2009 Sixth International Conference on Information Technology: New Generations

Extending ℓ-Diversity for Better Data Anonymization

Hongwei Tian and Weining Zhang
Department of Computer Science, University of Texas at San Antonio
{htian, wzhang}@cs.utsa.edu

***Abstract*—The notions of ℓ-diversity provides a strong privacy guarantee for generalization. However, existing ℓ-diversity algorithms may force users to choose between publishing no data or scarifying privacy if the data have a skewed distribution of SA values. In this paper, we solve this problem by extending ℓ-diversity in two ways. First, we allow the generalization of SA values and second, we use a simple function to constraint frequencies of SA values. The resulting (τ, ℓ)-diversity is more flexible and elaborate. We present an efficient heuristic algorithm that uses a novel order of quasi-identifier values to achieve (τ, ℓ)-diversity. We compare our algorithm with two state-of-the-art algorithms based on existing ℓ-diversity measures. Our preliminary experimental results indicate that our algorithm can not only effectively deal with data with skewed SA distributions but also result in better utility of anonymous data in general.**

***Index Terms*—Data Privacy, Anonymization, Generalization, Algorithm, Experiments**

I. Introduction

We consider anonymization of data tables for research and data mining. Many recent anonymization methods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] protect privacy using generalization, a technique that recodes specific data values using more general ones.

In a generalization framework, the attributes of a table are divided into a quasi-identifier (QI) and an sensitive attribute (SA). A table $\mathcal{T}$ can be partitioned on QI into a set $\mathcal{P}$ of equivalence classes (ECs) so that tuples in an EC have an identical QI value. A fundamental notion of privacy in this context is ℓ-diversity [1], [7], which requires each EC to contain at least ℓ well-represented SA values.

Definition 1: (based on [1], [7]) A table satisfies the *simple ℓ-diversity* if $f_1 \leq \frac{1}{\ell}$ for every equivalence class E, where f_1 is the frequency of the most frequent base SA value in E; it satisfies the *recursive (c, ℓ)-diversity* if $f_1 < c(\sum_{i=\ell}^{m} f_i)$ for every equivalence class E, where f_i is the frequency of the i^{th} frequent base SA value in E and $c \geq 1$ is a constant.

The ℓ-diversity guarantees that an adversary cannot identify the SA value of a target person with 100% confidence unless she has the background knowledge to rule out at least $\ell - 1$ SA values in the EC of the target person. Despite this strong privacy guarantee, since existing measures of ℓ-diversity do not permit the generalization of SA values, algorithms based on these measures can be difficult to use in practice if data have a skewed SA distribution.

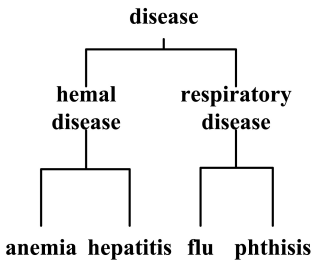
This work was supported in part by NFS grant IIS-0524612.

	Age	Zipcode	Disease
1 (Arthur)	20	10000	hepatitis
2 (Bob)	20	10001	hepatitis
3 (Carol)	20	10002	hepatitis
4 (David)	20	10003	hepatitis
5 (Ella)	29	10004	phthisis
6 (Frank)	33	10004	hepatitis
7 (Grace)	32	10005	phthisis
8 (Henry)	31	10006	phthisis
9 (Iris)	30	10007	phthisis
10 (Jim)	30	10005	anemia
11 (Kate)	28	10019	hepatitis
12 (Jim)	29	10017	flu

(a) Microdata

Age	Zipcode	Disease
[20,29]	100**	hepatitis
[20,29]	100**	hepatitis
[20,29]	100**	hepatitis
[20,29]	100**	hepatitis
[20,29]	100**	phthisis
[30,39]	1000*	hepatitis
[30,39]	1000*	phthisis
[30,39]	1000*	phthisis
[30,39]	1000*	phthisis
[30,39]	1000*	anemia
[20,29]	100**	hepatitis
[20,29]	100**	flu

(b) $(3, 2)$-diversity



(c) Taxonomy of Disease

Age	Zipcode	Disease
[20,39]	1000*	hepatitis
[20,39]	1000*	hepatitis
[20,39]	1000*	hepatitis
[20,39]	1000*	hepatitis
[20,39]	1000*	phthisis
[20,39]	1000*	hepatitis
[20,39]	1000*	phthisis
[20,39]	1000*	phthisis
[20,39]	1000*	phthisis
[20,39]	1000*	anemia
[20,29]	1001*	hepatitis
[20,29]	1001*	flu

