Small Domain Randomization: Same Privacy, More Utility

Rhonda Chaytor, Ke Wang
Simon Fraser University
{rlc8, wangk}@cs.sfu.ca

ABSTRACT

Random perturbation is a promising technique for privacy preserving data mining. It retains an original sensitive value with a certain probability and replaces it with a random value from the domain with the remaining probability. If the replacing value is chosen from a large domain, the retention probability must be small to protect privacy. For this reason, previous randomization-based approaches have poor utility. In this paper, we propose an alternative way to randomize sensitive values, called *small domain randomization*. First, we partition the given table into sub-tables that have smaller domains of sensitive values. Then, we randomize the sensitive values within each sub-table independently. Since each sub-table has a smaller domain, a larger retention probability is permitted. We propose this approach as an alternative to classical partition-based approaches to privacy preserving data publishing. There are two key issues: ensure the published sub-tables do not disclose more private information than what is permitted on the original table, and partition the table so that utility is maximized. We present an effective solution.

1. INTRODUCTION

The objective of *privacy preserving data mining* (PPDM) is to discover interesting patterns in a given dataset T, without deriving any sensitive information. To achieve this goal, *input perturbation* [3][4][8][9][12][14][17][21] converts T to another dataset T* that permits effective PPDM, and at the same time, protects the sensitive information in T. *Random perturbation* [3][4][8][9] is a promising input perturbation method, due to strong privacy guarantees. We focus on a specific, widely adopted, random perturbation technique, called *Uniform Perturbation* [3].

Consider a sensitive attribute SA in T. For a given *retention probability* p, *Uniform Perturbation* retains the sensitive value x $\in$ SA in a record with probability p and replaces x with a random value chosen from the domain of SA with probability (1 – p). The value of p represents the tradeoff between the utility of the perturbed T* and the strength of privacy protection; when p = 1, all sensitive values are revealed (no privacy) and when p = 0, all sensitive values are completely randomized (no utility).

1.1 Notoriously Low Retention

Random perturbation was initially used for collecting sensitive binary survey results [22] and later extended to a categorical attribute with an arbitrary domain size [3]. As the domain size increases, the retention probability p diminishes in order to protect privacy. Consider a sensitive attribute SA with the domain size m. The probability that the original value x is replaced with a *specific* value y chosen from the *entire* domain of SA is q = (1 – p)/m, where (1 – p) is the probability that x will be replaced by *any* value from the domain of SA. To hide x, the total probability (p + q) that x remains unchanged should not be "much larger" than the replacing probability q, i.e., the ratio γ = (p + q) / q should be a "small" value. Solving these equations, we get q = $1/(m - 1 + \gamma)$ and p = $(\gamma - 1)/(m - 1 + \gamma)$.

In practice, these probabilities can be too small due to a large m. Consider the 8 discrete attributes of the CENSUS dataset (see Appendix 10.8 for details). Assuming ratio γ = 5, Table 1 shows the domain size m and the probabilities p and q for each attribute. Unless the domain size m is very small (< 10), the retention probability p is very low (< 30%), rendering the perturbed T* too noisy for data mining.

Table 1 Probabilities p and q for CENSUS, γ = 5

	A1	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*A8*
m	77	70	50	14	9	7	6	2
p	5%	5%	7%	22%	31%	36%	40%	67%
q	1%	1%	2%	6%	8%	9%	10%	17%

The above situation can get much worse in some practical scenarios. The standardized medical diagnoses ICD-9 codes (http://icd9cm.chrisendres.com) consist of over m = 15,000 different values. With γ = 5, q = 6.7×10^{-5} and (p + q) = 3.3×10^{-4}. In the case of multiple sensitive attributes, previous studies [4][24] suggest considering the "compound attribute" SA with a domain equal to the cross-product of sensitive attributes. For example, the compound attribute for *Age* and *Country* has domain size m = 77 × 70 = 5390. With γ = 5, q = 1.9×10^{-4}, (p + q) = 7.4×10^{-4}. In both cases, the uncertainty created by such a low retention probability is far more than what is required for privacy and the perturbed data is nearly completely random and useless.

The above analysis shows that low retention probability is caused by randomization over the entire domain of SA. To address this problem, the first thought is partitioning the domain of SA into disjoint subsets $dom_1,\ldots,dom_k$ and perturbing the records T_i for dom_i independently. However, this approach is dangerous if T_i has a more skewed distribution of sensitive values. Suppose `H1N1`, `SARS`, `HIV`, and `cancer` occurs in 40%, 30%, 29%, and 1% of records in T, respectively. After partitioning the domain into dom_1 = {`H1N1`, `cancer`} and dom_2 = {`SARS`, `HIV`}, `H1N1` occurs in 98% of records in T_1, compared to only 40% in T. The increased dominance of `H1N1` in T_1 leads to a similar increase in the perturbed data, which poses a larger threat to the records in T_1. The other partitioning has a similar problem. Thus, the approach of simply partitioning the domain of SA is *not* a solution.

Proceedings of the VLDB Endowment, Vol. 3, No. 1

1.2 Contributions

Instead of partitioning the domain of SA, the key to our approach is to partition the table T into disjoint sub-tables $T_1,...,T_k$ and perturb each T_i independently within SA_i, where SA_i denotes the subset of SA that occur in T_i. Since $T_1,...,T_k$ is a partitioning of T, SA_i is allowed to overlap. With a reduced domain SA_i, both the retention probability (p + q) and the replacing probability q for T_i will increase (more utility), whereas the ratio (p + q)/q will remain unchanged (same privacy). We refer to this approach as *Small Domain Randomization*. Our contributions are as follows.

Privacy We ensure that the adversary learns no more sensitive information from the perturbed sub-tables $T_1*,...,T_k*$, than what is permitted by the privacy requirement on T. We consider a restricted form of (ρ_1, ρ_2)-privacy [9] and we derive a new privacy requirement for each T_i such that enforcing the new privacy on T_i is sufficient for enforcing the given privacy requirement on T.

Utility We ensure partitioning $\{T_1,...,T_k\}$ minimizes the error of reconstructing SA's probability distribution in T. The challenge is that $T_1*,...,T_k*$ is a random instance and minimizing error for such a *specific* instance makes little sense. We derive a *probabilistic error bound* that holds with a given probability over *all* instances and search for a partitioning $\{T_1,...,T_k\}$ that minimizes this bound.

Algorithm Finding $\{T_1,...,T_k\}$ described above is a clustering problem with a global error metric under a privacy constraint. Such problems are unlikely to have an efficient optimal solution. We present a practical and efficient solution by employing several non-trivial techniques, namely, *distribution-aware partitioning*, *band matrix technique*, and *dynamic programming*.

Results We demonstrate that our algorithm's perturbed data can be used to answer *count queries* more accurately than traditional randomization and partition-based approaches. Such queries concern a subset of records satisfying a condition on some non-sensitive and sensitive attributes and are crucial for data analysis.

In the rest of this paper, Section 2 reviews related work, Section 3 provides background knowledge, Section 4 defines the problem, Section 5 describes our algorithm, Section 6 presents the experimental evaluation, and we conclude the paper in Section 7.

2. RELATED WORK

We study *input perturbation*, where publishing data is essential, as in most data mining applications, and perturbation is applied to the data to protect privacy. In contrast, *output perturbation* perturbs and publishes the answer to a query, where the data is not published. An example of output perturbation is query answering under *differential privacy* [16]. Input perturbation can be divided into two categories: partition-based and randomization-based.

Partitioned-based approaches partition the set of records into anonymity groups and release only some information about groups. These approaches include *generalization* [19][20], *anatomy* [13][23], and *condensation* [1]. Since the partitioning is deterministic and binds individuals to small anonymity groups, it is vulnerable to attacks when the adversary possesses additional background knowledge [5][14][21] on group members. The *corruption attack* [21] is an example.

Randomization-based approaches have been used in collecting survey results [22], in *privacy preserving data mining* [3][8][9], and more recently in *privacy preserving data publishing* [17][21][24]. Either random noise is added to numeric values [2], or categorical values are randomly replaced with other values in the domain [3][8][9][12]. Randomization-based approaches are less vulnerable to attacks like corruption attacks because each record is randomized independently and the non-determinism makes it more difficult to corrupt individuals with deterministic background knowledge.

Initial steps have been taken to find optimal randomization schemes for improving data utility [4][12][17]. However, all previous works perform randomization over the entire domain of the sensitive attribute, which leads to a low retention probability, as discussed in Section 1.1. To our knowledge, our work is the first to improve utility through randomization in small domains.

3. PRELIMINARY

The dataset T consists of one sensitive attribute SA and several non-sensitive attributes; multiple sensitive attributes can be treated as one compound sensitive attribute with a domain defined by the cross-product of all sensitive attributes. We assume that SA has the domain $\{x_1,\ldots,x_m\}$, or simply SA = $\{x_1,\ldots,x_m\}$. SA's domain size is |SA| = m and each x_i is called a *SA-value*. |T| denotes the number of records in T. The *frequency* of x_i refers to the number of records in T having x_i, and the *relative frequency* of x_i refers to the frequency of x_i normalized by |T|. As in [3][4][8][9][12][24], each SA-value x_i is chosen *independently at random* according to some fixed *probability distribution* denoted by p_X. The publisher allows the researcher to learn p_X, but wants to hide the SA-value of an individual record.

