Anonymizing Temporal Data

Ke Wang[1], Yabo Xu[2], Raymond Chi-Wing Wong[3], Ada Wai-Chee Fu[4]
[1]Simon Fraser University, [2]Sun Yat-sen University
[3]The Hong Kong University of Science and Technology, [4]The Chinese University of Hong Kong
wangk@cs.sfu.ca, xuyabo@mail.sysu.edu.cn, raywong@cse.ust.hk, adafu@cse.cuhk.edu.hk

***Abstract* — Temporal data are time-critical in that the snapshot at each timestamp must be made available to researchers in a timely fashion. However, due to the limited data, each snapshot likely has a skewed distribution on sensitive values, which renders classical anonymization methods not possible. In this work, we propose the "reposition model" to allow a record to be published within a close proximity of original timestamp. We show that reposition over a small proximity of timestamp is sufficient for reducing the skewness of a snapshot, therefore, minimizing the impact on window queries. We formalize the optimal reposition problem and present a linear-time solution. The contribution of this work is that it enables classical methods on temporal data.**

I. INTRODUCTION

Imagine that a publisher wants to publish some microdata table D that contains *quasi-identifier* attributes QI={Age, Sex} and the *sensitive attribute* (SA) Disease. Suppose that a data recipient Alice, called the adversary below, wants to infer Bob's disease, knowing that Bob is a 26-year-old male and has a record in D. Alice identifies the records in D that match Bob's age and sex, and finds that 90% of matching records have a common disease, from there, concludes that Bob has that disease with a high probability. To eliminate such "homogeneity attacks", the *l-diversity principle* [7] requires that all records sharing the same values on QI must contain at least *l* "well-represented" SA values. One instantiation of this principle used in [9] is that the maximum relative frequency of a SA value in an A-group is $\leq 1/l$. In this work, *l*-diversity refers to this instantiation although other instantiations exist [7].

While most previous works dealt with transforming the original D into D' to satisfy *l*-diversity, few has examined the case that such transformation does not exist. In fact, a *l*-diversity transformation exists only if the maximum relative frequency of any SA value in the microdata D is $\leq 1/l$. This condition is called *eligibility* in [9] and *l-eligibility* in this paper. If D has a skewed frequency distribution on SA, *l*-eligibility will not be satisfied for a given *l* and no *l*-diverse transformation exists. As an example, suppose that H1N1 has the relative frequency of 50%, SARS has the relative frequency of 10%, and each of the remaining 10 diseases has the relative frequency 4% in D. For any $l>2$, *l*-eligibility condition is not satisfied, thus, no *l*-diversity transformation is possible.

A. Motivations

In this paper, we consider *temporal data* of the form D[1], D[2], …, where each D[i] is the snapshot containing the records collected at the time instance i. Examples of temporal data include sensor data, Internet traffic, financial tickers, web logs, fraudulent credit card transaction detection, network intrusion detection, financial record auditing, and telephone call records, inpatient discharge data, criminal reports, and population based disease monitoring data. Temporal data possesses the following time critical characteristics:

Timeliness*:* Each snapshot D[i] is collected and published at a fine resolution of time i. Thus, each D[i] has a small size, compared to data collected over a long period of time. For example, in tracking SARS cases, D[i] is collected and published daily because weekly or monthly collection does not provide detailed and timely information for SARS pattern analysis. ***Window queries****:* Temporal data is typically used to answer window queries for temporal pattern analysis [4]. A window query has the form $Q\wedge[a,b]$, where Q is a usual query predicate and $[a,b]$ is an ad hoc time interval. $Q\wedge[a,b]$ retrieves all records in $\cup_{a\leq i\leq b}D[i]$ that satisfy Q.

These characteristics lead to the following dilemma on privacy preservation for data publishing. On one hand, publishing each D[i] separately implies that the adversary could launch an attack on each D[i] based on the knowledge about timestamp i. To prevent such attacks, each D[i] must satisfy *l*-diversity. On the other hand, following the law of large numbers, a small D[i] tends to have a skewed distribution on SA, even though their union $\cup_i D[i]$ has a balanced distribution. In this paper, "temporal skewness" refers to such skewness of each D[i] due to small data size. This property implies that each D[i] often does not satisfy *l*-eligibility, thus, has no *l*-diverse transformation for a desired privacy level *l*. Our study on real life data sets confirmed these findings [13].