(d) Simple 2-diversity

Fig. 1. Examples of ℓ-Diversity Anonymous Data

We call privacy requirements (i.e., ℓ and/or c) that let an ℓ-diversity algorithm output tables of some utility the *eligible requirements*, and the range of such requirements the *eligible range*. The distribution of SA values of a table imposes limits on eligible ranges. In particular, for simple ℓ-diversity, it limits the maximum ℓ; for (c, ℓ)-diversity, if c is fixed, it limits the maximum ℓ, and if ℓ is fixed, it limits the minimum c. Many real-world datasets have skewed distributions and impose narrow eligible ranges. For example, for simple ℓ-diversity, most datasets in UCI repository [11] have an eligible range no larger than $2 \leq \ell \leq 6$. Some of them have an eligible range of $\ell = 2$ and some others even do not have an eligible range (that is, no ℓ can produce non-empty table). Using existing ℓ-diversity algorithms on such datasets can cause two problems.

1) Risk of negative disclosure.
   A negative disclosure occurs if the published table allows the adversary to determine with a high probability that a target person does not have a certain SA value. For example, by seeing a table containing 100 distinct

978-0-7695-3596-8/09 $25.00 © 2009 IEEE
DOI 10.1109/ITNG.2009.144

	Age	Zipcode	Disease
1	[20,29]	1000*	hemal disease
2	[20,29]	1000*	hemal disease
3	[20,39]	100**	hepatitis
4	[20,29]	1000*	hepatitis
5	[20,29]	1000*	phthisis
6	[30,39]	1000*	hepatitis
7	[30,39]	1000*	phthisis
8	[30,39]	1000*	phthisis
9	[20,39]	100**	phthisis
10	[30,39]	1000*	anemia
11	[20,39]	100**	hepatitis
12	[20,39]	100**	flu

(a) $(0.5, 3)$-diversity

EC	Disease	Induced Frequency
1	hepatitis	0.5
1	phthisis	0.25
1	anemia	0.25
1	flu	0
2	phthisis	0.5
2	hepatitis	0.25
2	anemia	0.25
2	flu	0
3	hepatitis	0.5
3	phthisis	0.25
3	flu	0.25
3	anemia	0

(b) Induced SA distribution

Fig. 2. An Example of (τ, ℓ)-Diversity

SA values and an EC containing 2 distinct SA values, the adversary knows with a probability of 1 which of the 98 SA values a target person does not have. A negative disclosure occurs if the adversary does not know this before seeing the data. To reduce the risk of negative disclosure using ℓ-diversity, the data owner needs to specify a large ℓ. But if eligible range of ℓ is too narrow, the data owner can be forced to choose between publishing no data and sacrificing privacy; neither is satisfactory.

2) Risk of positive disclosure.
A positive disclosure occurs if the published table allows the adversary to identify with a high probability that the SA value of a target person is within a very small set. For example, in Figure 1, table (a), a microdata of a hospital containing one tuple per patient, imposes an eligible range $\ell = 2$ and $c = 3$ for (c, ℓ)-diversity (for any other ℓ and c, the output table is either empty or of no utility). However, with table (b), a generalization of table (a) satisfying $(3, 2)$-diversity, the adversary can determine with a probability of 0.714 that any person whose anonymous tuple is in the EC with QI value <[20,29], 100**> will have *hepatitis*. Table (a) also imposes an eligible range of $\ell = 2$ for simple ℓ-diversity. However, with table (d), which satisfies 2-diversity, the adversary can determine with a probability of 0.9 that any person whose anonymous tuple is in the EC with QI value <[20,39], 1000*> will have either *hepatitis* or *phthisis*. In general, a positive disclosure can happen even if the adversary could not uniquely identify the true SA value of the target person. To reduce the risk of positive disclosure, the data owner needs to specify a large ℓ (and also a small c in case of (c, ℓ)-diversity). Again, if the eligible range is too narrow, the data owner can be forced to choose between publishing no data and sacrificing privacy.

