$, denoted by $\text{Cut}_i$, contains exactly one value on each root-to-leaf path. The value of the set-valued attribute of a record can be generalized to a cut if every item in the record can be generalized to a node in the cut and every node in the cut generalizes some items in the record. For example, the value $\{21,22\}$ can be generalized to the hierarchy cuts $\{2*\}$ and $\{**\}$, but not $\{1*,2*\}$. Figure 2 shows a solution cut indicated by the dashed curve representing Table 2, which corresponds to de-identified data to be disseminated.

![Figure 2: Tree for partitioning records. A randomized mechanism was deployed to specializing predictors in a top-down manner (using half of the privacy budget). At leave nodes, random noise is added to the count of elements using the second half of the privacy budget to ensure overall $\epsilon$-Differentially-private outputs.](image)

Initially, $\text{DiffGen}$ creates a single partition by generalizing all values in $\mathcal{A}^{pr}$ to the topmost value in their taxonomy trees (Line 1). The $\text{Cut}_i$ contains the topmost value for
each attribute $A_i \wedge pr$ (Line 2). The specialization starts from the topmost cut and pushes down the cut iteratively by specializing some value in the current cut. At each iteration, DiffGen uses an exponential mechanism \(^{31}\) (see Appendix A1) to select a candidate $v \in \cup Cut_i$ for specialization (Line 7). Candidates are selected based on their score values (see Appendix A2), and different heuristics (e.g., information gain and max frequency) can be used to determine the score of the candidates. Then, the algorithm specializes $v$ and updates $\cup Cut_i$ (Line 8). As taxonomy trees for the numerical attributes are not given, DiffGen again uses the exponential mechanism to determine the split value dynamically for each numerical candidate $v \wedge n \in \cup Cut_i$ (Lines 4 and 9). DiffGen specializes $v$ by recursively distributing the records from the parent partition into disjoint child partitions with more specific values based on the taxonomy tree. For set-valued attributes, the algorithm computes the noisy count of each child partition to determine whether it is empty or not. Only "non-empty" partitions are considered for further split in the next iteration. We provide further details for candidate selection and the split value determination steps in Appendix A2. DiffGen also calculates the score for each new candidate due to the specialization (Line 10). The algorithm terminates after a given number of specializations. Finally, the algorithm adds Laplace noise (see Appendix A1) to each equivalence group of the leaf-partition to construct the sanitized data table $D^\ast$. We use the following example to facilitate the understanding of how to use score functions, which are based on heuristics (e.g., information gain and max frequency), for specification.

**Example 2** Consider Table 1 with $\varepsilon = 1$ and $h = 2$. Initially, the algorithm creates one root partition containing all the records that are generalized to $\langle \text{Any Sex}, [18 - 65], \ast \rangle$. $\cup Cut_i$ includes $\{\text{Any Sex}, [18 - 65], \ast \}$. To find the first specialization among the
candidates in $\cup \text{Cut}_i$, we compute score of $(\text{Any Sex}, [18-65])$, and **. We show how to compute the information gain ($\text{InfoGain}$) and maximum frequency ($\text{Max}$) scores of $\text{Any Sex}$ for the specialization $\text{Any Sex} \rightarrow \{\text{Male,Female}\}$. Details of these two utility functions are discussed in Appendix 2.

(1) Information Gain:

\[
H_{(\text{Any Sex})} (\text{Table}) = -\frac{4}{8} \times \log_2 \frac{4}{8} - \frac{4}{8} \times \log_2 \frac{4}{8} = 1
\]

\[
H_{\text{Male}} (\text{Table}) = -\frac{2}{4} \times \log_2 \frac{2}{4} - \frac{2}{4} \times \log_2 \frac{2}{4} = 1
\]

\[
H_{\text{Female}} (\text{Table}) = -\frac{2}{4} \times \log_2 \frac{2}{4} - \frac{2}{4} \times \log_2 \frac{2}{4} = 1
\]

\[
\text{InfoGain} (\text{Table}, \text{Any Sex}) = 1 - (\frac{4}{8} \times 1 + \frac{4}{8} \times 1) = 0
\]

(2) Max: $\text{Max} (\text{Table}, \text{Any Sex}) = 2 + 2 = 4$.

Let the first specialization be $\ast \ast \rightarrow \{1 \ast, 2 \ast\}$. The algorithm then creates three child partitions with the child values $\{1 \ast\}$, $\{2 \ast\}$, and $\{1 \ast, 2 \ast\}$ respectively by replacing the node $\{\ast \ast\}$ with different combinations of its children, leading $r_{3}$, $r_{4}$, $r_{5}$, and $r_{6}$ to the child partition $\{1 \ast\}$ and $r_{7}$, $r_{8}$ and $r_{9}$ to the child partition $\{1 \ast, 2 \ast\}$. Suppose that the noisy counts indicate that these two child partitions are "non-empty". Further splits are needed for them. There is no need to explore the child partition $\{2 \ast\}$ any more, as it is considered "empty". $\cup \text{Cut}_i$ is updated to $\{\text{Any Sex}, [18 - 65], 1 \ast, 2 \ast\}$. Suppose that the next specialization is $\_\_\_\_\_\_\_\_ \rightarrow \{\text{Male,Female}\}$, which creates further specialized partitions. Finally, the algorithm outputs the equivalence groups of each leaf partition along with their noisy counts as shown in Figure 2 under the dotted line.