One solution is to generalize the time i as one additional attribute of the usual QI, essentially merging consecutive snapshots D[i] within a time interval. However, such generalized data does not meet the above requirements on timeliness and window queries because the fine resolution on time is lost. Another solution is to suppress some records in D[i] for high-frequency SA values until a desired *l*-eligibility is satisfied. This approach will suppress many records since small D[i] is naturally skewed on SA.

Several recent works considered data stream publishing and incremental publishing/re-publication, but none of them addressed the above requirements. [6] considers anonymizing data streams through random perturbation and their method applies to numeric data. [3][5] adopts *k*-anonymity to data streams whereas we consider *l*-diversity in order to prevent homogeneity attacks. [1][2][8][11] are concerned with correlation attacks arising from publishing a record in multiple timestamps. All these works require *l*-eligibility and limit queries to a single snapshot at a time. To our knowledge, [14] is the only work dealing with skewed distribution of SA, but it does not consider temporal data. The *t-closeness principle* [15] measures privacy threat by the distribution distance on SA between an anonymity group and the original data. Under this principle, even if all records in the original data are associated with the same disease, there is no privacy threat because there is no distribution distance. The *l*-diversity principle considers this case a privacy breach.

B. Contributions

Our contributions are as follows.

Temporal skewness We identify *temporal skewness* as a major challenge to privacy preservation in publishing temporal data for temporal analysis. Specifically, temporal skewness renders the traditional *l*-diversity transformation of D[i] non-existing for a desired privacy level *l* because the usually assumed *l*-eligibility is not satisfied.

Reposition model We propose a *reposition model* as a way to restore *l*-eligibility D[i] so that existing solutions can be applied to D[i]. Our observation is that neighbouring snapshots D[i] and D[i+Δ] (with a small Δ) often have "complementary" distribution on SA values because temporal skewness is due to the small size of each snapshot, not the inherently unbalanced distribution of SA. By repositioning some records across nearby D[i] and D[i+Δ] that have "complementary" SA values, it is often possible to restore *l*-eligibility of D[i] and D[i+Δ].

Optimal reposition solution We formalize the optimal reposition problem with respect to a cost metric and a reposition window size, and we present a linear-time solution to this problem. Our evaluation on real life data sets shows that often a small reposition distance Δ and a small amount of record suppression is all that is required to obtain an optimal solution. This finding leads to two important results: (1) repositioned data retains high utility for window queries; (2) the optimal solution obtained for a given reposition window size is likely an optimal solution when the reposition window size is infinite.

The novelty of this work is *enabling* existing solutions by restoring *l*-eligibility on D[i]. For example, by restoring *l*-eligibility of D[i], the bucketization scheme[9] and the optimal generalization/suppression scheme [12] can be applied to D[i]; the continuous publication and republication [1][2][8][11] can be applied to deal with temporal correlation of SA values. Note that these previous works cannot be directly applied to D[i] because they require *l*-eligibility on D[i], which is violated. Our work enables these existing solutions by first restoring *l*-eligibility on D[i].

The rest of the paper is organized as follows. Section II defines our privacy model and cost metric. Section III formulates the reposition problem. Section IV presents a relay model for reposition. Section V presents a linear-time optimal solution. Due to space limit, experimental evaluations are omitted, which can be found in [13]. Section VI concludes the paper.

II. PRIVACY AND REPOSITION MODELS

This section defines our privacy notion and reposition model. A temporal data D is a sequence D[1], D[2], …., where D[i] is the snapshot at timestamp i and each record in D[i] corresponds to an individual. As in [2][8], each individual has at most one record in the same D[i]. For a record r in D[i], r.TS denotes the timestamp of r, that is, r.TS=i. r.QI denotes the *quasi-identifier attribute* values and r.SA denotes the *sensitive attribute* value. |SA| denotes the number of distinct SA values in the domain of SA. We use D[i..j] to denote the subsequence D[i],D[i+1], …,D[j].

A. Privacy Model

For a set of records T, |T| denotes the number of records in T. |T[s]| denotes the number of records for the SA value s in T, called the *frequency* of s in T. Level(T) denotes the maximum |T[s]| for any SA value s. An anonymized version T* consists of *anonymity groups* or *A-groups*. For example, A-groups can be produced by generalization [7] or bucketization [9]. We use the following notion of *l*-diversity and *l*-eligibility used in [9].