In this paper, we solve these problems by extending ℓ-diversity in two ways. First, we allow the generalization of SA values. If an EC contains a tuple t with a general SA value a, all base (i.e., leaf) SA values under a in the taxonomy are said to be induced by a. Unless she has prior knowledge to believe otherwise, the adversary must assume that any base SA value induced by a is equally likely to be the original SA value of t. Thus, the frequencies of the induced base SA values are estimated from frequency of a. We can require that each EC must contain at least ℓ well-represented induced base SA values. Now, the eligible range of ℓ is no longer restricted by the distribution of SA values in the original table because it can be easily expanded by generalizing SA values. Thus, we can effectively control the negative disclosure. Secondly, we can use some simple function to restrict the frequencies of induced SA values in each EC, and require the highest frequency to be bounded by τ, a user specified threshold. This allows us to effectively control positive disclosure. As a result, we introduce a new ℓ-diversity measure called the functional (τ, ℓ)-diversity. Our specific contributions are as follows:

1) We define functional (τ, ℓ)-diversity and use a linear function to specifies an upper bound on cumulative frequency of ℓ dominant induced SA values.
2) We present a heuristic algorithm that finds a good table of (τ, ℓ)-diversity using a novel partial order of QI values that takes into consideration the importance of QI attributes and the information retained in QI values.
3) We compare the privacy and utility achieved by our algorithm with those achieved by two existing ℓ-diversity algorithms. Preliminary experimental results indicate that our method can produce useful anonymous data in many cases when existing methods cannot output any data, and in many other cases, our method can often produce much better data than existing methods can, and do so with a comparable performance.

The rest of the paper is organized as follows. In Section II, we define the notion of (τ, ℓ)-diversity and formulate the problem. In Section III, we present a heuristic algorithm for solving the (τ, ℓ)-diversity problem. In Section IV, we present experimental results. The Section V concludes the paper and discuss some future work.

II. Functional (τ, ℓ)-Diversity

Each attribute has a taxonomy in the form of a tree, in which leaf nodes are *base values* and nodes at higher levels are *more general than* nodes at lower levels. We assume that SA has m distinct base values. If a tuple contains only base values in its components, it is a *base tuple*, otherwise, it is a *general tuple*. We use $\succ$to denote "more general than" and $\succeq$ to denote "covers" (namely, $\succ$or $=$) relationships between values in a taxonomy or between tuples.

Definition 2: (**Induced Frequency**) Let E be an EC, a be a base SA value, and t be a tuple in E. The frequency of a induced by $t[SA]$ is

$$l(t[SA], a) = \begin{cases} \frac{p(a)}{\sum_{v \in leaves(t[SA])} p(v)}, & \text{if } a \preceq t[SA]; \\ 0, & \text{otherwise.} \end{cases}$$

where $leaves(v)$ is the set of leaves under v, and $p(a)$ is a weight assigned to base SA value a. The *induced frequency* of a in E is

$$f(a) = \frac{\sum_{t \in E} l(t[SA], a)}{|E|}$$

Intuitively, the weight of a base SA value can model the background knowledge of an adversary. If the adversary's

background knowledge is unknown, we can assign a weight 1 to a base SA value if it appears in the microdata, and 0, otherwise.

Example 1: Assume a weight of 1 for each base SA value. In Figure 2, table (b) shows the induced frequencies of ECs of table (a). Specifically, in the EC consisting of tuples 1, 2, 4, and 5, the induced frequency of *hepatitis* is $0.5 = \frac{0.5+0.5+1+0}{4}$.

Definition 3: (**Cumulative Frequency**) Let $f_1, \ldots, f_m$ be frequencies of base SA values $a_1, \ldots, a_m$ induced from an EC and $f_1 \geq f_2 \geq \cdots \geq f_m$. We call f_k the k^{th} *dominant SA frequency* and a_k *the* k^{th} *dominant SA value*. The *cumulative frequency* of the first k dominant SA values is $F(k) = \sum_{i=1}^{k} f_i$.

Example 2: Consider EC 1 of table (a) (with QI value <[20,29], 1000*>) in Figure 2. According to table (b), the cumulative frequencies of the EC are $F(1) = 0.5$, $F(2) = 0.75$, $F(3) = 1$ and $F(4) = 1$.

Definition 4: (**Functional** (τ, ℓ)**-diversity**) A partition $\mathcal{P}$ of microdata $\mathcal{T}$ is said to satisfy a functional (τ, ℓ)-diversity (or simply (τ, ℓ)-diversity) if for each equivalence class E, $F(k) \leq \psi(k)$, for every $1 \leq k \leq m$, where $F(k)$ is the cumulative frequency of the first k dominant SA values and $\psi(k)$ is an increasing function over $[1, m]$ with $\psi(1) = \tau$ and $\psi(\ell) = 1$.