Please refer to Appendix A2 for privacy analysis of DiffGen and Appendix A3 about implementation details.
4 Experimental Description

In this section our objectives are to study the impact of enforcing differential privacy on the data quality in terms of classification accuracy, and to evaluate the scalability of the proposed algorithm for handling large data sets. We also compare DiffGen with DiffP-C4.5\(^{29}\), a differentially-private interactive algorithm for building a classifier, and with the top-down specialization (TDS) approach\(^{30}\) that publishes \(k\)-anonymous data for classification analysis. All experiments were conducted on an Intel Core i7 2.7GHz PC with 12GB RAM.

We employed two real-life data sets: MIMIC and Adult. We retrieved the MIMIC data set from Multi-parameter Intelligent Monitoring in Intensive Care II research database\(^{32}\), which contains over 36,000 intensive care unit (ICU) episodes. Specifically, we picked eight features (i.e., marital status, gender, ethnic, payment description, religion description, admission type, admission source, and ICD9 code) and a target variable (i.e., mortality). Among all eight features, the first seven are categorical attributes and the last one is a set-valued attribute. The publicly available Adult\(^{33}\) data set is the 1994 US census data set that has been widely used for testing many sanitization algorithms. Adult has 45,222 census records with 6 numerical attributes, 8 categorical attributes, and a binary class column representing two income levels, \(<50\text{K} \text{ or } >50\text{K}\. Please refer to\(^{30}\) for the description of attributes.

To evaluate the impact on classification quality, we divide the data into training and testing sets. First, we applied our algorithm to sanitize the training set and to determine the \(\cup \text{Cut}_i\). Then, the same \(\cup \text{Cut}_i\) is applied to the testing set to produce a generalized test set. Next, we built a classifier on the sanitized training set and measure
the classification accuracy \((CA)\) on the generalized records of the test set. For classification models, we used the well-known C4.5 classifier\(^{34}\). For each experiment, we executed 10 runs and averaged the results over the runs.

![Figure 3: Classification accuracy for MIMIC data set using DiffGen based on two scoring functions: information gain and maximum frequency.](image)

**MIMIC Dataset.** We applied \textit{DiffGen} to MIMIC data set for both utility functions (i.e., Max and InfoGain). Figure 3 shows the classification accuracy \(CA\) for Max and InfoGain, where the privacy budget \(\varepsilon = 0.1, 0.25, 0.5, 1\), and the number of specializations \(h = 5\). We used \(2/3\) of records to build the classifier and measured the accuracy on the remaining \(1/3\) of the records. Both utility functions have similar performance, where \(CA\) spans from 86\% to 89\% under different privacy budgets. The experimental result suggests that the proposed algorithm can achieve good classification accuracy on heterogeneous health data. We could not directly compare our method with others for the MIMIC data set because we are not aware of a method that can sanitize heterogeneous data while ensuring \(\varepsilon\)-differential privacy.

**Adult Dataset.** To better visualize the cost and benefit of our approach, we provide
additional measures: *Baseline Accuracy* ($BA$) is the classification accuracy measured on the raw data without sanitization. $BA - CA$ represents the cost in terms of classification quality for achieving a given $\varepsilon$-differential privacy requirement. On the other extreme, we measure *Lower bound Accuracy* ($LA$), which is the accuracy on the raw data with all attributes (except for the *class* attribute) removed. $CA - LA$ represents the benefit of our method over the naive non-disclosure approach.

![Figure 4: Classification accuracy for Adult data set](image)

Figure 4(a) depicts the classification accuracy $CA$ for the utility function Max, where the privacy budget $\varepsilon = 0.1, 0.25, 0.5, 1$, and the number of specializations $4 \leq h \leq 16$. The $BA$ and $LA$ are 85.3% and 75.5%, respectively, as shown in the figure by the dotted lines. For $\varepsilon = 1$ and $h = 10$, $BA - CA$ is around 3% and $CA - LA$ is 6.74%. For $\varepsilon = 0.5$, $BA - CA$ spans from 3.57% to 4.8%, and $CA - LA$ spans from 5% to 6.23%. However, when $\varepsilon$ decreases to 0.1, $CA$ quickly decreases to about 78% (highest point), the cost increases to about 7%, and the benefit decreases to about 3%. These results suggest that for an acceptable privacy budget such as 1, the cost for achieving $\varepsilon$-differential privacy is small, while the benefit of our method over the naive method is large. Figure 4(b) depicts the classification accuracy $CA$ for the utility function InfoGain. The performance of the InfoGain is not as good as Max because the difference between
the scores of a good and a bad attribute is much smaller for InfoGain as compared to Max. Therefore, the exponential mechanism does not work as effectively in the case of InfoGain as it does for Max.

![Graph showing averaged accuracy and scalability](image)

**Figure 5:** Comparison of DiffGen vs. DiffP-C4.5 and TDS algorithms. (a) Evaluation of averaged accuracy, where bottom and topmost lines stand for the worst case (i.e., all records generalized to one super record) and the optimal case (i.e., no record is generalized at all); (b) Evaluation in terms of reading, anonymization, and writing time of all three algorithms.