Definition 1 An A-group g satisfies *l-diversity* if for every SA value s, $|g[s]| \le |g|/l$ (that is, the frequency of s in g is $\le |g|/l$). T* satisfies *l-diversity* or is *l-diverse* if every A-group g in T* satisfies *l*-diversity. T is *l-eligible* if Level(T) $\le |T|/l$; that is, the maximum frequency of any SA value in T is $\le |T|/l$. □

If T satisfies the *l*-eligibility condition, classical methods such as [7] and [9] can be applied to find an *l*-diverse transformation T*. However, these works did not consider the case that T does not satisfy the *l*-eligibility condition, thus, they cannot be applied to a snapshot D[i] that does not satisfy *l*-eligibility condition.

B. Overview

At each timestamp i, the snapshot D[i] arrives. Instead of publishing D[i], the publisher publishes an *l*-diverse version D'[i]. Over time, the recipient receives D'[1],…,D'[i] published so far, denoted by D'[1..i], and uses them to answer window queries. The problem is that D[i] is not necessarily *l*-eligible, thus, existing works fail to deal with such D[i]. To enable existing works, we focus on producing an *l*-eligible D'[i], and while doing so, minimizing the distortion to window queries. Prior to

publication, the *l*-eligible D'[1],…,D'[i] must be properly anonymized. In particular, to address the correlation of SA values of the individual in several snapshots, continuous publication or republication methods, such as [1][2][8][11], may be applied. The contribution of our work is enabling these great works by rendering D[i] into an *l*-eligible version.

To transform D[i] into *l*-eligible D'[i], we make use of a *reposition window* for record reposition. For a window of size n, at the time instant i+w the window holds the w latest snapshots D[i+1],..., D[i+w], or denoted by D[i+1..i+w]. By repositioning records currently in the window, we want to transform D[i+1..i+w] into D'[i+1..i+w] so that each D'[j] is *l*-eligible, i+1≤j≤i+w. At the next time instant i+w+1, the first snapshot D'[i+1] pops off the window and the next incoming snapshot D[i+w+1] joins the window at the right end. So the new window holds the old D'[i+2],..., D'[i+w], the new D[i+w+1]. Since D[i+w+1] is not necessarily *l*-eligible, reposition will be performed on the new window. The popped D'[i+1] is *l*-eligible, therefore, can be processed by an existing method.

The window size w represents the maximum delay of publication. Our study is that in most cases a record is repositioned over a small distance from its original timestamp in order to restore *l*-eligibility of D'[j]. This finding suggests that even if the window size is made infinite, the optimal solution would not be significantly better than that obtained under the constraint of a small reposition window size. Details are discussed in [13].

C. Cost Model

Our reposition model involves two types of operations, reposition and suppression. *Repositioning* a record from D[i] to D[j] refers to moving a record from D[i] to D[j]. Let D'[i] and D'[j] denote the resulting D[i] and D[j] after this reposition. For each record r in D'[j], r.TS' is defined as j, i.e., the timestamp after the reposition. |r.TS−r.TS'| denotes the *reposition distance* of r. We use a *cost function* μ(*d*) to model the information loss of this distortion, where *d*=|r.TS−r.TS'|. *Suppressing* a record r refers to withholding r from publication. The cost of suppressing a record for a SA value s is denoted by β_s.

As a result of reposition and suppression, each transformed D'[i] consists of a partition (P_i,S_i). P_i, the *eligible set*, is *l*-eligible and is for publication, and S_i, the *suppressed set*, will not be published. A suppressed record r *always* belongs to the home suppressed set S_i, where i=r.TS, independently of where it was suppressed from. For example, if r was originally in D[i], was repositioned to D'[j], and then was suppressed from D'[j], r should belong to S_i, not S_j. We use D'[i] and (P_i,S_i) interchangeably.