In this paper, we use the following linear function:

$$\psi(k) = \begin{cases} \tau + \frac{1-\tau}{\ell-1}(k-1), & \text{if } 1 \leq k \leq \ell; \\ 1, & \text{if } \ell < k \leq m. \end{cases}$$

Example 3: In Figure 2, table (a) satisfies the $(0.5, 3)$-diversity. It contains three ECs each of which has the identical set of cumulative frequencies $F(1) = 0.5$, $F(2) = 0.75$, $F(3) = 1$ and $F(4) = 1$. These cumulative frequencies are no larger than their respective limits $\psi(1) = \tau = 0.5$, $\psi(2) = 0.75$, $\psi(3) = 1$ and $\psi(4) = 1$.

Theorem 1: Let E be an equivalence class of a (τ, ℓ)-diversified partition of original microdata and o be any individual whose tuple is in E. An adversary can infer, with a probability no higher than $\psi(k)$, that the SA value of o is among one of the k dominant SA values.

We measure the utility of anonymous tuples by the amount of information retained in them as follows.

Definition 5: (**Utility Measure**)The information of a value v in a taxonomy is $I(v) = \frac{1}{|leaves(v)|}$. The *information* of a tuple t is $I(t) = \sum_{A \in \mathcal{A}} I(t[A])$, where $\mathcal{A}$ is the set of attributes. The information of a table G is $I(G) = \sum_{t \in G} I(t)$. The *utility of* G *is* the fraction of information in the original table O that is retained by G, that is, $um(G) = \frac{I(G)}{I(O)}$.

Example 4: In Figure 2, the information of tuple 1 of table (a) is $\frac{1}{10} + \frac{1}{10} + \frac{1}{2} = 0.7$.

We can now define the (τ, ℓ)**-Diversity Problem** as follows.

> Given a microdata $\mathcal{T}$, a set of QI attributes, a set of SA attributes, and a privacy requirement (τ, ℓ), find a partition $\mathcal{P}$ of $\mathcal{T}$ that satisfies the (τ, ℓ)-diversity requirement and $um(\mathcal{P})$ is maximized.

Since the optimal (τ, ℓ)-diversity problem is *NP-hard*, we present a heuristic algorithm in Section III.

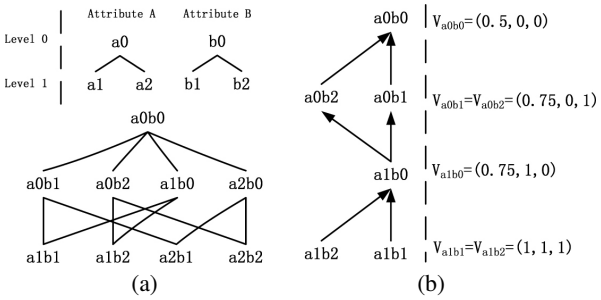


Fig. 3. Order of QI Values

III. Sequential Sweep Algorithm

The algorithm finds a partition of a table by moving base tuples from their initial ECs into some other ECs until all ECs are (τ, ℓ)-diversified. Since each EC has a unique QI value, the space of ECs is the lattice of QI values in a given set of QI attributes (see Figure 3(a) for an example). Initially, tuples are in ECs of their original QI values. If an EC does not satisfy the (τ, ℓ)-diversity requirement, some tuples in the EC can be moved (or swept) into other ECs that have more general QI values. The algorithm uses a heuristic order among ECs to determine a unique *next* EC for each tuple to move into. Thus, we can sweep ECs one by one in this order (thus the name sequential sweep). If the last EC still does not satisfy the distribution requirement, we can generalize SA values of some selected tuples in the EC.

A. Heuristic Order of QI Values

Let the QI consist of attributes $A_1, \ldots, A_d$. Each QI value is a d-tuple and also a node in the lattice defined by the taxonomies of these attributes. For any QI value q and any QI attribute A_i, $q[A_i]$ is at some level in taxonomy of A_i, where the root is at level 0. We denote the level j of taxonomy of attribute A_i by $L_{i,j}$ and the set of values at $L_{i,j}$ by $S_{i,j}$. We assume that taxonomies of attributes are represented in an appropriate form so that both $S_{i,j}$ and $L_{i,j}$ can be easily obtained. In addition, each value (node) in a taxonomy is associated with two quantities: its level and the number of leaves it covers. We also assume that A_i is more important than A_{i+1} for $1 \leq i < d$. Here the importance of attributes can be based on their semantics, the complexity of their taxonomy, or the preference of a user.