Figure 5(a) shows the classification accuracy (CA) of DiffGen, DiffP-C4.5, and TDS. For DiffGen, we use utility function Max and fix the number of specializations $h = 15$. DiffP-C4.5 also uses Adult data set and all the results of the DiffP-C4.5 are taken from their paper. For TDS we fixed the anonymity threshold $k = 5$ and conducted the experiment ourselves. Following the same setting, we executed 10 runs of 10-fold cross-validation to measure the CA.

The accuracy of DiffGen is clearly better than DiffP-C4.5 for privacy budgets $\varepsilon \leq 2$. Note that the privacy budget should be typically smaller than $1^{26}$. Even for a higher budget, the accuracy of DiffGen is comparable to DiffP-C4.5. The major advantage of our algorithm is that we publish data and the data miner has much better flexibility to perform the required data analysis. On the other hand, in DiffP-C4.5 the classifier is built through interactive
queries; therefore, the database has to be permanently shut down to satisfy the privacy requirement after generating only one classifier. The experimental result also shows that \textit{DiffGen} performs better than \textit{TDS}. For a higher anonymity threshold \(k\), the accuracy of \textit{TDS} will be lower. One advantage of \textit{DiffGen} is that, unlike \textit{TDS}, it does not need to ensure that every equivalence group contains \(k\) records; therefore, \textit{DiffGen} is able to provide more detailed information than \textit{TDS}. This result demonstrates for the first time that, if designed properly, a differentially private algorithm can provide better utility than a partition-based approach.

All the previous experiments can finish the sanitization process within 30 seconds. We further study the scalability of our algorithm over large data sets. We generate different data sets of different sizes by randomly adding records to the Adult data set. For each original record \(r\), we create \(\alpha - 1\) variations of the record by replacing some of the attribute values randomly from the same domain. Here \(\alpha\) is the blowup scale and thus the total number of records is \(\alpha \times 45,222\) after adding random records. Figure 5(b) depicts the runtime from 200,000 to 1 million records for \(h = 15\) and \(\epsilon = 1\).

\textbf{Summary.} We observed two general trends from the experiments. First, the privacy budget has a direct impact on the classification accuracy. A higher budget results in better accuracy since it ensures better attribute partitioning and lowers the magnitude of noise that is added to the count of each equivalence group. Second, the classification accuracy initially increases with the increase of the number of specializations, however, decreases with the increase of the number of specializations after a certain threshold. This is an interesting observation. The number of equivalence groups increases quite rapidly with an increase in the number of specializations, resulting in a smaller count per
group. Up to a certain threshold it has a positive impact due to more precise values; however, the influence of the Laplace noise gets stronger as the number of specializations grows. Note that if the noise is as large as the count, then the disclosed data are useless. This confirms that listing all the possible combination of values (i.e., contingency table) and then adding noise to their counts is not a good approach for high-dimensional data since the noise will be as big as the count. Since this is a non-interactive approach, the data owner can try different values of $h$ to find the threshold and then release the sanitized data. Determining a good value of $h$ adaptively, given the data set and the privacy budget, is an interesting approach for future work that we plan to investigate.

5 Discussion

Does the algorithm yield globally optimal solution? Can the algorithm be easily modified to anonymize sequential data? Are noise addition and generalization-based techniques desirable for the users of medical data sets? In this section, we provide answers to these questions.

**Globally optimal:** The proposed algorithm does not yield an optimal solution cut rather it is suboptimal. The algorithm uses exponential mechanism which probabilistically chooses a candidate with high score. Thus, it is possible that a different solution cut may provide better utility. However, it is important to note that maximizing the overall sum of the Score for specializations in the training data does not guarantee having the lowest classification error in the testing data.

**Sequential data:** While set-valued data (not considering the order) is useful for many data analysis tasks, we acknowledge that in some other analysis tasks the order of items could provide extra useful information and may poses new re-identification risks. The
The proposed anonymization algorithm can be extended to handle sequential data as well. In order to anonymize sequential data, we need to preprocess each item in the sequence by its order and then consider the item-order pair as the new “item”. Then, these new item-sets could be used as an input to our algorithm, which will then prevent re-identification attacks based on the order of items. But this is not currently implemented in the paper.

**Usefulness of our approach:** One of the inputs of our algorithm is the number of generalizations. Data users could specify the desired degree of generalizations by setting a proper value for $h$. Concerning the negative impact of noise (i.e., to the value of the data utility) added to satisfy differential privacy, we expect the sanitized data from our algorithm to be useful for a few tasks (for example, classification as illustrated in our paper), but not all data analysis tasks, especially those focus on attribute values of individuals. This is an inherent limitation of differential privacy methodology, and it is the price we pay to achieve provable privacy guarantee. We acknowledge this limitation and will sought for better solutions in future work.

In summary, the generalization technique used in *DiffGen* might result in loss of information, which leads to a tradeoff problem between data utility and privacy protection, like many other privacy enhancement algorithms. The generalization is also context dependent. For example, if we want to generalize diseases (i.e., asthma with respiratory disease and psoriasis with dermatological disease), resultant outputs might be adequate to answer some scientific questions (i.e., classification tasks determined upfront) but may be insufficient to answer others, (e.g. immunologic diseases, which could encompass both asthma and psoriasis). This is a fundamental limitation of generalization techniques and the type of lumping must be used carefully.
6 Conclusion

This paper presents a new anonymization algorithm that achieves \( \varepsilon \)-differential privacy and supports effective classification analysis for heterogeneous health data. The proposed solution connects the classical generalization technique with output perturbation to effectively sanitize raw data. Experimental results suggest that the proposed solution provides better utility than a pure differentially private interactive approach or an approach to simply produce \( k \)-anonymous data.
Acknowledgement

We thank Dr. Shuang Wang for helping on the experiments and helpful discussions.