Suppose that D[L..U] is transformed to D'[L..U]. Recall that D[L..U] denotes the collection D[L],D[L+1],…,D[U]; similarly, D'[L..U]. Let us consider the cost of this transformation. Let D'[i] consists of (P_i, S_i), L≤i≤U. For each record r in P_i, the reposition cost is μ(|r.TS-r.TS'|), and for each record r in S_i, the suppression cost is β_s, where s is the SA value of r. The *information loss (IL)* of a record r in D'[i] is defined by

$$IL(r) = \begin{cases} \mu(|\, r.TS - r.TS'|), & r \in P_i \\ \beta_s & r \in S_i \end{cases} \qquad (1)$$

The *information loss* of D'[L..U] is

$$IL(D'[L..U]) = \sum_{r \in D'[L..U]} IL(r) \qquad (2)$$

Example 1 Figure 1 shows the computation of IL(D'[6..8]). Each record r_i is denoted by r_i:TS, where TS is the timestamp. P_6={r_1,r_2}, S_6={r_0}, P_7={r_3,r_4}, S_7=∅, P_8={r_5,r_6} and S_8=∅. Following Equation (1), IL(r_0)= β_s, and IL(r_5)= μ(8−7)=μ(1) because r_5 is repositioned from D[7] to D[8]. All other records have zero cost. IL(D'[6..8])= IL(r_0) + IL(r_5)= β_s + μ(1). □

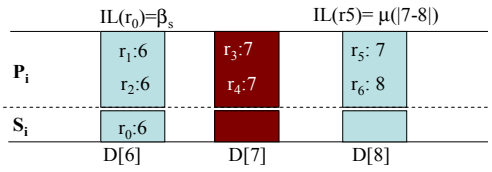

Figure 1. Computing IL(D'[6..8])

We consider a cost function μ satisfying the following three conditions.

(C1): μ(0)=0;
(C2): μ(d_1)>μ(d_2) if d_1> d_2≥0;
(C3): μ(d_1+Δ)−μ(d_1) ≥ μ(d_2+Δ) − μ(d_2), if d_1>d_2≥0 and Δ>0.

(C1) and (C2) are natural. (C3) models the non-linear deterioration of utility as the reposition distance increases. (C3) says that if two records r_1 and r_2 have been previously repositioned the distances d_1 and d_2, respectively, $d_1 > d_2$, and if we further reposition r_1 and r_2 by the same additional distance Δ, the cost increase for r_1 is no less than the cost increase for r_2. μ is *non-linear* if ≥ in (C3) is replaced with >, and μ is *linear* if ≥ in (C3) is replaced with =. For example, μ(d)=d^2 is a non-linear cost function and μ(d)=c×d is a linear cost function for any constant c.

The cost metric in Equation (2) models the cost pertaining to transforming D[i] to an *l*-eligible D'[i], which does not include the information loss for transforming an *l*-eligible D'[i] to an *l*-diverse D"[i]. We focus on the former because the latter is incurred by all classical methods. In a sense, we are interested in seeing how much extra cost must be paid to address the additional challenge brought up by temporal skewness.

III. THE PROBLEM

Without loss of generality, we assume that the reposition window holds D[1..w]. Suppose that we have an optimal solution D'[1],…,D'[w], denoted by D'[1..w], in the current window. Each D'[i], 1≤i≤w, consists of the pair (P_i, S_i) of the eligible set and suppressed set, and the cost

IL(D'[1..w]) is minimized. For the *very first* window, we can compute D'[1..w] by the more expensive integer linear program (ILP) with variables representing the number of SA values repositioned. For subsequent windows, however, ILP is too slow because ILP problems in general are NP-hard.

At the next time instant w+1, D'[1] is removed from the window and D[w+1] joins the window from the right end, so the new window holds D'[2..w] plus D[w+1]. D[w+1] is not necessarily *l*-eligible. Our task is to find a new solution D'[2],…,D'[w+1], denoted by D'[2..w+1], through reposition and suppression, such that IL(D'[2..w+1]) is minimized, and for $2 \le i \le w+1$, P_i is *l*-eligible. The notation in Table 1 will be used in the rest of the paper.

Table 1. Notation

$(P_2,S_2),\ldots,(P_w,S_w)$, (P_{w+1}, S_{w+1}) --- the initial solution.
$S_i[s]$ --- the set of records in S_i with SA=s.
$P_i[s]$ --- the set of records in P_i with SA=s.
$S[2..w+1]:s$ --- shorthand for $S_2[s],\ldots,S_{w+1}[s]$.
$P[2..w+1]:s$ --- shorthand for $P_2[s],\ldots,P_{w+1}[s]$.
$S_{max}[s]$ --- the nonempty $S_i[s]$ with the largest i.