As a heuristic, we measure the information retained by *any* value in level $L_{i,j}$ of an attribute A_i as

$$I(S_{i,j}) = \frac{\sum_{v \in S_{i,j}} p'(v) \cdot I(v)}{\sum_{w \in S_{i,j}} p'(w)}$$

where $p'(v)$ is the weight of a value v and $I(v)$ is the information retained by v defined in Definition 5. Here the weight $p'(v)$ can be given by a user or simply be the number of leaves that appear in original table and are covered by v. Since a taxonomy tree may have an arbitrary shape, nodes at the same level may cover different number of leaves and retain different amount of information. Intuitively, $I(S_{i,j})$ is the average of the amount of information retained by a value

in $S_{i,j}$. Next, for every QI value q whose component $q[A_i]$ is at level L_{i,j_i} (that is, $q[A_i] \in S_{i,j_i}$) for $1 \leq i \leq d$, we define the average information retained by q as

$$I(< L_{1,j_1}, \ldots, L_{d,j_d} >) = \frac{1}{d} \sum_{i=1}^{d} I(S_{i,j_i})$$

Notice that different QI values may retain the same amount of information.

Definition 6: (**Order** $\succ_I$) The *level vector* of a QI value q is $v_q =< I(< L_{1,j_1}, \ldots, L_{d,j_d} >), L_{1,j_1}, \ldots, L_{d,j_d} >$, where L_{i,j_i} is the level of $q[A_i]$. A QI value q precedes another QI value q' (written $q \succ_I q'$) if $v_q > v_{q'}$ (that is, if there exists some $0 \leq h < d$, such that $v[i] = v'[i]$ for $0 \leq i \leq h$, and $v[h+1] > v'[h+1]$).

Example 5: Figure 3(a) shows the taxonomies of two attributes A and B and the lattice of QI values they define. We assume that A is more important than B and the weight of each base value is 1. Consider QI value $a0b1$. Since $a0$ is at level 0 in attribute A, $b1$ is at level 1 of attribute B, the set of values in A at level 0 has only one value $a0$, and the set of values in B at level 1 has two value $b1$ and $b2$, the information retained by $a0b1$ is $I(< 0, 1 >) = \frac{1}{2} \cdot (\frac{1 \cdot \frac{1}{2}}{1} + \frac{1 \cdot 1 + 1 \cdot 1}{1+1}) = 0.75$, and the level vector $v_{a0b1} =< 0.75, 0, 1 >$. Similarly, $v_{a1b1} =< 1, 1, 1 >$, $v_{a1b0} =< 0.75, 1, 0 >$, and $v_{a0b0} =< 0.5, 0, 0 >$, as shown in Figure 3(b). Thus, $a1b1 \succ_I a1b0 \succ_I a0b1 \succ_I a0b0$. Notice that $a1b0$ and $a0b1$ cannot be ordered by $\succ$(more general than), but can be ordered by $\succ_I$.

Theorem 2: For any two distinct QI values q_1 and q_2 that are more general than the same base QI value q, either $q_1 \succ_I q_2$ or $q_2 \succ_I q_1$.

Example 6: In Figure 3(a), both $a0b1$ and $a1b0$ are more general than $a1b1$, as shown in Example 5, $a1b0 \succ_I a0b1$.

Definition 7: (**Next EC**) Let q_0 be a base QI value of a base tuple t and $q \succ q_0$ be the QI value of the EC containing t. The QI value of next EC of t based on $\succ_I$ is q' such that $q' \succ q_0$, $q \succ_I q'$, and there exists no another QI value q'', such that, $q'' \succ q_0$ and $q \succ_I q'' \succ_I q'$.

Example 7: In Figure 3(b), if the EC of QI value $a1b0$ contains two tuples t_1 and t_2, where $t_1[QI] = a1b1$ and $t_2[QI] = a1b2$. Then, the next EC for t_1 is the EC of $a0b1$ and the next EC for t_2 is the EC of $a0b2$.

B. The Algorithm

Our algorithm is given in Figure 4. There are a total $H = \prod_{i=1}^{d} h_i$ items in V, where h_i is the height of taxonomy of attribute A_i. Initially, each item in V contains an empty list of ECs. At beginning, if the original microdata does not satisfy the (τ, ℓ)-diversity, the algorithm will generalize SA values to satisfy the requirement. It will then assign tuples to their initial ECs and sweep tuples one EC at a time. Some details are given in the following.