3.1 Uniform Perturbation

Like [3][4][8][17][21][24], we focus on *Uniform Perturbation*, because it maximizes retention probability [4]. In this perturbation scheme, SA-value x in record $r \in T$ is processed by tossing a coin with head probability p, called *retention probability*. If the coin lands on heads, x is retained in perturbed record r*; otherwise, x is replaced with a random value in SA in r*. Non-sensitive values are unchanged and T* contains all perturbed records r*, |T*| = |T|.

Let X and Y be random variables denoting the original and perturbed values, respectively. Both X and Y have domain SA. The probability of perturbing value $x \in SA$ to $y \in SA$ is given by:

$$\Pr[x \to y] = \begin{cases} p + (1-p)/m & if \quad x = y \\ (1-p)/m & if \quad x \neq y \end{cases} \qquad (1)$$

Recall m is the domain size of SA. In the case of x = y, p + (1-p)/m is the sum of the probability that x is retained and the probability that x is replaced with a specific value y from SA, where y happens to be equal to x. We refer to the set $\{\Pr[x \to y] \mid x, y \in SA\}$ as the *perturbation operator*, or *matrix*.

3.2 Privacy

(ρ_1, ρ_2)-privacy [9] Let Q(X) be a predicate on any sensitive value X in the original data T, Y be a perturbed version of X in the perturbed data T*, Pr[Q(X)] be the adversary's belief in Q(X) before observing Y = y (i.e., the *prior*), and Pr[Q(X) | Y = y] be the adversary's belief in Q(X) after observing Y = y (i.e., the *posterior*). The *(ρ_1, ρ_2)-privacy* in [9] states that

$$\Pr[Q(X)] \leq \rho_1 \text{ implies } \Pr[Q(X) \mid Y = y\,] \leq \rho_2,$$

where ρ_1 and ρ_2 are two constants in (0, 1] such that $\rho_1 < \rho_2$. In essence, (ρ_1, ρ_2)-privacy limits the increase of the degree of the adversary's belief after observing the published data. Notice (ρ_1, ρ_2)-privacy bounds the posterior Pr[Q(X) | Y = y] by ρ_2 only if the prior Pr[Q(X)] is not more than ρ_1.

A key requirement for ensuring (ρ_1, ρ_2)-privacy is that $Pr[x_k \rightarrow y]$ and $Pr[x_j \rightarrow y]$ for two distinct SA-values x_k and x_j should not differ "too much". This requirement is formalized by the following *γ-amplification condition* in [9], $\gamma \geq 1$: for all $y \in SA$,

$$\frac{\Pr[x_k \rightarrow y]}{\Pr[x_j \rightarrow y]} \leq \gamma, \forall j, k = 1,..., m \quad (2)$$

The next theorem shows that, for a "suitably small" γ value, the condition in (2) ensures (ρ_1, ρ_2)-privacy.

Theorem 1 [9] Assume that for every $y \in SA$, there exists $x_j \in SA$ such that $Pr[x_j \rightarrow y] > 0$. Suppose that the γ-amplification condition holds and

$$\gamma \leq \frac{\rho_2}{\rho_1} \times \frac{1-\rho_1}{1-\rho_2} \quad (3)$$

Then (ρ_1, ρ_2)-privacy is ensured.□

With Theorem 1, we can derive the optimal *Uniform Perturbation* matrix for ensuring (ρ_1, ρ_2)-privacy. From Equation (1), if x = y, $Pr[x \rightarrow y] = p + q$, and if $x \neq y$, $Pr[x \rightarrow y] = q$, where $q = (1 - p)/m$. Since $(p + q) \geq q$, Equation (2) reduces to $(p + q) / q \leq \gamma$, and to maximize (p + q), let $(p + q) / q = \gamma$. Solving these equations, we get

$$q = 1 / (m - 1 + \gamma) \text{ and } p = (\gamma - 1) / (m - 1 + \gamma). \quad (4)$$

With Equation (4), we rewrite the matrix defined in Equation (1) as the following $m \times m$ matrix:

$$P = \frac{1}{m-1+\gamma} \begin{bmatrix} \gamma & 1 & \cdots & 1 \\ 1 & \gamma & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & \gamma \end{bmatrix} \quad (5)$$

Let both x_i and y_i be the i^{th} value in SA, where x_i occurs in T and y_i occurs in T*. Each entry P[j][i] stores $Pr[x_i \rightarrow y_j]$, for i, j = 1,…,m. To ensure (ρ_1, ρ_2)-privacy, Theorem 1 suggests that the maximum γ value satisfying Equation (3) is given by

$$\gamma = \frac{\rho_2}{\rho_1} \times \frac{1-\rho_1}{1-\rho_2} \quad (6)$$

From the above derivation, the matrix P in Equation (5) with the γ value in Equation (6) ensures (ρ_1, ρ_2)-privacy. This P looks the same as the *gamma-diagonal matrix* in [4], which was shown to maximize the retention probability [4]. However, there is one important difference: the randomization in [4] is over the domain defined by the cross-product of all attributes, whereas the randomization defined by Equation (5) is over the domain of SA. Thus, the domain size m for the *gamma-diagonal matrix* is much larger the domain size m in Equation (5), which means that the retention probability in [4] is much smaller.

Restricted (ρ_1, ρ_2)-privacy used in this paper We consider the restricted form of (ρ_1, ρ_2)-privacy in which the predicate Q(X) has the form X = x and x is a specific SA-value. Specifically, we want to ensure

$$Pr[X = x] \leq \rho_1 \text{ implies } Pr[X = x \mid Y = y] \leq \rho_2.$$

This restricted version assumes the adversary's goal is to infer an *individual* SA-value x, and the publisher wants to limit the posterior probability of such inferences. Note, the L-diversity principle [15] also aims to limit the inference of an individual SA-value. We will show that, under this restricted privacy notion, small domain randomization will provide the same level of privacy as the table-wise randomization approach. In the rest of the paper, the term (ρ_1, ρ_2)-privacy refers to this restricted form.

In the absence of further knowledge, X with distribution $p_X(x)$ is the best description of the adversary's prior knowledge. Therefore, we model the prior Pr[X = x] by $p_X(x)$, i.e., the relative frequency of x in T.

3.3 Reconstruction Model

We evaluate the utility of perturbed data T* for reconstructing the SA probability distribution $p_X(x)$ and for answering count queries. Let $F = <f_1,…,f_m>$ denote the frequencies of SA-values $x_1,...,x_m$ in T. Let $F^* = <f_1^*,…,f_m^*>$ denote the estimate of F reconstructed using T* and P. The *reconstruction error* of f_i^* is defined by $|f_i - f_i^*| / |f_i|$, i=1,…,m. We can estimate F* from T* and P as follows (see more details in Appendix 10.2). Let $o = <o_1,…,o_m>$ be the observed frequencies from a *specific* published instance of T*.

$$f_i^* = \frac{(m-1+\gamma)o_i - |T^*|}{\gamma - 1} \quad (7)$$

Let $\{A_1, …, A_d\}$ be a subset of all non-sensitive attributes, a_j be a value from the domain of A_j, j = 1,...,d, and x_i be a SA-value.

A *count query* has the form

$$Q: \texttt{SELECT COUNT(*) FROM T} \\ \texttt{WHERE } A_1 = a_1 \texttt{ AND } … \texttt{ AND } A_d = a_d \texttt{ AND SA} = x_i \quad (8)$$

In other words, each query counts the size of one group defined by a GROUP-BY query with $\{A_1,…,A_d, SA\}$ being the GROUP-BY list. To compute the estimate est of the query result, we consider the set of records, Res, from T* satisfying the condition $A_1 = a_1$ AND … AND $A_d = a_d$ as a sample of T* and apply the above reconstruction to Res. Specifically, we instantiate o_i with the frequency of x_i in Res and replace |T*| with |Res| in Equation (7). The estimated frequency f_i^* of x_i in the query condition is returned as the query answer. We can treat Res as a sample of T* because the randomization on SA is entirely independent of the non-sensitive attributes A_1, …, A_d; therefore, the randomization for Res follows exactly the same probability distribution as for T.

4. THE PROBLEM STATEMENT

As illustrated in Section 1.1, the retention probability for *Uniform Perturbation* in Equation (5) diminishes as the domain size m of SA increases. To boost the retention probability, we propose to partition the input T into disjoint sub-tables $T_1,…,T_k$ such that each T_i involves a small sub-domain SA_i of SA with the size m_i, and perturb each T_i independently with perturbation matrix P_i over SA_i (more details later). Since SA_i has a smaller size m, P_i has a larger retention probability. The researcher can reconstruct the probability distribution $p_X(x)$ for T by computing an estimate

est_j from T_j* and P_j, as described in Section 3.3, for each j=1,…,k. The estimate on T is the sum $\sum_j est_j$.

What properties must the partitioning $T_1,\ldots,T_k$ satisfy? Assuming a (ρ_1, ρ_2)-privacy requirement is specified on T by the publisher, two issues arise. First, a skewed SA distribution in a sub-table T_i may expose the records in T_i to a greater privacy risk than permitted by the given privacy requirement on T (see Section 1.1). Second, the reconstruction error defined in Section 3.3 is for a *given* instance of the perturbed data, but the perturbed instance actually published is *randomly* determined, thus, is unknown prior to randomization. Therefore, our partitioning problem must answer two questions:

Question 1: *What privacy requirement must be ensured on each T_i in order to ensure the given privacy requirement on T?*

Question 2: *What metrics should be used to quantify the utility of the partitioning $T_1,\ldots,T_k$?*

In the rest of this section, we answer these questions. Let $Pr_i[X = x]$ and $Pr_i[X = x \mid Y = y]$ denote the adversary's belief on X = x in T_i before and after seeing Y = y in T_i*, respectively.