To formalize the notion of optimality for our solution, we first motivate our reposition model used. Suppose that we can obtain an "initial solution" by suppressing a minimum number of records from D[w+1] until *l*-eligibility is achieved. This can be done by repeatedly suppressing a record for the most frequent sensitive value until *l*-eligibility is achieved. Let $D'[w+1]=(P_{w+1},S_{w+1})$ denote the set of remaining records and the set of suppressed records. The *initial solution* D'[2..w+1] consists of $(P_2,S_2),\ldots,(P_w,S_w)$, (P_{w+1}, S_{w+1}), where (P_i, S_i), $2 \le i \le w$, are inherited from the previous window. Note that P_i is *l*-eligible for $2 \le i \le w+1$.

Starting with the above initial solution D'[2..w+1] in the current window, we can reduce IL(D'[2..w+1]) while maintaining the *l*-eligibility of all P_i by repositioning records within the current window. In general, a record can be repositioned forward or backward in time. Under the assumption that the data is distributed randomly in time, a similar result would be expected if we consider reposition in only one direction. This observation prompts us to consider *forward reposition* where a record from D[i] is repositioned to D[j] with $i \le j$. Forward reposition models "publishing after the event", which is more natural than the backward reposition that models "publishing before the event". Since the first w-1 snapshots D'[2..w] in the current window were inherited from the optimal solution for the previous window, further reposition among D'[2..w] will not be effective. Therefore, we are left with two possible types of forward reposition:

Type I: Reposition of records from P[2..w] to P_{w+1}. Such reposition increases the reposition distance for the records in P[2..w] and impair the *l*-eligibility on P[2..w].

Type II: Reposition of records from S[2..w+1] to P_{w+1}. This type "recycles" previously suppressed records in S[2..w+1] for publication in P_{w+1}, and preserves the established *l*-eligibility on P[2..w] because P[2..w] is untouched. By recycling a suppressed record, the suppression cost of the record should be revoked. If the revoked suppression cost exceeds the cost of repositioning the suppressed record, the overall IL can be reduced.

Motivated by the above discussion, we consider a restricted form of reposition called recycle-reposition. *Recycle-reposition* refers to the forward reposition of records from P_i and S_i in such a way that $P_i[s]$ never changes for all SA values s and $2 \le i \le w$. As an example, if we first reposition a suppressed record r from S_i to P_j (thus, revoking the suppression) where $i \le j < w+1$, and then reposition a record r' in P_j to P_{w+1}, where r and r' have the *same* SA value, $|P_j[s]|$ is unaffected because the incoming r and the outgoing r' have the same SA value. Therefore, applied to the initial solution $(P_2,S_2),\ldots,(P_w,S_w)$, (P_{w+1}, S_{w+1}), recycle-reposition will preserve the *l*-eligibility of $P_2,\ldots,P_w$ and it is only necessary to check the *l*-eligibility of P_{w+1}.

Definition 2 (The optimal reposition problem) For a given reposition window size w, we want to obtain a solution D'[2..w+1] from the initial solution through recycle-reposition such that IL(D'[2..w+1]) is minimized and P_{w+1} is *l*-eligible. □

IV. TIMESTAMP ORDER PRESERVATION

A seemingly simple solution to the problem in Definition 2 is directly repositioning records from S[2..w+1] to P_{w+1}. Unfortunately, such reposition does not give an optimal solution because it violates "timestamp-order". We say that P[2..w+1] is *timestamp-order preserving* if for any r_1 in $P_i[s]$ and r_2 in $P_j[s]$ (i.e., r_1 and r_2 have the same SA value s), $i<j$ implies $r_1.TS \le r_2.TS$.

Theorem 1 For a non-linear cost function μ, *every* optimal solution is timestamp-order preserving. For a linear cost function μ, there is *some* optimal solution that is timestamp-order preserving.