Theorem 3: The time complexity of Sequential Sweep Algorithm is $O((H + h_{SA}) \cdot n)$, where h_{SA} is the height of taxonomy of SA, $H = \prod_{i=1}^{d} h_i$, h_i is the height of the taxonomy of QI attribute A_i, and $n = |\mathcal{T}|$.

```
Input: a table T; a set of attribute taxonomies; a (τ, ℓ) requirement
Output: a table satisfying (τ, ℓ)-diversity
Method:
1. while T as an EC is not (τ, ℓ)-diversified do
2.     generalize SA value of a tuple of T;
3. let V be a sorted set of items <levelVector, ECList>;
4. for each tuple t in T do
5.     EC = findNextEC(t[QI], null, V); place t into EC;
6. closedECList=∅; count=|T|; currItem = first item of V;
7. while currItem is NOT the last in V and count > 0 do
8.     while ECList in currItem is not empty do
9.         currEC = removeFirst(ECList);
10.        while currEC is not (τ, ℓ)-diversified do
11.            remove a tuple t from currEC;
12.            nextEC= findNextEC(t[QI], currItem, V);
13.            put t into the nextEC;
14.        If currEC is not empty
15.            add currEC into closedECList;
16.            count = count - |curEC|;
17.    currItem = next item in V;
18. if ECList in currItem is not empty
19.    currEC = removeFirst(ECList);
20.    while currEC is not (τ, ℓ)-diversified do
21.        generalize SA of a tuple in currEC;
22.    add currEC into closedECList;
23. return tuples in closedECList.
```

Fig. 4. Sequential Sweep Algorithm

C. Generalize SA Values

We select tuples for SA values generalization according to the following heuristic. To reduce the skewness of an SA distribution, we select a tuple whose SA value covers the dominant base SA value of the EC. If more than one candidate exists, we choose the tuple that has the least general SA value. The SA value of the selected tuple is replaced by its parent SA value. This step is repeated until the set of tuples satisfies the (τ, ℓ)-diversity requirement.

D. Select Tuples to Sweep

As long as an EC does not satisfy (τ, ℓ)-diversity and it is not the last EC, we sweep tuples one by one into other ECs. Let E be the EC, a be the dominant base SA value, and $f(a)$ be its induced frequency. According to Definition 2, the frequency of a induced by tuple t is $l(t[SA], a)$. If t is removed from this EC, the new induced frequency of a becomes $\frac{f(a) \cdot |E| - l(t[SA], a)}{|E| - 1}$. Thus, removing t can effectively reduce skewness of SA distribution only if $l(t[SA], a) > f(a)$. Consequently, our heuristic is to select the tuple whose removal maximizes the reduction of the induced frequency of the dominant base SA value.

E. Find Next EC

To locate the next EC for a given base tuple t in the current EC, the function findNextEC (in steps 5 and 12 of the algorithm) performs the following tasks. First, it finds the level vector $v_{t[QI]}$ of QI value of t. Then, it finds the first item subsequent to the current item whose level vector v satisfies the following conditions: $v[0] \leq currV[0]$, where $currV$ is the level vector of the current item, and for $1 \leq i \leq d$,

$v[i] \leq v_{t[QI]}[i]$, where $v[i]$ is the i^{th}component of vector v. Then, it finds a unique QI value q, such that, $q \succ t[QI]$ and $v_q = v$. The next EC should be the EC of q in the ECList of the item. If this EC does not already exist, a new EC will be created and inserted into the ECList.

IV. Experiments

We implemented the Sequential Sweep algorithm (SWEEP) and two existing ℓ-diversity algorithm: the Incognito-based (c, ℓ)-diversity [1] (CLD) and 1-D ℓ-diversity [7] (1DLD). Our experiments were based on Adult and Nursery datasets from the UCI Machine Learning Repository [11]. We used all 45,222 records of the Adult dataset and all 12,960 records of the Nursery dataset. The experiments were run on a PC with a 2.8 GHz processor and 1 GB RAM.

A. Privacy Protection

Since it is more general than other ℓ-diversity measures, we use (τ, ℓ)-diversity in this experiment to specify privacy requirement and to measure the privacy actually achieved. Since other ℓ-diversity measures do not match (τ, ℓ)-diversity perfectly, CLD and 1DLD often achieve a protection beyond what is required.

Definition 8: (**Excessive Protection**)Let $f_1, f_2, ..., f_m$ be frequencies of base SA values of an EC E. The excessive protection of E with respect to a (τ, ℓ)-diversity requirement is defined as $e(E) = \sum_{k=1}^{m} |\psi(k) - F(k)|$. The excessive protection of a partition $\mathcal{P}$ is $e(\mathcal{P}) = \min_{E \in \mathcal{P}}\{e(E)\}$.

Figure 5 shows excessive protection (on the y-axis) of the three algorithms for increasingly stronger (τ, ℓ)-diversity requirements (on the x-axis). To measure excessive protection of CLD and 1DLD for a given (τ, ℓ)-diversity, we run these algorithms with all possible values of ℓ (and c), and measure the excessive protection of each resulting dataset according to Definition 8. We report the minimum excessive protection in Figure 5. Notice that our measure of excessive protection favors CLD and 1DLD.