Funding

BF, NM and RC is supported in part by the Discovery Grants (356065-2008), and Canada Graduate Scholarships from the Natural Sciences and Engineering Research Council of Canada (NSERC). XJ and LO-M were funded in part by the NIH grants 1K99LM 011392–01, R01LM009520, U54 HL108460, R01HS019913, and UL1RR031980.

Competing interests

None

Contributorship Statement

The authors are ranked according to their contributions. The authorship credit is based on meeting all following criteria: (1) substantial contributions to conception and design, acquisition of data, or analysis and interpretation of data; (2) drafting the article or revising it critically for important intellectual content; and (3) final approval of the version to be published.

License

This is an open-access article distributed under the terms of the Creative Commons Attribution Non-commercial License, which permits use, distribution, and reproduction in any medium, provided the original work is properly cited, the use is noncommercial and is otherwise in compliance with the license. See:

http://creativecommons.org/licenses/by-nc/2.0/ and
http://creativecommons.org/licenses/by-nc/2.0/legalcode.
References

15. Wong RC, W. L, J., Fu AWC, WANG K. (a,k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing. International Conference on Knowledge


32. Saeed M, Villarroel M, Reisner AT, et al. Multiparameter Intelligent Monitoring in

Appendix A1: Differential Privacy

Differential privacy is a recent privacy definition that provides a strong privacy guarantee. Partition-based privacy models ensure privacy by imposing syntactic constraints on the output. For example, the output is required to be indistinguishable among \( k \) records, or the sensitive value to be well represented in every equivalence group. Instead, differential privacy guarantees that an adversary learns nothing more about an individual, regardless of whether her record is present or absent in the data. Informally, a differentially private output is insensitive to any particular record. Therefore, from an individual's point of view, the output is computed as if from a data set that does not contain her record.

**Definition A1.1 (\( \varepsilon \)-differential privacy)** A randomized algorithm \( Ag \) is differentially private if for all data sets \( D \) and \( D' \) where their symmetric difference contains at most one record (i.e., \(|D \Delta D'| \leq 1\)), and for all possible sanitized data sets \( \tilde{D} \),

\[
Pr[Ag(D) = \tilde{D}] \leq e^{\varepsilon} \times Pr[Ag(D') = \tilde{D}],
\]

where the probabilities are over the randomness of the \( Ag \).

The parameter \( \varepsilon > 0 \) is public and specified by a data owner. Lower values of \( \varepsilon \) provide a stronger privacy guarantee. Typically, the values of \( \varepsilon \) should be small, such as 0.01, 0.1, or in some cases \( \ln 2 \), or \( \ln 3 \). When \( \varepsilon \) is very small, we have \( e^{\varepsilon} \approx 1 + \varepsilon \).

A standard mechanism to achieve differential privacy is to add random noise to the true output of a function. The noise is calibrated according to the sensitivity of the function. The sensitivity of a function is the maximum difference of its outputs from two data sets that differ only in one record.

**Definition A1.2 (sensitivity)** For any function \( f : D \to \mathbb{R}^d \), the sensitivity of \( f \) is

\[
\Delta f = \max_{D, D'} \|f(D) - f(D')\|_1
\]

(2)
for all $D, D'$ with $|D \Delta D'| \leq 1$.

**Example A1.1** Consider the raw data set of Table 1. Let $f$ be a function that counts the number of records with Age less than 40. Then, the $\Delta f$ is 1 because $f(D)$ can differ at most 1 due to the addition or removal of a single record.

**Laplace Mechanism.** Dwork et al. propose the Laplace mechanism. The mechanism takes a data set $D$, a function $f$, and the parameter $\lambda$ that determines the magnitude of noise as inputs. It first computes the true output $f(D)$, and then perturbs the output by adding noise. The noise is generated according to a Laplace distribution with probability density function $Pr(x|\lambda) = \frac{1}{2\lambda} \exp(-|x|/\lambda)$; its variance is $2\lambda^2$ and mean is 0. The following theorem connects the sensitivity to the magnitude of noise and guarantees that the perturbed output $\hat{f}(D) = f(D) + \text{Lap}(\lambda)$ satisfies $\varepsilon$-differential privacy, where $\text{Lap}(\lambda)$ is a random variable sampled from the Laplace distribution.

**Theorem A1.1** For any function $f: D \to \mathbb{R}^d$, the algorithm $Ag$ that adds independently generated noise with distribution $\text{Lap}(\Delta f / \varepsilon)$ to each of the $d$ outputs satisfies $\varepsilon$-differential privacy.

**Example A1.2** Continue from Example 2. The mechanism that returns $\hat{f}(D) = f(D) + \text{Lap}(1/\varepsilon)$ gives $\varepsilon$-differential privacy.