Proof: It is always possible to remove a violation of timestamp-order by swapping the pair of records that causes the violation. The swapping reduces the cost for a non-linear μ and preserves the cost for a linear μ.□

In light of Theorem 1, we can focus on timestamp-order preserving P[2..w+1] without affecting optimality of solutions. This focus allows us to design a linear-time algorithm for finding an optimal solution. Observe that the initial solution P[2..w+1] described above is timestamp-order preserving because P[2..w] comes from the optimal solution for the previous window. To further reduce the cost by recycle reposition, we want to reposition a suppressed record r_0 in $S_i[s]$ to P_{w+1}, where $i<w+1$. To preserve the timestamp-order, we first reposition r_0 to P_{i_1} and then reposition r_1 "on behalf of r_0". Since r_0 and r_1 have the same SA value s, the frequency distribution of SA in P_{i_1} is

unchanged, so the *l*-eligibility of P_{i_1} is not affected. The relay model below formalizes this type of reposition.

The relay model Assume i<w+1. Let r_0 be a (suppressed) record in $S_i[s]$. First, we move r_0 from S_i to P_i. Let $P_{i_0}, \ldots, P_{i_k}$ be the *all* P_j with nonempty $P_j[s]$ such that $i=i_0 < \ldots < i_k=w+1$. For $1 \le p \le k$, let r_{p-1} be the record in $P_{i_{p-1}}[s]$ having the largest TS. Note that all these records r_{p-1} have the same SA value s. The *relay* of r_0 is the sequence of repositions:

$$P_{i_0} \xrightarrow{r_0} P_{i_1} \xrightarrow{r_1} P_{i_2} \xrightarrow{r_2} \ldots \xrightarrow{r_{k-1}} P_{i_k} \qquad (3)$$

Here, each step

$$P_{i_{p-1}} \xrightarrow{r_{p-1}} P_{i_p} \qquad (4)$$

repositions r_{p-1} from $P_{i_{p-1}}$ to P_{i_p}. The above relay preserves the timestamp-order by repositioning the record in $P_{i_{p-1}}[s]$ with the largest TS. Moreover, it does not change the frequency distribution of SA in P_i, $2 \le i \le w$, since whenever a record is moved into P_i, a record for the same SA value is moved out of P_i. The gain of the relay of a record $r=r_0$ is defined by

$$IG(r) = \beta_s - \psi(r). \qquad (5)$$

β_s is the suppression cost of r that now is revoked. $\psi(r)$ is the increase of reposition cost due to the reposition of all records r_{p-1} in Equation (4).

$$\psi(r) = \sum_{p=1}^{k} [\mu(i_p - r_{p-1}.TS) - \mu(i_{p-1} - r_{p-1}.TS)]. \qquad (6)$$

In each summed term, the two elements are the reposition cost of r_{p-1} before and after its reposition. Property (C2) and $i_p > i_{p-1}$ imply that each term in Equation (6) is positive.

V. THE ALGORITHM

We now present the complete algorithm for finding an optimal solution D'[2..w+1]. Starting from the initial solution D'[2..w+1] described in Section III, we greedily relay a set of suppressed records r from S[2..w+1] with the largest IG(r) to reduce IL(D'[2..w+1]). At the ith iteration, we relay a set of suppressed records M_i to raise Level(P_{w+1}) to the next level such that (1) *l*-eligibility of P_{w+1} is preserved, (2) $IG(M_i)>0$, and (3) $IG(M_i)$ is as large as possible. When this is not possible, we show that $IG(\cup M_i)$ is maximized. We apply two strategies alternately to find M_i.

A. Two Strategies

Strategy I: Level Preserving Reposition Strategy (*LP*-Reposition) This strategy greedily relays a set of records r with G(r)>0 from S[2..w+1] to P_{w+1} *without* increasing Level(P_{w+1}). This strategy preserves *l*-eligibility of P_{w+1} because it does not increase Level(P_{w+1}).

Example 2 (*LP*-Reposition) In Figure 2, the initial solution is on the right and S[2..w+1] is on the left. The records in S[2..w+1] are grouped according to the SA values $s_1,\ldots,s_5$. Records are written in the form r: (TS, IG), ordered by TS. Level(P_{w+1})=2. To preserve Level(P_{w+1}), we can relay one record in S[2..w+1]:s_4 and one record in S[2..w+1]:s_5. S[2..w+1]:s_4 contains one record r_8 and S[2..w+1]:s_4 contains one record r_9. Since $IG(r_8)>0$ and $IG(r_8)> IG(r_9)$, we relay r_8 at Line 2, as highlighted. In the next iteration, we fail to relay any record with positive IG without increasing Level(P_{w+1}). At this point, *LP*-Reposition stops.□