4.1 Privacy Requirement

Recall that the (restricted) (ρ_1, ρ_2)-privacy requirement on T states that if $Pr[X = x] \le \rho_1$, $Pr[X = x \mid Y = y] \le \rho_2$. Since T_i's are disjoint and are perturbed independently, it suffices to enforce $Pr_i[X = x \mid Y = y] \le \rho_2$ for T_i, i = 1,…,k. Therefore, to ensure (ρ_1, ρ_2)-privacy on T, we can ensure a new (ρ_{1i}, ρ_2)-privacy on T_i, i=1,…,k, such that (a) $\rho_{1i} < \rho_2$ and (b) $Pr[X = x] \le \rho_1$ implies $Pr_i[X = x] \le \rho_{1i}$. To see this, suppose $Pr[X = x] \le \rho_1$. Our choice of ρ_{1i} implies $Pr_i[X = x] \le \rho_{1i}$, and then ($\rho_{1i}$, ρ_2)-privacy implies $Pr_i[X = x \mid Y = y] \le \rho_2$, as required. This discussion leads to the next definition.

Definition 1 (ρ_{1i}, ρ_2)-privacy on T_i *acts as* (ρ_1, ρ_2)-privacy on T if $\rho_{1i} < \rho_2$, and $Pr[X = x] \le \rho_1$ implies $Pr_i[X = x] \le \rho_{1i}$. □

Simply speaking, if (ρ_{1i}, ρ_2)-privacy on T_i acts as (ρ_1, ρ_2)-privacy on T, (ρ_{1i}, ρ_2)-privacy on T_i ensures (ρ_1, ρ_2)-privacy on T. Therefore, to ensure (ρ_1, ρ_2)-privacy on T, for i=1,…,k, we look for a ρ_{1i} such that $\rho_{1i} < \rho_2$ and for every SA-value x with $Pr[X = x] \le \rho_1$, $Pr_i[X = x] \le \rho_{1i}$. Among all such ρ_{1i}, we prefer the smallest one in order to maximize γ_i (Equation (6)), thus, maximize retention probability p_i (Equation (4)). This smallest ρ_{1i} is determined as follows. Let

$$SA' = \{x \in SA \mid \Pr[X = x] \le \rho_1\}$$
$$\rho_{1i} = \max\{\Pr_i[X = x] \mid x \in SA'\} \quad (9)$$

SA' is the set of SA-values x with $Pr[X = x] \le \rho_1$, that is, the set of SA-values for which (ρ_1, ρ_2)-privacy places the bound ρ_2 on $Pr[X = x \mid Y = y]$. ρ_{1i} is the maximum relative frequency of such values in T_i. To ensure (ρ_1, ρ_2)-privacy, it suffices to ensure (ρ_{1i}, ρ_2)-privacy because $Pr[X = x] \le \rho_1$ implies $Pr_i[X = x] \le \rho_{1i}$. With Definition 1 and this discussion, we have

Corollary 1 Let ρ_{1i} be defined in Equation (9). If $\rho_{1i} < \rho_2$, (i) (ρ_{1i}, ρ_2)-privacy on T_i acts as (ρ_1, ρ_2)-privacy on T, (ii) (ρ_{1i}, ρ_2)-privacy ensures that if $Pr[X = x] \le \rho_1$, $Pr_i[X = x \mid Y = y] \le \rho_2$. □

Therefore, given the partitioning $T_1,\ldots,T_k$ and ρ_{1i} defined in Equation (9), our privacy goal is to ensure (ρ_{1i}, ρ_2)-privacy on T_i, i=1,…,k. To this end, we perturb T_i using the $m_i \times m_i$ matrix perturbation matrix P_i defined in Equation (5), with m being replaced with m_i (the domain size of SA_i) and γ being given by

$$\gamma_i = \frac{\rho_2}{\rho_{1i}} \times \frac{1-\rho_{1i}}{1-\rho_2} \quad (10)$$

Example 1 Consider the following partitioning {T_1, T_2} of T = {12, 8, 6, 5, 4, 3, 0, 0, 0, 0}, where the numbers inside brackets indicate the frequency of SA-values $x_1,\ldots, x_{10}$ in T_i in order:

T_1: {12, 8, 6, 4, 4, 2, 0, 0, 0, 0}
T_2: { 0, 0, 0, 1, 0, 1, 1, 1, 1, 1}

|SA| = 10, |T| = 42, $SA_1 = \{x_1,\ldots,x_6\}$, $SA_2 = \{x_4, x_6,\ldots,x_{10}\}$. Notice that SA_1 and SA_2 are not disjoint. Suppose the publisher specifies (1/3, 2/3)-privacy on T. Since the relative frequency of all SA-values is no more than ρ_1, SA' = SA. The maximum frequency in T_1 = 12 and $|T_1|$ = 36, so ρ_{11} = 1/3. Similarly, ρ_{12} = 1/6. Let us derive P_1 and P_2 for T_1 and T_2 using Equations (5) and (6). For T_1, m_1 = 6 and $\gamma_1 = (\rho_2 \times (1 - \rho_{11})) / (\rho_{11} \times (1 - \rho_2)) = 4$. For T_2, m_2 = 6 and $\gamma_2 = (\rho_2 \times (1 - \rho_{12})) / (\rho_{12} \times (1 - \rho_2)) = 10$. Therefore

$$P_1[j][i] = \begin{cases} 4/9, \textit{ if } j = i \\ 1/9, \text{ otherwise} \end{cases} \qquad P_2[j][i] = \begin{cases} 2/3, \textit{ if } j = i \\ 1/15, \text{ otherwise} \end{cases}$$

If T is not partitioned, γ = 4, m = 10, and P[j][i] = 4/13 if j = i, and P[j][i] = 1/13, otherwise. As we can see, by partitioning T into T_1 and T_2, the retention probabilities on the main diagonal (i.e., for j = i) increase from 4/13 to 4/9 for T_1 and to 2/3 for T_2. □

4.2 Utility Requirement

The utility metrics in Section 3.3 are difficult to incorporate into the search for an optimal partitioning because we do not know the count queries in advance. Even for reconstructing the probability distribution $p_X(x)$, the reconstruction error defined in Section 3.3 is for a *specific* instance of T_1*,…,T_k*; minimizing this error is not meaningful because the published instance is randomly generated by our perturbation matrix. It makes more sense to minimize a *probabilistic error bound* that holds with a certain probability over *all* possible random instances generated by the perturbation matrix. We now develop this metric.

We first consider T and then the partitioning $T_1,\ldots,T_k$. Let Y_i be a random variable representing the event that record r_i in T has perturbed SA-value y after perturbation, $1 \le i \le n$, where n = |T|. Let $Y = Y_1 + \ldots + Y_n$ be the frequency of y in T*. The mean of Y, E[Y], is $\mu = E[Y_1] + \ldots + E[Y_n]$. According to *Chernoff bound* [6], the probability that the error of Y is larger than a fraction θ of μ is $\le 2\exp(-\mu\theta^2/4)$, i.e., $Pr[|Y - \mu| > \theta\mu] < 2\exp(-\mu\theta/4)$ [3]. This error bound is for the observed frequency on the *perturbed* T*; however, we are interested in an error bound for the reconstructed frequency on the *original* data T. Theorem 2 gives such a bound.

Theorem 2 Consider the data T and the perturbed T* produced by applying *Uniform Perturbation* in Equation (1) on T, with retention probability p. Let f be the frequency of a SA-value x in T and f* be the estimate of f using T*. For a allowable error ε and confidence level ($1 - \delta$),

$$\Pr\left[\left|\frac{f - f^*}{|T|}\right| < \varepsilon\right] > 1 - \delta \quad \textit{if} \quad |T| \ge \frac{4}{\varepsilon^2 p^2}\log\left(\frac{2}{\delta}\right)$$

(See Appendix 10.2 for proof). □

The tightest bound ε is obtained by taking the equality in the if-condition. Substituting p in Equation (4) into this equality, we get:

$$\varepsilon = \frac{a}{\sqrt{|T|}}\left(\frac{m}{\gamma - 1} + 1\right) \quad (11)$$

where $a = 2 \times (\log(2 / \delta))^{1/2}$. Now consider a partitioning Part = $\{T_1,...,T_k\}$ of T. Let m_i and γ_i on T_i be the counterparts of m and γ on T defined in Section 3. Adapting Equation (11) to T_i, we get the error bounds for T_i and the partitioning Part = $\{T_1,\ldots,T_k\}$:

$$\varepsilon_i = \frac{a}{\sqrt{|T_i|}}\left(\frac{m_i}{\gamma_i - 1} + 1\right), \quad \varepsilon(Part) = \sum_{i=1..k} \frac{|T_i|}{|T|} \times \varepsilon_i \quad (12)$$

ε(Part) is a probabilistic error bound, in that, minimizing this bound will minimize the error on a randomly generated instance with confidence level $(1 - \delta)$. The equation for ε_i reveals two interesting points: first, the ε_i is expressed explicitly in terms of $|T_i|$, m_i and γ_i, which are easily computed from T_i; second, ε_i linearly decreases as m_i decreases and γ_i increases, but decreases much slower as $|T_i|$ increases. We will exploit this relationship to find a good partitioning in Section 5.