**Exponential Mechanism.** McSherry and Talwar propose the exponential mechanism that can choose an output $t \in \mathcal{T}$ that is close to the optimum with respect to a utility function while preserving differential privacy. The exponential mechanism takes as inputs a data set $D$, output domain $\mathcal{T}$, privacy parameter $\varepsilon$, and a utility function $u: (D \times \mathcal{T}) \to \mathbb{R}$ that assigns a real valued score to every output $t \in \mathcal{T}$, where a higher score means better utility.

The mechanism induces a probability distribution over the domain $\mathcal{T}$ and then samples an output $t \in \mathcal{T}$. Let $\Delta u = \max_{t \in \mathcal{T}, D, D'} |u(D, t) - u(D', t)|$ be the sensitivity of the utility.
function. The probability associated with each output is proportional to \( \exp\left(\frac{Eu(D,t)}{2\Delta u}\right) \); that is, the output with a higher score is exponentially more likely to be chosen.

**Theorem A1.2**

For any function \( u: (D \times T) \rightarrow \mathbb{R} \), an algorithm \( Ag \) that chooses an output \( t \) with probability proportional to \( \exp\left(\frac{Eu(D,t)}{2\Delta u}\right) \) satisfies \( \varepsilon \)-differential privacy.

### Appendix A2: Privacy analysis

We elaborate the key steps of the algorithm: (1) selecting a candidate for specialization, (2) determining the split value, and (3) publishing the noisy counts. We show that each of these steps preserves privacy, and then we use the composition properties of differential privacy to guarantee that DiffGen is \( \varepsilon \)-differentially private.

**1) Candidate Selection.** We use exponential mechanism (see Appendix A1) to select a candidate for specialization in each round. We define two utility functions to calculate the score of each candidate \( v \in \cup \text{Cut}_i \). The first utility function is information gain. Let \( D_v \) denote the set of records in \( D \) generalized to the value \( v \). Let \( |D_v^{cls}| \) denote the number of records in \( D_v \) having the class value \( cls \in \Omega(A^{cls}) \). Note that \( |D_v| = \sum_c |D_{v,c}| \), where \( c \in \text{child}(v) \). Then, we get

\[
\text{InfoGain}(D,v) = H_v(D) - H_v|c(D),
\]

where \( H_v(D) = -\sum_{cls} \frac{|D_v^{cls}|}{|D_v|} \times \log_2 \frac{|D_v^{cls}|}{|D_v|} \) is the entropy of candidate \( v \) with respect to the class attribute \( A^{cls} \) and \( H_v|c(D) = \sum_c \frac{|D_{v,c}|}{|D_v|} H_c(D) \) is the conditional entropy given the candidate is specialized. The sensitivity of \( \text{InfoGain}(D,v) \) is \( \log_2 |\Omega(A^{cls})| \), where \( |\Omega(A^{cls})| \) is the domain size of the class attribute \( A^{cls} \). It is because the value of the
entropy $H_v(D)$ must be between 0 and $\log_2 |\Omega(A^{\text{cls}})|$. And, the value of the conditional entropy $H_{v|c}(D)$ lies between 0 and $H_v(D)$. Therefore, the maximum change of $\text{InfoGain}(D, v)$ due to the addition or removal of a record is bounded by $\log_2 |\Omega(A^{\text{cls}})|$.

The second utility function is:

$$Max(D, v) = \sum_{c \in \text{child}(v)} (\max_{D_c^{\text{cls}}})$$

(2)

$Max(D, v)$ is the summation of the highest class frequencies over all child values and the sensitivity of this function is 1 because the value of $Max(D, v)$ can vary at most 1 due to the change of a record.

Given the scores of all the candidates, exponential mechanism selects a candidate $v_i$ with the following probability,

$$\frac{\exp\left(\frac{\varepsilon}{\Delta u} u(D, v_i)\right)}{\sum_{v \in \text{Cut}_i} \exp\left(\frac{\varepsilon}{\Delta u} u(D, v)\right)}$$

(3)

where the $u(D, v)$ is either $\text{InfoGain}(D, v)$ or $Max(D, v)$ and the sensitivity of the function $\Delta u$ is $\log_2 |\Omega(A^{\text{cls}})|$ and 1, respectively. Thus, from Theorem A1.2, Line 7 of Algorithm 1 satisfies $\varepsilon'$-differential privacy. The beauty of the exponential mechanism is that while it ensures privacy, it also exponentially favors a candidate with a high score.

(2) **Split Value.** Once a candidate is determined, $\text{DiffGen}$ splits the records into child partitions. The split value of a categorical attribute is determined according to the taxonomy tree of the attribute. Since the taxonomy tree is fixed, the sensitivity of the split value is 0. Therefore, splitting the records according to the taxonomy tree does not violate differential privacy.