As shown in Figure 5, while all (τ, ℓ)-diversity requirements are eligible for SWEEP, some are not eligible for CLD and 1DLD (the curves show no result at those (τ, ℓ) for which the corresponding algorithms output an empty table.) Actually, there are many more (τ, ℓ)-diversity requirements than shown in Figure 5 for which CLD and 1DLD can only output an empty table. For those requirements that are eligible to all the three algorithms, SWEEP results in a much less excessive protection than CLD and 1DLD do.

B. Utility and Accuracy

In this experiment, we compared the utility of anonymous data produced by the three algorithms. In Figure 6, the utility is measured with the information-based measure given in Definition 5 and the (τ, ℓ)-diversity requirements are the same as those in Figure 5. For (τ, ℓ)-diversity requirements that CLD and 1DLD do not output non-empty tables, the corresponding curves do not show any result. As shown by Figure 6, for both Adult and Nursery datasets, SWEEP often achieves much higher utility than 1DLD and CLD do. Also, 1DLD achieves higher utility than CLD does. Notice that, we only considered the data produced by 1DLD and CLD that had the least excessive protection. Otherwise, the utility of these algorithms will be much worse than what is shown here.

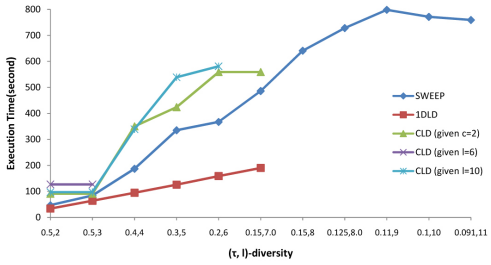


Fig. 8. Execution Time of Algorithms Relative to That of 1DLD

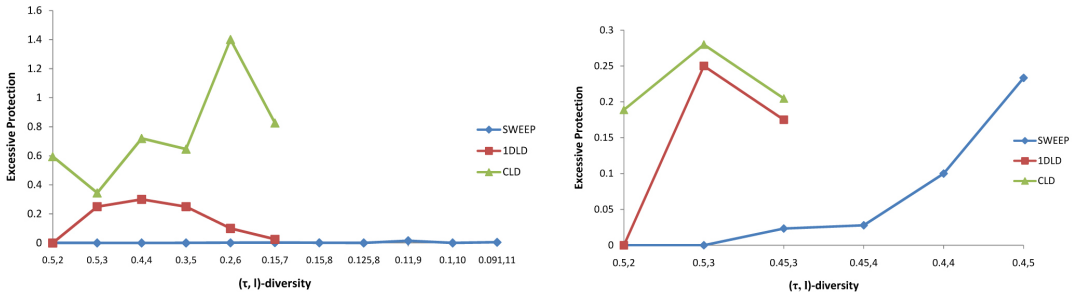
We also measured utility based on classification accuracy of decision trees learned using ID3 algorithm from anonymous data produced by the three algorithms. The utility measure in Figure 7 is the ratio of the classification accuracy of decision trees learned from the anonymous data over the classification accuracy of decision trees learned directly from the microdata. Again, for both Adult and Nursery datasets, SWEEP exhibits better utility than CLD and 1DLD.

C. Execution Time

In this experiment, we considered the execution time of the three algorithms. For CLD and 1DLD, finding the best anonymous data that satisfies a given (τ, ℓ)-diversity often requires a brute force search that runs the algorithms for different ℓ (and c). For example, for 1DLD the search has to start from $\ell = 2$ and try each subsequent ℓ until it output a table that satisfies the given (τ, ℓ)-diversity. The situation is much worse for CLD because the search space is 2-dimensional (c and ℓ) without a total order. This brute force search definitely has a negative impact on the performance of 1DLD and CLD. Although we may theoretically determine the weakest simple ℓ-diversity and (c, ℓ)-diversity requirements that guarantees a given (τ, ℓ)-diversity, such "best" measures typically cause much worse loss of utility than that shown in Figures 6 and 7. As shown in Figure 8, for the (τ, ℓ)-diversity requirements that are eligible to all algorithms, execution time of SWEEP is less than three times of that of 1DLD, and is always less than that of CLD; for the (τ, ℓ)-diversity requirements that are only eligible to SWEEP, execution time of SWEEP increases linearly.