4.3 The Problem

Definition 2 (Small Domain Randomization) Given a dataset T, a (ρ_1, ρ_2)-privacy requirement on T, and confidence level $(1 - \delta)$, find a partitioning $\{T_1,\ldots,T_k\}$ of T, such that (i) for i = 1,…,k, $\rho_{1i} < \rho_2$, where ρ_{1i} is defined by Equation (9), and (ii) $\varepsilon(\{T_1,\ldots,T_k\})$ defined by Equation (12) is minimized. □

From Corollary 1(ii), if $\rho_{1i} < \rho_2$, (ρ_{1i}, ρ_2)-privacy on T_i ensures (ρ_1, ρ_2)-privacy on T. Thus the condition (i) and (ρ_{1i}, ρ_2)-privacy on T_i together ensure the (ρ_1, ρ_2)-privacy on T. A summary of the notation used in this paper can be found in Appendix 10.1.

We note that for reconstruction purposes, SA_i and P_i will be made public. With such information, the adversary can infer $\Pr[X = x_1 \vee \ldots \vee X = x_k \mid Y = y] = 1$ for T_i where $SA_i = \{x_1,\ldots,x_k\}$ is the subset of SA-values that occur in T_i. We will discuss this disclosure in Appendix 10.3 and show that as far as protecting individual values x_i is concerned, small domain randomization provides the same level of protection as table-wise randomization.

5. THE ALGORITHM

We now present a solution to the Small Domain Randomization problem in Definition 2. This is a clustering problem with a global utility metric subject to a privacy constraint. Such problems are unlikely to have efficient optimal solutions because the number of plausible partitionings is too large for a large T. We propose an efficient solution that heuristically minimizes the global metric.

Given a (ρ_1, ρ_2)-privacy requirement on T, we want to find a partitioning Part = $\{T_1,\ldots,T_k\}$ of T such that ε(Part) is minimized and $\rho_{1i} < \rho_2$, where ρ_{1i} , i=1,…,k, is defined by Equation (9). Minimizing ε(Part) requires minimizing the error bound ε_i, thus, minimizing m_i and maximizing γ_i (Equation (12)). From Equation (10), γ_i is maximized if ρ_{1i} is minimized (for the fixed ρ_2). Therefore, our algorithm must find a partitioning $\{T_1,\ldots,T_k\}$ satisfying the following two requirements:

Requirement I: T_i contains as few distinct SA-values as possible, in order to minimize m_i. This requirement calls for partitioning the records according to the similarity of their SA-values.

Requirement II: the maximum relative frequency of an SA'-value in T_i is as small as possible, in order to minimize ρ_{1i}. Recall SA' defined in Equation (9) is the set of SA-values with a relative frequency $\leq \rho_1$ in T. This requirement calls for distributing the records for the same SA'-value among $T_1,\ldots,T_k$. We use the following terminology to express Requirement II.

Definition 3 (θ-balanced) Let R be a set of records and let θ be an integer > 0. R is θ-*balanced* wrt SA' if $f / |R| \leq 1/\theta$ for every SA'-value x, where f is the frequency of x in R. □

Consider $\theta = \lfloor |T| / f^m \rfloor$ where f^m is the maximum frequency of any SA'-value in T. T is θ-balanced wrt SA' (because $\lfloor |T|/f^m \rfloor \leq |T|/f^m$) and T is not θ'-balanced wrt SA' for θ' > θ. So θ represents the maximum balancing level of SA'-values in T. Requirement II requires that each T_i is θ-balanced wrt SA'. Intuitively, this means that each T_i is as balanced as T.

Our algorithm, called *Perturbation Partitioning* (*PP*), finds the partitioning Part = $\{T_1,...,T_k\}$ of T in three phases.

5.1 Phase 1: Balancing Phase

This phase partitions T into disjoint *initial groups* $\{g_1,\ldots,g_t\}$, where each g_j contains the fewest possible SA-values and is θ-balanced wrt SA' and $\theta = \lfloor |T| / f^m \rfloor$, as defined above. The purpose of this phase is to break T into θ-balanced groups that have a minimum number of distinct SA-values, so that they can later be merged according to the similarity of SA-values (Requirement I). For ease of presentation, we first consider the case of SA' = SA; so we only refer to SA in the discussion below.

The initial groups g_j are created iteratively as follows (see Figure 1). Initially, $T_0 = T$ and T_0 is θ-balanced wrt SA. In the j^{th} iteration, g_j is created by selecting h records for *each* of the θ most frequent SA-values from T_0. h is defined in Equation (12) and, as shown in Lemma 1, is the maximum number such that the remaining data $T_0 - g_j$ is θ-balanced. The purpose of maximizing h is twofold: (1) maximizing the number of records in g_i without increasing the number of distinct SA-values (small m_i implies small ε_i), and (2) minimizing the number of initial groups, which is a key factor in reducing the complexity of subsequent phases.

```
1. T0 ← T; j = 1;
2. While T0 ≠ ∅ do
3.   Let x1,…,xθ be the θ most frequent SA-
   values in T0;
4.   Compute h by Equation (12);
5.   If h = 0 then gj ← T0 else gj contains h
   records in T0 for each of x1,…,xθ;
6.   j++;
7.   T0 ← T0 − gj;
8. Return all gj;
```

Figure 1: Balancing Phase

Let μ_i denote the i^{th} highest frequency of SA-values in T_0, then

$$h = \begin{cases} \mu_\theta & \text{if} \quad \sigma(\mu_\theta) \geq \mu_\theta \\ \lfloor |T_0| / \theta - \mu_{\theta+1} \rfloor & \text{otherwise} \end{cases} \quad (12)$$

where $\sigma(v) = |T_0| / \theta - \max(\mu_1 - v, \mu_{\theta+1})$. $h \geq 0$ because $|T_0|/\theta - \mu_{\theta+1} \geq 0$ (T_0 is θ-balanced wrt SA). Now we show several important properties of this phase. First, generated groups are θ-balanced wrt SA; second, after each iteration, the remaining data is θ-balanced wrt SA for the next iteration and there are indeed h records for each of the θ most frequent SA-values in T_0; third, h is maximized.

Lemma 1 Let gj be the initial group created by the jth iteration of the balancing phase and let h be computed by Equation (13). (i) gj is θ-balanced wrt SA. (ii) If T0 is θ-balanced wrt SA before the jth iteration, T0 – gj is θ-balanced wrt SA and $h \leq \mu\theta$, and (iii) h is maximum such that (ii) holds. (See Appendix 10.5 for proof). □

Example 2 Continue with the dataset T and (1/3, 2/3)-privacy in Example 1. SA' = SA, $f^m = 12$, $|T| = 42$, and $\theta = \lfloor |T| / f^m \rfloor = 3$. Initially, $T_0 = T$. In the first iteration, $|T_0| = 42$, $\mu_\theta = f_3 = 6$, and $\mu_{\theta+1} = f_4 = 5$, so $\sigma(\mu_\theta) = 42/3 - \max(12 - 6, 5) = 8$, which is greater than $\mu_\theta = 6$, so we set h to 6. Therefore, g_1 contains 6 records for each of the first θ = 3 SA-values. Then we remove these records from T_0 and repeat the process. After five iterations, we generate the five initial groups:

g_1: {6, 6, 6, 0, 0, 0, 0, 0, 0, 0}
g_2: {4, 0, 0, 4, 4, 0, 0, 0, 0, 0}
g_3: {2, 2, 0, 0, 0, 2, 0, 0, 0, 0}
g_4: {0, 0, 0, 1, 0, 1, 1, 0, 0, 0}
g_5: {0, 0, 0, 0, 0, 0, 0, 1, 1, 1} □

In the case of SA' ⊂ SA, only a proper subset of SA has a relative frequency ≤ ρ_1 and only these values are required to have the posterior bounded by ρ_2. Therefore, the creation of initial groups only need to minimize the maximum relative frequency for the SA'-values. Extension details can be found in Appendix 10.7.

5.2 Phase 2: Rearranging Phase

Initial groups often are too small to perform random perturbation because small $|T_i|$ implies a large error bound ε_I; therefore, we have to merge them to minimize the error bound. Requirement I implies that merging groups should have common SA-values. In this phase, we first rearrange the initial groups $g_1,\ldots,g_t$ into a sequence $g_1',\ldots,g_t'$ so that adjacent groups share common SA-values as much as possible. Later in the next phase we merge adjacent groups to minimize ε(Part).

We model the rearranging problem as finding the *band matrix* [7] of a square and symmetric matrix. Informally, the band matrix of a symmetric matrix is a permutation of rows/columns so that the non-zero entries are clustered along the main diagonal as much as possible. First, we represent the initial groups $g_1,\ldots,g_t$ by the following t × m matrix A: A[i][j] = f, if $x_j \in g_i$, otherwise A[i][j] = 0, where a row corresponds to an initial group g_i and a column corresponds to a SA-value x_j, and the entry A[i][j] stores the frequency of x_j in g_i. We apply the band matrix technique to the symmetric matrix $B = A \times A^T$, as suggested in [18]. In our context, a band matrix corresponds to a rearrangement of rows (i.e., initial groups) such that adjacent rows share common SA-values as much as possible. We use the *Reverse Cuthill-McKee (RCM)* algorithm, a variation of the *Cuthill-McKee* algorithm [7]. A band matrix was recently used in [11] to group sparse set-valued data. Notice that the dimensionality of our matrix, t × m, is much smaller than the data cardinality |T|, as Phase 1 minimizes the number of initial groups t by maximizing the size of initial groups.