For numerical attributes, a split value cannot be directly chosen from the attribute values that appear in the data set $D$, because the probability of selecting the same split value from a different data set $D'$ not containing this value is 0. We again use an exponential
mechanism to determine the split value. We first partition the domain into intervals $I_1, \ldots, I_k$ such that all values within an interval have the same score. Then, the exponential mechanism is used to select an interval $I_i$ with the following probability,

$$\frac{\exp\left(\frac{\epsilon}{2\Delta u} u(D, v_i) \times |\Omega(I_i)|\right)}{\sum_{j=1}^{k} \left(\exp\left(\frac{\epsilon}{2\Delta u} u(D, v_j) \times |\Omega(I_j)|\right)\right)}$$

(4)

where $v_i \in \Omega(I_i)$, and $|\Omega(I_i)|$ is the length of the interval. After selecting the interval, the split value is determined by sampling a value uniformly from the interval. Thus, the probability of selecting a value $v_i \in \Omega(A_i)$ is

$$\frac{\exp\left(\frac{\epsilon}{2\Delta u} u(D, v_i)\right)}{\iint_{v \in \Omega(A_i)} \exp\left(\frac{\epsilon}{2\Delta u} u(D, v)\right) dv}$$

(5)

This satisfies $\epsilon'$-differential privacy because the probability of choosing any value is proportional to $\exp\left(\frac{\epsilon u(D, v_i)}{2\Delta u}\right)$.

For set-valued attributes, specialization results in a total of $2^{|\text{child}(v)|}$ child partitions, where $|\text{child}(v)|$ is the number of $v$'s children. Hence, we want to prune empty child partitions as early as possible. Due to noise required by differential privacy, a child partition cannot be deterministically identified as non-empty. We issue a counting query for the noisy size of each child partition by Laplace mechanism. We use the noisy size to make our decision. We consider a sub-partition ``non-empty'' if its noisy size $\geq \frac{\sqrt{2}}{\epsilon}$.

We design the threshold as a function of the standard deviation of the noise. While this heuristic is arbitrary, it performs well experimentally.

(3) Noisy Counts. Each leaf partition contains $|\Omega(A^{\text{leaf}})|$ equivalence groups. Publishing the exact counts of these groups does not satisfy differential privacy since for a different data set $D'$, the counts may change. This change can be easily offset by adding noise to the count of each group according to the Laplace mechanism (See Theorem A1.1). As
discussed earlier, the sensitivity of count query is 1; therefore, to satisfy $\frac{\epsilon}{2}$-differential privacy, DiffGen adds $\text{Lap}(2/\epsilon)$ noise to each true count $c$ of the groups (Line 12). We post-process the noisy counts by rounding each count to the nearest non-negative integer. Note that post-processing does not violate the differential privacy.

Next, we use composition properties of differential privacy to guarantee that the proposed algorithm satisfies $\epsilon$-differential privacy as a whole.

**Lemma 1 (Sequential composition)** Let each $A_{\epsilon_i}$ provide $\epsilon_i$-differential privacy. A sequence of $A_{\epsilon_i}(D)$ over the data set $D$ provides $(\sum \epsilon_i)$-differential privacy.

**Lemma 2 (Parallel composition)** Let each $A_{\epsilon_i}$ provide $\epsilon$-differential privacy. A sequence of $A_{\epsilon_i}(D_i)$ over a set of disjoint data sets $D_i$ provides $\epsilon$-differential privacy.

Any sequence of computations that each provides differential privacy in isolation also provides differential privacy in sequence, which is known as **sequential composition**. However, if the sequence of computations is conducted on disjoint data sets, the privacy cost does not accumulate but depends only on the worst guarantee of all computations. This is known as **parallel composition**.

**Theorem 1** DiffGen is $\epsilon$-differentially private.

Proof. (Sketch) The algorithm first determines the split value for each numerical attribute using the exponential mechanism (Line 4). Since the cost of each exponential mechanism is $\epsilon'$, Line 4 of the algorithm preserves $\epsilon'|A_n^{PR}|$-differential privacy, where $|A_n^{PR}|$ is the number of numerical attributes.

In Line 7, the algorithm selects a candidate for specialization. This step uses the exponential mechanism and thus, candidate selection step guarantees $\epsilon'$-differential privacy for each iteration. In Line 8, the algorithm splits the records into child partitions. For set-valued candidate, $\epsilon'$ privacy budget is used to determine the non-empty partitions. In Line 9, the nomalgorithm determines the split value for each new numerical
candidate \( v_n \in \cup \text{Cut}_i \). All records in the same partition have the same generalized values on \( A^{pr} \); therefore, each partition can only contain at most one candidate value \( v_n \). Thus, determining the split value for the new candidates requires at most \( \epsilon' \) privacy budget for each iteration due to the parallel composition property. Thus, for each iteration, the required privacy budget is \( \epsilon, 2\epsilon, \) or \( 2\epsilon \), if the candidate is categorical, set-valued, or numerical, respectively.

Finally, the algorithm outputs the noisy count of each group (Line 12) using the Laplace mechanism and guarantees \( \frac{\epsilon}{2} \)-differential privacy. Therefore, for \( \epsilon' = \frac{\epsilon}{2(|A^{pr}|+2n)} \), \( \text{DiffGen} \) is \( \epsilon \)-differentially private due to the sequential composition property.

Appendix A3: Implementation of the Algorithm

A simple implementation of \( \text{DiffGen} \) is to scan all data records to compute scores for all candidates in \( \cup \text{Cut}_i \). Then scan all the records again to perform the specialization. A key contribution of this work is an efficient implementation of the proposed algorithm that computes scores based on some information maintained for candidates in \( \cup \text{Cut}_i \) and provides direct access to the records to be specialized, instead of scanning all data records. We briefly explain the efficient implementation of the algorithm as follows.