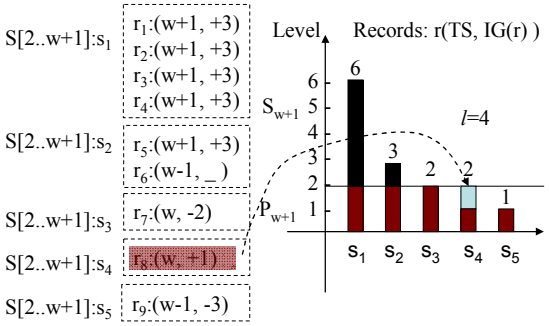

Figure 2. P_{w+1} after *LP*-Reposition

Strategy II: Level Lifting Reposition Strategy (*LL*-Reposition) This strategy further reduces IL by lifting P_{w+1} to *the next level* by relaying more than 1 records while preserving *l*-eligibility of P_{w+1}. We first calculate the minimum number of records to be repositioned into P_{w+1}, denoted as *m*. Assume Level(P_{w+1})=i prior to applying this strategy. In *m* iterations, this strategy searches for *m* records from S[2..w+1] for relay. In each iteration, it relays the record r from S[2..w+1] having largest IG(r) and satisfying $|P_{w+1}[r.SA]| \le i$. M^2 accumulates the records relayed. This strategy stops when either $|M^2|=m$ or no record can b relayed.

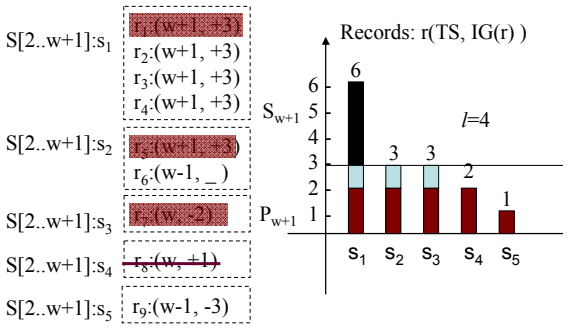

Figure 3. P_{w+1} after *LL*-Reposition

Example 3 (*LL*-Reposition) Continue with Example 2. At the end of *LP*-Reposition, Level(P_{w+1})=2 and P_{w+1} contains 9 records. To lift Level(P_{w+1}) to level 3 while satisfying 4-eligibility, P_{w+1} must contain at least 4×3 =12 records. So the *minimum* number of records that should be relayed into P_{w+1} is $m=12-|P_{w+1}|=3$. To maintain Level(P_{w+1})=3, for each SA value s_i, the *maximum* number of records that can be relayed into P_{w+1} is $m(s_i)= 3-|P_{w+1}[s_i]|$: $m(s_1)=1$, $m(s_2)=1$, $m(s_3)=1$, $m(s_4)=1$ and $m(s_5)=2$. So the problem becomes

finding m=3 records to maximize their total IG under the maximum number constraint $m(s_i)$ for each s_i. In our example, under the maximum number constraint $m(s_i)$, our pick of $m=3$ records is $\{r_1, r_5, r_7\}$ with a positive total IG=3+3−2=4. After relaying these records, the resulting P_{w+1} and S_{w+1} are shown on the right side in Figure 3. □

B. The Complete Algorithm

The complete algorithm *OptimalReposition* is given in Figure 4. It starts with the initial solution and relays records in S[2..w+1] in multiple iterations. In each iteration, it applies *LP*-Reposition and *LL*-Reposition to lift Level(P_{w+1}) to the next level. To raise P_{w+1} to the next level Level(P_{w+1})+1 while preserving l-eligibility, the minimum number of records that should be relayed into P_{w+1} is m =max{0, $(i+1)\times l-|P_{w+1}|$} because the minimum number of records for l-eligibility at level i+1 is $(i+1)\times l$.