Example 3 Let A be the matrix for the initial groups $g_1,\ldots,g_5$ in Example 2. Applying RCM to $A \times A^T$ gives us the rearranged order g_1, g_3, g_2, g_4, g_5. □

5.3 Phase 3: Merging Phase

Finally, we merge adjacent initial groups in the sequence $g_1,\ldots,g_t$ returned by the rearranging phase in order to minimize ε(Part). We model the optimal merging by dynamic programming. Let g[i..j] denote the merged group $g_i \cup \ldots \cup g_j$, $i \leq j$, and |g[i..j]| denote the number of records in g[i..j]. A *merging* of $g_1,\ldots,g_i$ is a partitioning of g[1..i] of the form g[1..r_1] ,…, g[r_{j-1} + 1..r_j], g[r_j + 1..r_{j+1}],…, g[r_{k-1} + 1..i], where $1 \leq r_1 < r_2 < \ldots < r_{k-1} < i$. Let ε_O([1..i]) denote the minimum error bound for any partitioning produced by a merging of $g_1,\ldots,g_i$. Let ε(g[i..j]) be the ε_i defined by Equation (12) with T_i= g[i..j]. The following dynamic programming finds the optimal merging of $g_1,\ldots,g_t$ that has the minimum error ε_O([1..t]).

$$\varepsilon_O([1..1]) = \frac{|g[1..1]|}{|T|}\varepsilon(g[1..1])$$

For $1 < i \leq t$,

$$\varepsilon_O([1..i]) = \min \begin{cases} \min_{1 \leq r \leq i-1}\{\varepsilon_O([1..r]) + \frac{|g[r+1..i]|}{|T|}\varepsilon(g[r+1..i])\} \\ \frac{|g[1..i]|}{|T|}\varepsilon(g[1..i]) \end{cases} \quad (14)$$

The first case in Equation (14) selects one boundary point r and the second case selects no boundary point. The final partitioning $\{T_1,\ldots,T_k\}$ is determined by all boundary points $r_1 < r_2 < \ldots < r_{k-1}$.

Example 4 The optimal merging of the sequence g_1, g_3, g_2, g_4, g_5 in Example 3 is $\{T_1 = g_1 \cup g_3 \cup g_2, T_2 = g_4 \cup g_5\}$, which gives the partitioning in Example 1. □

5.4 Analysis

The next theorem summarizes some properties of the partitioning $\{T_1,\ldots,T_k\}$ produced by the *PP* algorithm.

Theorem 3 Let SA' = SA, $\{T_1,\ldots,T_k\}$ be the partitioning of T returned by *PP*, and $\theta = \lfloor |T| / f^m \rfloor$, where f^m is the maximum frequency of a SA-value in T. (i) T_i is θ-balanced wrt SA, and $\rho_{1i} \leq 1/\theta$. (ii) If $\rho_2 > 1/\theta$, $\rho_{1i} < \rho_2$. (See Appendix 10.6 for proof). □

(i) ensures that T_i is as balanced as T, which is good for maximizing γ_i (Equation (10)), therefore, minimizing the error bound ε_i (Equation (12)). The inequality $\rho_{1i} < \rho_2$ in (ii) is required by Definition 2. The question is how likely the condition $\rho_2 > 1/\theta$ holds. The answer is *as likely as* the gap $\rho_2 - \rho_1$ is greater than $1/\theta - f^m/|T|$. The latter is the gap created by the effect of the floor $\theta = \lfloor |T|/f^m \rfloor$, and such gap is typically small. The counterpart of Theorem 3 for the case of SA' ⊂ SA is given in Appendix 10.7.

Theorem 4 The time complexity of the *PP* algorithm is $O(t^2 \log t + t^2 m + n + m \log m)$, where n = |T|, m is the domain size of SA, and t is the number of initial groups produced by the balancing phase. (See Appendix 10.6 for proof). □

Our experiments show that the number of initial groups, t, is quite small on real life datasets (no more than 20). This is because the balancing phase aims to maximize the size of each initial group. Therefore, the *PP* algorithm is linear in the cardinality of T.

6. EXPERIMENTAL EVALUATION

6.1 Experimental Setup

We compare our *Perturbation Partitioning* (*PP*) algorithm against two competitors. The first is *Anatomy* (*Ana*) [23] (code downloaded from the author's website), a partition-based algorithm known to have lower error for count queries than generalization [23]. The second is the optimal *Uniform Perturbation* (*UP*) defined by Equation (5) without data partitioning, which is known to maximize retention probability for ensuring (ρ_1, ρ_2)-privacy [9]. Our code is written in C++ and all experiments were run on a Core(TM) 2 Duo CPU 3.00 GHZ PC with 4GB of RAM. For *PP*, we use $\delta = 0.05$ for the confidence level ($1 - \delta$) of the error bound defined in Section 4.2.

We use the CENSUS dataset (see Appendix 10.8 for more details) like previous work [23]. Specifically, we use datasets of varied cardinality |T| = 100k-500k: OCC-|T| and EDU-|T| denote datasets having |T| records, with SA = *Occupation* (m = 50) and SA = *Education* (m = 14), respectively. *Occupation* is a balanced dataset, while *Education* is more skewed. In other real datasets, m can be much larger, which leaves more room for the proposed small domain randomization to increase retention probability.

We consider the utility of answering *count queries*, a metric used in both partition-based [13][23] and randomization-based [17] approaches. For this purpose, we generate a *query pool* of count queries and report the average *relative error* of query estimates for queries that pass a selectivity s = 0.1%, 0.5%, 1%, where *selectivity* of a query is defined as the % of records satisfying the query condition. See Appendices 10.9 and 10.10 for more details.

6.2 Publishing the Balanced Data

The balanced OCC-|T| datasets have a maximum relative frequency of 7%. For (ρ_1, ρ_2)-privacy (used by *UP* and *PP*), to protect all SA-values, we set $\rho_1 = 1/13$, so SA' = SA, and we set ρ_2 = 1/6, 1/5, 1/4, 1/3. *Ana* uses the L-diversity [15] privacy requirement to limit the probability of inferring an SA-value by 1/L. We set L = 6, 5, 4, 3 for *Ana*, to match the above setting of ρ_2. Notice 1/L and ρ_2 = 1/L equally bound posterior probability. For this reason, we simply refer to L. Figure 2 shows the comparison of errors for various L (upper part) and |T| (lower part).

We observe that *PP* incurs less error than *Ana* and *UP*. As L decreases for *Ana* (thus, ρ_2 increases for *UP* and *PP*), *PP* benefits from a weaker privacy requirement, whereas *Ana* has little change. Also, *PP* benefits from a larger minimum selectivity *s* (more random trials available for reconstruction), whereas the same trend is less obvious for *Ana*. In fact, [10] shows *Ana*'s performance deteriorates for GROUP-BY queries. This study shows that *PP* is useful for aggregate data analysis.

To get a better understanding of the reason for the smaller errors of *PP*, Table 2 compares the retention probability of *UP* and *PP*, and Table 3 shows statistics for T_i produced by *PP*. The average retention probability for T_i in Table 2 is significantly larger than for T. As shown in Table 3, this is because the domain sizes m_i for T_i is much smaller than m = 50 for T, whereas γ_i for T_i is very close to $\gamma = 2.5$ for T. This characteristic is exactly what is required to minimize the error bound ε_i in Equation (12).

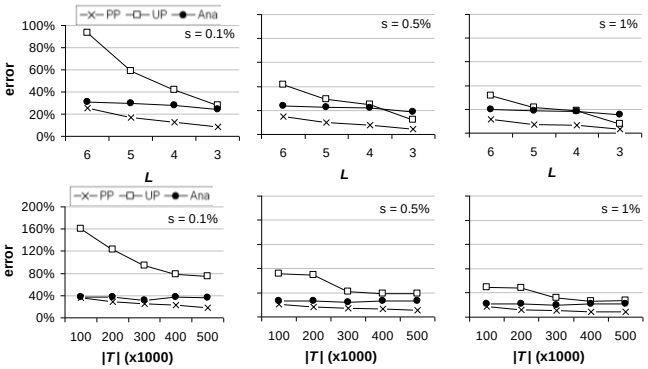

Figure 2. OCC: Error vs. L (|T|=300k), Error vs. |T| (L=6)

Table 2. Retention Probability, OCC-300k

L	6	5	4	3
UP	2.9%	4.0%	5.9%	9.4%
PP	9.0%	12.3%	17.3%	25.7%

Table 3. Statistics for *PP*, OCC-|T|, ρ_2 = 1/6

|T|	100K	200K	300K	400K	500K
# init. grps t	17	18	19	20	20
# of T_i*	15	8	10	11	12
avg. m_i	15.0	20.9	17.3	17	18.4
avg. γ_i	2.6	2.7	2.6	2.6	2.8

Figure 3 plots one point (act, est) for each query passing minimum selectivity s = 0.1%, where est and act are the estimated and actual query answers, respectively. The diagonal line act = est represents the perfect case of no error. *PP* has the best concentration of points near the diagonal line.