**Initial Steps (Lines 1-5).** Initially, we determine split points for all numerical candidates (Line 4). First, the data is sorted with respect to the split attribute, which requires \( O(|D| \log |D|) \). Then the data is scanned once to determine the score for all attribute values that appear in the data set \( D \). An interval is represented by two successive different attribute values. Finally, the exponential mechanism is used to determine the split point. We also compute the scores for all candidates \( v \in \cup \text{Cut}_i \) (Line 5). This can be done by scanning the data set once. However, for each subsequent iteration, information
needed to calculate scores comes from the update of the previous iteration (Line 10). Thus the worst-case runtime of this step is $O(|\mathcal{A}^{PT}| \times |D| \log |D|)$.

**Perform Specialization (Line 8).** To perform a specialization $v \rightarrow \text{child}(v)$, we need to retrieve $D_v$, the set of data records generalized to $v$. To facilitate this operation we organize the records in a tree structure, with each root-to-leaf path representing a generalized record over $\mathcal{A}^{PT}$, as shown in Figure 2. Each leaf partition (node) stores the set of data records having the same generalized record for $\mathcal{A}^{PT}$ attributes. For each $v$ in $\bigcup \text{Cut}_i$, $P_v$ denotes a leaf partition whose generalized record contains $v$, and $\text{Link}_v$ provides direct access to all $P_v$ partitions generalized to $v$. For example, $\text{Link}_{\text{Professional}}$ provides a direct access to all partitions containing the value Professional as shown in Figure 2.

Initially, the tree has only one leaf partition containing all data records, generalized to the topmost value on every attribute in $\mathcal{A}^{PT}$. In each iteration we perform a specialization $v$ by refining the leaf partitions on $\text{Link}_v$. For each value $c \in \text{child}(v)$ for the categorical and numerical attribute, a new child partition $P_c$ is created from $P_v$, and data records in $P_v$ are split among the new partitions. For set-valued attribute, the child partitions can be exhaustively generated by replacing $v$ by the combinations of its children $c \in \text{child}(v)$. For example, the partition $\{**\}$ generates three child partitions: $\{1\}$, $\{2\}$ and $\{1,2\}$. This technique, however, is inefficient. We propose an efficient implementation by separately handling non-empty and empty child partitions of a partition $P_v$. Non-empty child partitions, usually of a small number, need to be explicitly generated. For empty child partitions, we do not explicitly generate all possible ones, but employ a test-and-generate method: generate a uniformly random empty child partition without replacement only if the noisy count of an empty child partition is greater than or equal to a threshold. To satisfy differential privacy, empty and non-empty child partitions must
use the same threshold $\sqrt{2}/\varepsilon'$.

This is the only operation in the whole algorithm that requires scanning data records. In the same scan, we also collect the following information for each $c$: $|D_c|$, $|D_g|$, $|D_c^{cls}|$ and $|D_g^{cls}|$, where $g \in \text{child}(c)$ and $cls$ is a class label. These pieces of information are used in Line 10 to update scores. The main computational cost comes from the distribution of records from a partition to its child partitions. Thus, the total runtime of this step is $O(|D|)$ because the partitioning process for specialization can affect at most $|D|$ records in each iteration.

**Determine the Split Value (Line 9).** If a numerical candidate $v_n$ is selected in Line 7, then we need to determine the split points for two new numerical candidates $c_n \in \text{child}(v_n)$. This step takes time $O(|D| \log |D|)$.

**Update Score (Line 10).** Both InfoGain and Max scores of the other candidates $x \in \bigcup \text{Cut}_i$ are not affected by $v \to \text{child}(v)$, except that we need to compute the scores of each newly added value $c \in \text{child}(v)$. The scores of the new candidates are computed using the information collected in Line 8. Thus, this step can be done in constant $O(1)$ time.

**Exponential Mechanism (Lines 4, 7 and 9).** The cost of the exponential mechanism is proportional to the number of discrete alternatives from which it chooses a candidate. For Line 7, the cost is $O(| \bigcup \text{Cut}_i |)$, and for Lines 4 and 9 the cost is $O(|I|)$, where $|I|$ is the number of intervals. Usually both $| \bigcup \text{Cut}_i |$ and $|I|$ are much smaller than $|D|$.

In summary, the cost of the initial steps and Lines 7-10 are $O(|A^{pt'}| \times |D| \log |D|)$ and $O(h \times |D| \log |D|)$, respectively. Hence, for a fixed number of attributes the total runtime of **DiffGen** is $O(h \times |D| \log |D|)$. 
Appendix A4: Related Work

Achieving an appropriate trade-off between individual privacy and data utility has been the essential goal in many data sharing applications, especially in the case of secondary use of health data. In this section, we review the state-of-the-art techniques for privacy-preserving relational and set-valued data publishing.