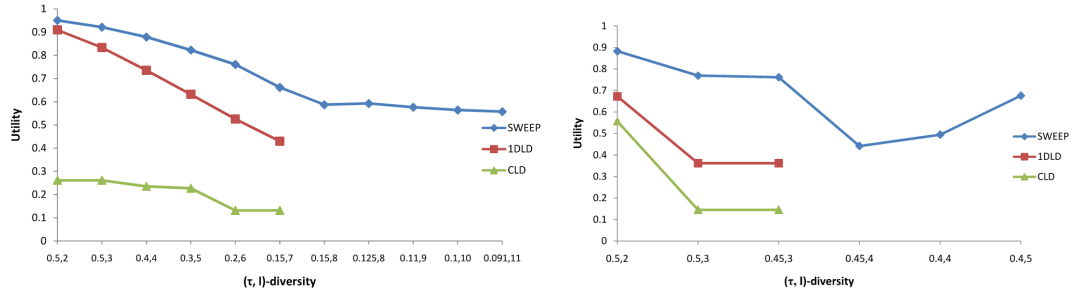
V. Conclusion

In this paper, we introduce a new measure of ℓ-diversity called the (τ, ℓ)-diversity, which extends ℓ-diversity in two ways. First, it allows the generalization of SA values. Second, it uses a function to explicitly specify constraints on frequencies of the first ℓ dominant induced SA values. We analyze the strength of various ℓ-diversity measures and give an efficient heuristic algorithm that uses a novel heuristic order of quasi-identifier values to obtain (τ, ℓ)-diversified anonymous data.



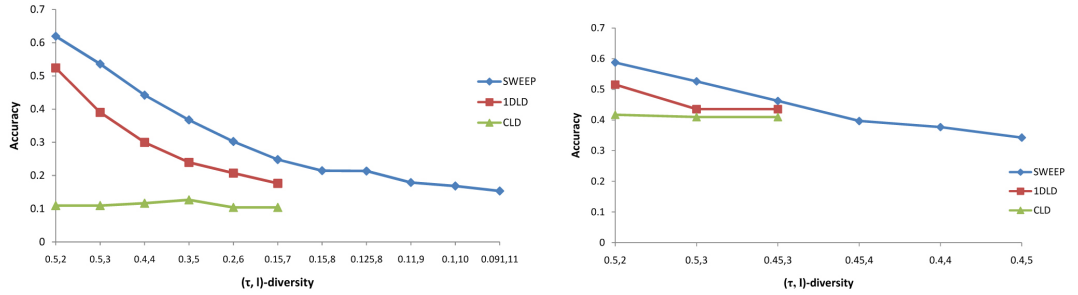
(a) Adult Dataset (b) Nursery Dataset

Fig. 5. excessive protection Produced by Algorithms



(a) Adult Dataset (b) Nursery Dataset

Fig. 6. Utility of Datasets Measured by Information



(a) Adult Dataset (b) Nursery Dataset

Fig. 7. Utility of Datasets Measured by Decision Tree Accuracy

We compare our algorithm with two state-of-the-art ℓ-diversity algorithms. Our preliminary experimental results indicate that our algorithm not only provides a stronger privacy protection but also results in better utility of anonymous data.

REFERENCES

[1] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *IEEE International Conference on Data Engineering*, 2006.

[2] Xiaokui Xiao and Yufei Tao. Anatomy: Simple and effective privacy preservation. In *International Conference on Very Large Data Bases*, 2006.

[3] Xiaokui Xiao and Yufei Tao. Personalized privacy preservation. In *ACM SIGMOD International Conference on Management of Data*, pages 229–240, 2006.

[4] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Incognito: Efficient fulldomain k-anonymity. In *ACM SIGMOD International Conference on Management of Data*, 2005.

[5] Ninghui Li and Tiancheng Li. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE International Conference on Data Engineering*, 2007.

[6] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *IEEE International Conference on Data Engineering*, 2006.

[7] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. Fast data anonymization with low information loss. In *International Conference on Very Large Data Bases*, pages 758–769, 2007.

[8] Yang Du, Tian Xia, Yufei Tao, Donghui Zhang, and Feng Zhu. On multidimensional k-anonymity with local recoding generalization. In *IEEE International Conference on Data Engineering*, 2007.

[9] Bee-Chung Chen, Kristen LeFevre, and Raghu Ramakrishnan. Privacy skyline: Privacy with multidimentional adversarial knowledge. In *International Conference on Very Large Data Bases*, page 770781, 2007.

[10] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Ke Wang, and Jian Pei. Minimality attack in privacy preserving data publishing. In *International Conference on Very Large Data Bases*, pages 543–554, 2007.

[11] The uci machine learning repository. http://mlearn.ics.uci.edu/MLRepository.html.