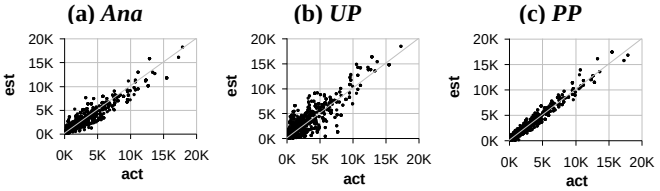

Figure 3. est vs. act: OCC-300k, L = 6, s = 0.1%

6.3 Publishing the Skewed Data

The skewed EDU-|T| datasets have a highest relative frequency = 27%, and 9 SA-values have relative frequency < 3.33%. Protecting all SA-values requires $\rho_1 \geq 27\%$, but such (ρ_1, ρ_2)-privacy is too weak to protect the less frequent SA-values since $\rho_2 > \rho_1$. It is more meaningful to protect the less frequent SA-values because the adversary has poorer prior knowledge on them. Therefore, we set $\rho_1 = 1/30$ to ensure the above 9 SA-values are protected by a tighter bound ρ_2 = 1/L, L = 10, 8, 6, 4. In this setting, SA' contains the 9 least-frequent SA-values.

The L-diversity privacy used by *Ana* requires that T satisfy the *eligibility condition* [23]: 1/L ≥ highest relative frequency of any SA-value in T. This condition is not satisfied by the above settings of L, therefore, *Ana* cannot be directly applied. This reveals a drawback of L-diversity and *Ana* in particular: they

cannot be applied on skewed data while providing sufficient protection for less frequent SA-values. To run *Ana*, we have no choice but to first suppress records with the most-frequent SA-values until the highest relative frequency in the remaining data is ≤ 1/L. The following refers to *Ana* with this pre-processing step.

Figure 4 shows the comparison of errors. In this experiment, *PP* degenerates into *UP* because the small domain size m = 14 make it unnecessary to partition T. *Ana* has a significantly larger error than *PP* and *UP* because record suppression leads to a significant under-estimation of query counts.

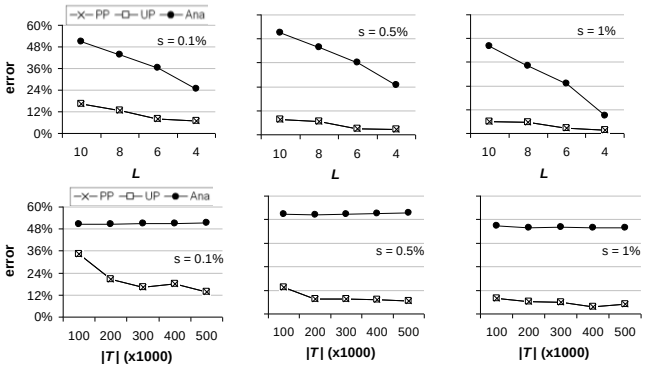

Figure 4. EDU: Error vs. L (|T|=300k), Error vs. |T| (L=10)

6.4 Perturbation Time

PP finishes in no more than 30 seconds for all cardinalities tested and is comparable to *Ana* and *UP* (see Figure 5). *Ana* is faster on EDU because pre-processing decreases the table size, at a heavy cost of a much larger error for count queries (see Figure 4).

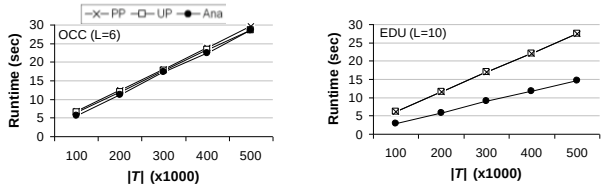

Figure 5. Runtime vs. |T|

7. CONCLUSION

Random perturbation has been extensively studied in the literature as an important technique for privacy protection; however, previous methods suffer from a notoriously low retention probability under most practical scenarios, due to "over-randomization" over the entire sensitive attribute domain. To address this problem, we proposed *small domain randomization*, which randomizes a sensitive value only within a subset of the entire domain. This approach retains more data while providing the same level of privacy. With improved utility, we proposed this approach as an alternative to classical partition-based approaches to privacy preserving data publishing. On CENSUS datasets, we observed a relative increase of over 100% in retention probability, compared to the optimal *Uniform Perturbation*. The higher retention probability translates into a relative decrease of over 200% in the reconstruction error for answering count queries.

8. ACKNOWLEDGMENTS

We would like to thank Drs. Tao and Xiao for answering questions on Anatomy and our reviewers for their helpful feedback. This research was supported by a Natural Sciences and Engineering Research Council of Canada (NSERC) Post Graduate Scholarship and Discovery Grant.

9. REFERENCES

[1] Aggarwal, C. C. and Yu, P. S. A condensation approach to privacy preserving data mining. *EDBT'04*.

[2] Agrawal, R. and Srikant, R. Privacy-preserving data mining. *SIGMOD'00*.

[3] Agrawal, R., Srikant R., and Thomas, D. Privacy preserving olap. *SIGMOD'05*.

[4] Agrawal, S. and Haritsa, J. A framework for high-accuracy privacy preserving mining. *ICDE'05*.

[5] Chen, B.-C., Ramakrishnan, R., and LeFevre, K. Privacy skyline: privacy with multidimensional adversarial knowledge. *VLDB'07*.

[6] Chernoff, H. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23, 4 (1952), 493–507.

[7] Cuthill, E. and McKee, J. Reducing the bandwidth of sparse symmetric matrices. In *Proc. of Nat. ACM Conf.*, 1969.

[8] Du, W. and Zhan, Z. Using randomized response techniques for privacy preserving data mining. *SIGKDD'03*.

[9] Evfimievski, A., Gehrke, J., and Srikant, R. Limiting privacy breaches in privacy preserving data mining. *PODS'03*.

[10] Ghinita, G., Karras, P., Kalnis, P., and Mamoulis, N. Fast data anonymization with low information loss. *VLDB'07*.

[11] Ghinita, G., Tao, Y., and Kalnis, P. On the anonymization of sparse high-dimensional data. *ICDE'08*.

[12] Huang, Z. and Du, W. OptRR: optimizing randomized response schemes for privacy-preserving data mining. *ICDE'08*.

[13] Koudas, N., Srivastava, D., Yu, T., and Zhang, Q. Aggregate query answering on anonymized tables. *ICDE'07*.

[14] Li, T. and Li, N. Modeling and integrating background knowledge. *ICDE'09*.

[15] Machanavajjhala, A., Gehrke, J., Kifer, D., and Venkitasubramaniam, M. l-diversity: privacy beyond k-anonymity. *ICDE'06*.

[16] McSherry, F. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *SIGMOD'09*.

[17] Rastogi, V., Hong, S., and Suciu, D. The boundary between privacy and utility in data publishing. *VLDB'07*.

[18] Reid, J. K. and Scott, J. A. Reducing the total bandwidth of a sparse unsymmetric matrix. *SIAM J. Matrix Anal. Appl.*, 28, 3 (2006), 805–821.

[19] Samarati, P. Protecting respondents' identities in microdata release. *TKDE*, 13, 6 (2001), 1010–1027.

[20] Sweeney, L. Achieving k-anonymity privacy protection using generalization and suppression. *Int'l J. on Uncert., Fuzz. and Knowledge-based Sys.*, 10, 5 (2002), 571–588.

[21] Tao, Y., Xiao, X., Li, J., and Zhang, D. On anti-corruption privacy preserving publication. *ICDE'08*.

[22] Warner, S. L. Randomized response: a survey technique for eliminating evasive answer bias. *The A. Stat. Assoc*., 60, 309 (1965), 63–69.

[23] Xiao, X. and Tao, Y. Anatomy: simple and effective privacy preservation. *VLDB'06*.

[24] Xiao, X., Tao, Y., and Chen, M. Optimal random perturbation at multiple privacy levels. *VLDB'09*.

10. APPENDIX

10.1 Table of Notations

Table 4 summarizes the notations used in this paper.

Table 4: Notations

SA - the set of SA-values for T; m = |SA|.
(ρ_1, ρ_2) - privacy parameters on T, specified by the publisher
SA' - the set of SA-values with $f / \|T\| \le \rho_1$ in T.
f^m – the maximum frequency of SA'-values in T.
T'– the set of records in T for SA'-values.
$\theta = \lfloor \|T\| / f^m \rfloor$ and $\theta' = \lfloor \|T'\| / f^m \rfloor$.
SA_i - the set of SA-values for T_i; $m_i = \|SA_i\|$.
(ρ_{1i}, ρ_2) – derived privacy parameters on T_i.
P_i – the perturbation matrix for T_i.
ε_i - the probabilistic error bound for T_i.
$Pr_i[A]$ – the probability of event A occurring on T_i

10.2 Reconstruction Error

Let $F = \langle f_1,\ldots,f_m \rangle$ denote the frequencies of SA-values $x_1,...,x_m$ in T. Let $F^* = \langle f_1^*,\ldots,f_m^* \rangle$ denote the estimate of F reconstructed using T* and P. All vectors are column vectors. The *reconstruction error* of f_i^* is defined by $|f_i - f_i^*| / |f_i|$, i=1,…,m. F* is reconstructed from T* and P as follows. Let O_i be the random variable representing the frequency of a SA-value y_i in T*, let $E(O) = \langle E(O_1),\ldots,E(O_m) \rangle$, where $E(O_j) = \sum_{i=1\ldots m} Pr[x_i \to y_j] \times f_i$ is the mean of O_j. Note $E(O) = P \times F$. The researcher has only a *specific* instance $o = \langle o_1,\ldots,o_m \rangle$ of $O = \langle O_1,\ldots,O_m \rangle$ observed on the published instance of T*, so the researcher resorts to the approximation $o = P \times F^*$. Applying P in Equation (5) to this equation and observing $f_1^*+\ldots+f_m^* = |T^*|$, we get

$$f_i^* = \frac{(m-1+\gamma)o_i - |T^*|}{\gamma - 1}$$

Notice that everything on the right hand side is directly obtained from T*, or Equation (6), or the domain size of SA. Thus, this reconstruction is extremely efficient because it does not require computing the inverse matrix P^{-1} [12] or any iterative computation as suggested in [3].