A4.1 Relational Data Sanitization

In the current practice of secondary use of health data, several de-identification heuristics are commonly used for the purpose of privacy protection\(^7\). These heuristics are all based on the concept of quasi-identifiers. Samarati and Sweeney formalized the \(k\)-anonymity privacy model\(^6\) to prevent privacy attacks from quasi-identifiers. There have been a plethora of research works proposing different algorithms for achieving \(k\)-anonymity and its extensions\(^9, 10\). El Emam et al.\(^11\) proposed the optimal lattice anonymization (OLA) algorithm for sanitizing health data, which searches for an optimal \(k\)-anonymous solution over a generalization lattice generated by global generalization. Iyengar\(^12\) considered the anonymity problem for classification and proposed a genetic algorithmic solution. Bayardo and Agrawal\(^13\) also addressed the classification problem using the same classification metric as that in\(^12\). Fung et al.\(^14\) proposed a top-down specialization (TDS) approach to generalize a relational data table. Recently, LeFevre et al.\(^15\) proposed another anonymization technique for classification using multidimensional recoding\(^16\). A more complete summary of existing techniques for achieving \(k\)-anonymity and its extensions can be found in the survey paper\(^17\). A review of anonymization techniques particular for health data is provided in\(^18\).

Differential privacy has received considerable attention recently as a substitute for
partition-based privacy models in privacy-preserving data publishing. However, most of the research on differential privacy so far concentrates on the interactive setting with the goal of reducing the magnitude of added noise\cite{3,19,20}, releasing certain data mining results\cite{21,22}, or determining the feasibility and infeasibility results of differentially-private mechanisms\cite{23,24}. A general overview of various research works on differential privacy can be found in the recent survey\cite{2}. Current techniques that adopt the non-interactive approach publish contingency tables or marginals of raw data\cite{3,25-27}. Barak \textit{et al.} \cite{25} addressed the problem of releasing a set of consistent marginals of a contingency table. Xiao \textit{et al.} \cite{27} proposed \textit{Privelet}, a wavelet-transformation-based approach that lowers the magnitude of noise needed to ensure differential privacy to publish a multidimensional frequency matrix. Hay \textit{et al.} \cite{26} proposed a method to publish differentially private histograms for a one-dimensional data set. Though closely related, all these works do not address the problem of privacy-preserving data publishing for heterogeneous health data, which is the primary theme of this paper.

\textbf{A4.2 Set-Valued Data Sanitization}

The increasing prevalence of set-valued data has resulted in new types of privacy attacks. A large number of research works on privacy-preserving set-valued data publishing have appeared in the literature. Ghinita \textit{et al.}\cite{28} proposed a bucketization-based approach, which limits the probability of inferring a sensitive item to a specified threshold, while preserving correlations among items for frequent pattern mining. Xu \textit{et al.} \cite{29} bounded the background knowledge of an adversary to at most $p$ non-sensitive items, and employed global suppression to preserve as many item instances as possible. Xu \textit{et al.} \cite{30} improved the technique in \cite{29} by preserving frequent itemsets and presenting a border representation. Cao \textit{et al.} \cite{31} further assumed that an adversary may possess
background knowledge on sensitive items and proposed a privacy notion $\rho$-uncertainty, which bounds the confidence of inferring a sensitive item from any itemset to $\rho$. Terrovitis et al.\textsuperscript{32,33} and He and Naughton\textsuperscript{34} eliminated the distinction between sensitive and non-sensitive items. Any item could be both sensitive and non-sensitive at the same time. Similar to the ideas\textsuperscript{29,30}, Terrovitis et al.\textsuperscript{33} proposed to bound the background knowledge of an adversary by the maximum number $m$ of items and propose $k^m$-anonymity, a relaxation of $k$-anonymity. They achieved $k^m$-anonymity by a bottom-up global generalization solution. To improve the utility, recently Terrovitis et al.\textsuperscript{32} provided a local recoding method for achieving $k^m$-anonymity. He and Naughton\textsuperscript{34} pointed out that $k^m$-anonymity provides a weaker privacy protection than $k$-anonymity and proposed a top-down local generalization solution under $k$-anonymity. However, recent research has indicated that even $k$-anonymity provides insufficient privacy protection for set-valued data. Lately, Chen et al.\textsuperscript{35}, for the first time, applied differential privacy to set-valued data sanitization. They proposed a probabilistic top-down partitioning algorithm, which scales linearly with the input data size. In spite of the extensive research on privacy-preserving relational and set-valued data publishing, the emerging publishing scenario in which relational data and set-valued data need to be published simultaneously, a very common scenario in secondary use of health data is largely neglected. This motivates our research, which addresses the problem of heterogeneous health data publication. This paper is the extension of our previous work\textsuperscript{36}, where we proposed a generalized-based differentially private data release algorithm for relational data. In this paper, we formally define the heterogeneous data sanitization problem and extend our proposed algorithm to handle both relational and set-valued data.
Appendix A5: Open Source Software

We developed DiffGen and made it public on sourceforge. Please check out through the following address using subversion (SVN) http://sourceforge.net/projects/diffgen/. The algorithms were developed in c++ in a windows environment. We used Microsoft Visual studio for development. Under the truck of the SVN directory, users can find our source code under the code folder. We also provide an example folder containing two examples of DiffGen, i.e., Information gain version (IG) and maximum frequency version (MAX) for readers' reference. Two .bat files (i.e., runMimic_IG.bat and runMimic_MAX.bat) were provided for users to quickly test. Our sample data for this example were extracted from the MIMIC II database, for which a data agreement must be obtained from Physionet (http://mimic.physionet.org/) before we can share the data with authorized users.

Figure 1: Screenshot of running diffGen (i.e., maximum frequency version).
References