Algorithm *OptimalReposition*
1. Let $(P_2,S_2),\ldots,(P_w,S_w)$, (P_{w+1}, S_{w+1}) be the initial solution;
2. **Repeat**
3. *LP-Reposition* ();
4. m=max{0, $(i+1)\times l-|P_{w+1}|$}, where i=Level(P_{w+1});
5. M^2 = *LL-Reposition* (m) ;
6. **Until** ($|M^2|<m$ **OR** $IG(M^2)\leq 0$);
7. Roll back the relay of the records in M^2;

Figure 4. The complete algorithm

The iterative process stops when either $|M^2|<m$ or $IG(M^2)\leq 0$, where M^2 is the set of records relayed by *LL*-reposition in the last iteration. This condition implies that *LL*-Reposition in the last iteration fails to lift P_{w+1} to the next level with a positive $IG(M^2)$. In this case, no further *LP*-reposition can relay any record either, because any such records have been picked up by the last *LL*-reposition to increase either $|M^2|$ or $IG(M^2)$. Line 7 rolls back the relays performed by the last *LL*-Reposition call.

Theorem 2 For the reposition window of size w, the work of *OptimalReposition* is bounded by $\sigma\times(w+|SA|)$, where σ is the number of suppressed records that are relayed in the current window. □

Proof: The detail is in [13].

Theorem 3 The *OptimalReposition* algorithm produces an optimal solution to the problem in Definition 2.

Proof: The key reason that the greedy algorithm actually produces an optimal solution is that, at any step, if we relay a suppressed record r having a SA value s, the relay does not affect the chance of relaying any suppressed record r' having a different SA value s', in terms of preserving the l-eligibility constraint on P_{w+1}, because $|P_{w+1}[s']|$ is unaffected by the relay of r. The detail is in [13]. □

VI. Conclusion

The time-criticalness of temporal data calls for the ability to anonymize sensitive values that have a more skewed frequency distribution. This challenges the classical methods that require the special l-eligibility condition. The contribution of this work is a novel method to restore the l-eligibility on temporal data; thus, this work enables classical methods on temporal data.

Acknowledgement: The research of Ke Wang is supported by a Discovery Grant from NSERC. The research of Yabo Xu is supported by Sun Yat-sen University Youth Scholar Grant. The research of Raymond Chi-Wing Wong is supported by HKRGC GRF 621309.

References

[1] Y. Bu, A. Fu, R. Wong, L. Chen and J. Li. Privacy preserving serial data publishing by role composition. In VLDB 2008.

[2] J. Byun, Y. Sohn, E. Bertino, and N. Li, Secure anonymization for incremental datasets. In SDM Workshop 2006.

[3] J. Cao, B. Carminati, E. Ferrari and K. Tan.. CASTLE: A delay-constrained scheme for k_s-anonymizing data streams. In ICDE 2008.

[4] Laxman and P. S. Sastry, A survey of temporal data mining, In SADHANA, Academy Proceedings in Engineering Sciences, The Indian Academy of Sciences, Vol 31, 2006

[5] J. Li, B. C. Ooi, and W. Wang. Anonymizing streaming data for privacy protection. In ICDE 2008.

[6] F. Li, J. Sun, S. Papadimitriou, G. Mihaila and I. Stanoi. Hiding in the crowd: privacy preservation on evolving streams through correlation tracking. In ICDE. 2007.

[7] A. Machanavajjhala, J. Gehrke, D. Kifer and M.Venkitasubramaniam. *l*-Diversity: Privacy beyond k-anonymity. In ICDE 2006.

[8] X. Xiao, and Y. Tao. m-Invariance: towards privacy preserving re-publication of dynamic datasets. In SIGMOD 2007.

[9] X. Xiao, and Y. Tao. Anatomy: simple and effective privacy preservation. In VLDB 2006.

[10] R. Wong, J. Li, A. Fu and K. Wang. (□, *k*)-Anonymity: An Enhanced *k*-Anonymity Model for Privacy-Preserving Data Publishing. In KDD 2006.

[11] B. Fung, K. Wang, A. Fu, J. Pei. Anonymity for continuous data publishing. In EDBT 2008.

[12] J. Liu and K. Wang. On optimal anonymization for l+-diversity. In ICDE 2010.

[13] K. Wang, Y. Xu, A. Fu, R. Wong. Anonymizing temporal data. Technical Report, School of Computing Science, Simon Fraser University 2010.

[14] Y. Xu, K. Wang, R. Wong, A. Fu. Publishing skewed sensitive microdata. In SDM 2010.

[15] N. Li, T. Li and S. Venkatasubramanian.. *t*-closeness: privacy beyond *k*-anonymity and l-diversity. In ICDE 2007.