10.3 Disclosure in Our Privacy Model

For reconstruction purposes, SA_i and P_i will be made available to the researcher. With such information, the adversary can infer $Pr[X = x_1 \vee\ldots\vee X = x_k \mid Y = y] = 1$ for T_i where $SA_i = \{x_1,\ldots,x_k\}$ is the subset of SA-values that occur in T_i; that is, the original value must be one of $x_1,\ldots,x_k$. This kind of disclosure is not specific to our approach; in fact, all partition-based approaches have similar disclosures (e.g., [19][20][23]) in which SA_i is the set of sensitive values occurring in an anonymity-group. What is important is that the threat of guessing an *individual* value x_i is still bounded by $Pr[X = x_i \mid Y = y] \le \rho_2$, thanks to the restricted (ρ_{1i}, ρ_2)-privacy for T_i. As far as protecting individual values x_i is concerned, small domain randomization provides the same level of protection as table-wise randomization. This is why we consider the restricted form of (ρ_1, ρ_2)-privacy, where the predicate Q(X) has the form X = x.

We also note that our sub-tables T_i are different from anonymity groups in the classical partition-based approaches (e.g., [19][20][23]) in two major aspects. First, each T_i is much larger than an anonymity group because accurate reconstruction depends on having a large table size $|T_i|$ (Equation (12)). Second, the published version T_i^* is produced by randomizing the SA-value in each record in T_i and is robust to corruption attacks [21]. Since each record is randomized independently, even if the adversary succeeds in inferring the SA-value of some record in T_i, this information does not help to infer the SA-value of other records in T_i. In contrast, the SA-values associated with an anonymity group are not randomized; therefore, corrupting one record will lead to the increased threat of remaining records in the same anonymity group [21].

10.4 Proof of Theorem 2

Theorem 2 Consider the data T and the perturbed T* produced by applying *Uniform Perturbation* in Equation (1) on T, with retention probability p. Let f be the frequency of SA-value x in T and f* be the estimated frequency using T*. For an allowable error ε and confidence level $(1 - \delta)$,

$$\Pr\left[\left|\frac{f - f^*}{|T|}\right| < \varepsilon\right] > 1 - \delta \quad \text{if} \quad |T| \ge \frac{4}{\varepsilon^2 p^2}\log\left(\frac{2}{\delta}\right)$$

Proof: Let n = |T|. From Equation (1), the probability that the k^{th} row in T* has x, denoted $Pr[Y_k = 1]$, is $t = p \times f/n + (1 - p)/m$. Y_k's are independent Bernoulli random variables, so is $Y = \sum_k Y_k$, and the mean of Y is $\mu = nt$. Applying Chernoff bounds [6] from [3], we get

$$\Pr[|Y - \mu| > \theta\mu] < 2\exp(-\mu\theta^2/4).$$

Let $\varepsilon = \theta t/p$. $\theta\mu = \theta nt = n\varepsilon p$. With $t \le 1$, $2\exp(-n(\varepsilon p)^2/4) = 2\exp(-\mu t\theta^2/4) \ge 2\exp(-\mu\theta^2/4)$. For any $\delta \ge 2\exp(-n(\varepsilon p)^2/4)$, the above inequality implies

$$-n\varepsilon p < Y - nt < n\varepsilon p$$

with probability $\ge (1 - \delta)$. Notice that $\delta \ge 2\exp(-n(\varepsilon p)^2/4)$ is exactly the if-condition of the theorem.

Now let us rewrite $-n\varepsilon p < Y - nt < n\varepsilon p$ into the bound for the error of f*. Substituting $t = p \times f/n + (1 - p)/m$, we get

$$-n\varepsilon p < Y - n(pf/n + \frac{1-p}{m}) < n\varepsilon p$$

$$f/n - \varepsilon < \frac{1}{np}(Y - \frac{n(1-p)}{m}) < f/n + \varepsilon \qquad (15)$$

Let o be the observed frequency of x_i in T*. Applying Equation (1), we have the

$$o = f^* p + \frac{1}{m}\sum_{j=1}^{m} f_j^*(1-p) = f^* p + n\frac{1-p}{m}$$

$$f^* = \frac{1}{p}(o - \frac{n(1-p)}{m})$$

Instantiating the variable Y to the observed frequency o in Equation (15) and applying the above equation, we get $f/n - \varepsilon < f^*/n < f/n + \varepsilon$, as required. □

10.5 Proof of Lemma 1

Lemma 1 (SA' = SA) Let g_j be the initial group created by the j^{th} iteration of the balancing phase and let h be computed by Equation (13). (i) g_j is θ-balanced wrt SA. (ii) If T_0 is θ-balanced wrt SA' before the j^{th} iteration, $T_0 - g_j$ is θ-balanced wrt SA and $h \le \mu_\theta$, and (iii) h is the maximum such that (ii) holds.

Proof: If g_i is the last initial group, $g_i = T_0$, so g_j is θ-balanced wrt SA. If g_i is not the last initial group, g_j contains h records for each of the θ most frequent SA-values in T_0. Therefore, g_j is also θ-balanced wrt SA. This shows (i).

Now we show (ii). To show that $T_0 - g_j$ is θ-balanced wrt SA, we show $\max\{\mu_1 - h, \mu_{\theta+1}\} / (|T_0| - \theta \times h) \le 1/\theta$, where $\max\{\mu_1 - h, \mu_{\theta+1}\}$ is the maximum frequency of SA-values in $T_0 - g_j$ and $|T_0| - \theta \times h$ is the size of $T_0 - g_j$. This condition reduces to $h \le \sigma(h)$, where $\sigma(h) = |T_0| / \theta - \max(\mu_1 - h, \mu_{\theta+1})$. In the following, we show $h \le \sigma(h)$.

Consider the two cases for computing h in Equation (13). In the first case, $\sigma(\mu_\theta) \ge \mu_\theta$ and $h = \mu_\theta$, so $h \le \sigma(h)$. Also, $h = \mu_\theta$ is maximum such that $h \le \mu_\theta$, thus, (iii) is also shown for this case.

In the second case, $\sigma(\mu_\theta) < \mu_\theta$ and $h = \lfloor |T_0|/\theta - \mu_{\theta+1} \rfloor$. First, we show $h \le \mu_\theta$. $\sigma(\mu_\theta) < \mu_\theta$ implies

$$\max\{\mu_1 - \mu_\theta, \mu_{\theta+1}\} > T_0/\theta - \mu_\theta \qquad (16)$$

Suppose $\mu_1 - \mu_\theta > \mu_{\theta+1}$, Equation (16) becomes $\mu_1 - \mu_\theta > T_0/\theta - \mu_\theta$, which contradicts $\mu_1 \le T_0/\theta$ (the θ-balancing of T_0). Therefore, $\mu_1 - \mu_\theta \le \mu_{\theta+1}$, in which case Equation (16) becomes $\mu_{\theta+1} > T_0/\theta - \mu_\theta$, i.e., $\mu_\theta > T_0/\theta - \mu_{\theta+1} \ge \lfloor |T_0|/\theta - \mu_{\theta+1} \rfloor = h$. This shows $h \le \mu_\theta$. Finally, we show that $h = \lfloor |T_0|/\theta - \mu_{\theta+1} \rfloor$ is the maximum such that $\sigma(h) \ge h$. Let $\sigma(v) \ge v$, i.e.,

$$\max\{\mu_1 - v, \mu_{\theta+1}\} \le T_0/\theta - v. \qquad (17)$$

If $\mu_1 - v > \mu_{\theta+1}$, Equation (17) becomes $\mu_1 - v \le T_0/\theta - v$, which is trivial because T_0 is θ-balanced. Since we are interested in the largest v such that Equation (17) holds, we assume $\mu_1 - v \le \mu_{\theta+1}$. Then Equation (17) becomes $\mu_{\theta+1} \le T_0/\theta - v$, or $v \le T_0/\theta - \mu_{\theta+1}$. The largest v satisfying this condition is $v = h = \lfloor |T_0|/\theta - \mu_{\theta+1} \rfloor$. □

10.6 Analysis

Theorem 3 Let SA' = SA, $\{T_1,\ldots,T_k\}$ be the partitioning of T returned by *PP*, and $\theta = \lfloor |T| / f^m \rfloor$, where f^m is the maximum frequency of a SA-value in T. (i) T_i is θ-balanced wrt SA, and $\rho_{1i} \le 1/\theta$. (ii) If $\rho_2 > 1/\theta$, $\rho_{1i} < \rho_2$.□

Proof: From Lemma 1, every initial group created in the balancing phase is θ-balanced wrt SA'. T_i is the union of one or more initial groups, therefore, is θ-balanced wrt SA' due to the inequality $(f_1 + f_2) / (|g_1| + |g_2|) \le \max\{f_1 / |g_1|, f_2 / |g_2|\}$, where f_j is the frequency of a SA'-value in an initial group g_i. Therefore, $\rho_{1i} \le 1/\theta$. (ii) follows from $\rho_2 > 1/\theta$.□

Theorem 4 The time complexity of the PP algorithm is $O(t^2 \log t + t^2 m + n + m \log m)$, where $n = |T|$, m is the domain size of SA, and t is the number of initial groups produced by the balancing phase.

Proof: The balancing phase examines each record only once after SA-values are sorted initially, so it takes time $O(n + m \log m)$. In the rearranging phase, the matrix multiplication takes time $O(t^2 m)$ and the RCM takes time $O(t^2 \log t)$ [7], where t is the number of initial groups. The merging phase is dominated by the recursion in Equation (14), which takes time $O(t^2)$, since there are t different values of i. The overall time is $O(t^2 \log t + t^2 m + n + m \log m)$.□

10.7 Balancing Phase for SA' ⊂ SA

In the case of SA' ⊂ SA, only a proper subset of SA has a relative frequency $\le \rho_1$ and only these values are required to have the posterior knowledge bounded by ρ_2. Therefore, we only need to minimize the maximum relative frequency of such values in T_i. Let T' denote the set of records in T for the values in SA', and let T" = T − T'. First, we apply the balancing phase to T' to ensure that the distribution of SA'-values is balanced in the initial groups. Let $g_1,\ldots,g_t$ be the initial groups created. Then, we distribute the records in T" = T − T' to the initial groups proportionally to the size $|g_i|$: for each g_j, j = 1,...,t, distribute $\lfloor (|g_j|/|T'|) \times |T"| \rfloor$ records in T" to g_j. This proportional distribution ensures a minimum change of relative frequency of SA'-values in g_j. To minimize the number of distinct SA-values in each g_j, we first distribute all the records for the most frequent SA-value in T", then all the records for the second most frequent SA-values, and so on. We distribute any residue records to the last group g_t.

Recall that f^m is the maximum frequency of a SA'-value in T. f^m is also the maximum frequency in T'. Define $\theta' = \lfloor |T'|/f^m \rfloor$. As before, let $\theta = \lfloor |T|/f^m \rfloor$. Then T is θ-balanced wrt SA' and T' is θ'-balanced wrt SA'. The next lemma shows that the initial groups in the case of SA' ⊂ SA are "nearly" θ-balanced wrt SA'.

Lemma 2 (SA' ⊂ SA) Let g and g_a denote the corresponding initial groups before and after distributing the records in T". Let f be the frequency of a SA'-value in g and let $\alpha = 1 / (1 - (1/\theta'))$. If $\lfloor (|g|/|T'|) \times |T"| \rfloor = (|g|/|T'|) \times |T"|$, $f/|g_a| \le \alpha/\theta$, otherwise, $f/|g_a| \le \alpha/(\theta - \alpha)$.

Proof: Since g and T' are θ'-balanced wrt SA', $f \le |g|/\theta'$, $|g| \ge \theta'$, and $f^m/|T'| \le 1/\theta'$. Since T is θ-balanced wrt SA', $f^m/|T| \le 1/\theta$. Since $\theta' = \lfloor |T'|/f^m \rfloor$, $\theta' \ge |T'|/f^m - 1$, thus $1/\theta' \le f^m/(|T'| - f^m)$. Consider two cases.

- Case 1: $\lfloor (|g|/|T'|) \times |T"| \rfloor = (|g|/|T'|) \times |T"|$. In this case, $|g_a| = |g| + (|g|/|T'|) \times |T"| = |g|(|T|/|T'|)$. With the above inequalities,

$$\frac{f}{|g_a|} \le \frac{|g|/\theta'}{|g| \times |T|/|T'|} = \frac{|T'|}{|T|} \times \frac{1}{\theta'} \le \frac{|T'|}{|T|} \times \frac{f^m}{|T'| - f^m}$$
$$= \frac{f^m}{|T|} \times \frac{|T'|}{|T'| - f^m} \le \frac{1}{\theta} \times \frac{1}{1 - f^m/|T'|} \le \frac{1}{\theta} \times \frac{1}{1 - 1/\theta'} = \frac{1}{\theta}\alpha.$$

- Case 2: $\lfloor (|g| / |T'|) \times |T"| \rfloor \ne (|g| / |T'|) \times |T"|$. In this case, $|g_a| \ge |g| + (|g|/|T'|) \times |T"| - 1 = |g| \times |T|/|T'| - 1$. With the above,

$$\frac{f}{|g_a|} \le \frac{|g|/\theta'}{|g| \times |T|/|T'| - 1} = \frac{1/\theta'}{|T|/|T'| - 1/|g|} \le \frac{1/\theta'}{|T|/|T'| - 1/\theta'}$$
$$\le \frac{f^m/(|T'| - f^m)}{|T|/|T'| - f^m/(|T'| - f^m)} = \frac{1/(1 - f^m/|T'|)}{|T|/f^m - 1/(1 - f^m/|T'|)}$$
$$\le \frac{1/(1 - 1/\theta')}{\theta - 1/(1 - 1/\theta')} = \frac{\alpha}{\theta - \alpha}.$$

□

In the case of SA' = SA, Lemma 1 bounds the maximum relative frequency of a SA'-value in an initial group by $1/\theta$. In comparison, Lemma 2 gives the looser bounds α/θ or $\alpha/(\theta - \alpha)$, where $\alpha = 1 / (1 - (1/\theta'))$. For a large θ', α approaches to 1 and these bounds approach $1/\theta$ or $1/(\theta - 1)$. The next theorem is the counterpart of Theorem 3 for the case of SA' $\subset$ SA.

Theorem 5 Let SA' $\subset$ SA, $\{T_1,\ldots,T_k\}$ be the partitioning of T returned by the *PP* algorithm, and $\alpha = 1 / (1 - (1/\theta'))$. (i) $\rho_{1i} \leq \alpha/(\theta - \alpha)$. (ii) If $\rho_2 > \alpha/(\theta - \alpha)$, $\rho_{1i} < \rho_2$.

Proof: From Lemma 2, an SA'-value has a relative frequency $\leq \alpha/(\theta - \alpha)$ in an initial group. This remains true for a final group after the merging phase, for the same reason as in the proof of Theorem 3. Therefore, $\rho_{1i} \leq \alpha/(\theta - \alpha)$, so (i) is proved. (ii) follows from (i) and the assumption on $\rho_2 > \alpha/(\theta - \alpha)$. □

10.8 Description of CENSUS

The CENSUS dataset has 8 discrete attributes (domain size in brackets): *Age* (77), *Gender* (2), *Education* (14), *Marital* (6), *Race* (9), *Work-class* (7), *Country* (70), and *Occupation* (50). We used two datasets of varied cardinality |T| downloaded from [23]. OCC denotes the dataset with *Occupation* as SA and all other attributes as non-sensitive attributes. EDU denotes the dataset with *Education* as SA and the remaining attributes as non-sensitive attributes. OCC-|T| and EDU-|T| denote the samples of OCC and EDU with the size |T|, where |T| ranges over 100K,…,500K. Figure 6 shows that OCC-300K has a more balanced SA distribution, whereas EDU-300K has a much more skewed SA distribution. The choice of these datasets enables us to evaluate the utility for both balanced distribution and skewed distributions.

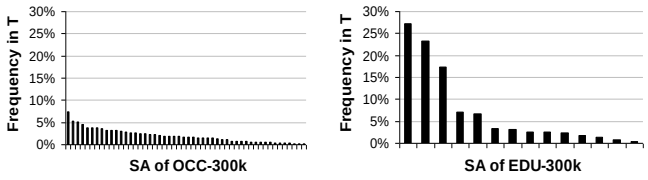

Figure 6. Frequency distributions

10.9 Generating the Query Pool

A *count query* has the following form

$$Q\text{: SELECT COUNT(*) FROM T WHERE } A_1 = a_1 \text{ AND } \ldots \text{ AND } A_d = a_d \text{ AND SA} = x_i \quad (18)$$

where $\{A_1, \ldots, A_d\}$ is a subset of non-sensitive attributes and a_j is a value from the domain of A_j, $j = 1,...,d$, and x_i is a SA-value. We generated a random *query pool* of count queries as follows. First, we created 200 random conditions of the form $A_1 = a_1$ AND … AND $A_d = a_d$. Specifically, we randomly select a value d from {1, 2, 3} (with equal probability), randomly sample d non-sensitive attributes $A_1,\ldots,A_d$ without replacement, and for each A_i, we randomly select a value a_i from A_i's domain. Then, for each of these 200 conditions, and for each of the m values x_i in the domain of SA, we generated a count query following the template in Equation (18).

10.10 Estimating Count Queries

In Section 3.3, we discussed how to estimate the answer for the count query Q in Equation (18) for *Uniform Perturbation*. Let us explain how to estimate the answer for *Perturbation Partitioning* and *Anatomy*.

Perturbation Partitioning Let $T_1^*,\ldots,T_k^*$ be produced by perturbing each sub-table T_i using P_i. An estimate est_j for each T_j^*, $j=1,\ldots,k$, can be computed as discussed in Section 3.3. The sum $\sum_j est_j$ is returned as the estimate for the query answer.

Anatomy Assume that the table T contains a set of non-sensitive attributes denoted QI and the sensitive attribute SA. *Anatomy* partitions T into anatomized groups, or *groups* for short, and publishes such groups in two tables. Let GID be the new attribute for storing the group identifier. The first table QIT contains all non-sensitive attributes and GID. The second table ST contains GID and SA. Suppose that a group g with GID = i contains the records $r_1,\ldots,r_k$. Then $(r_1[QI], i),\ldots,(r_k[QI], i)$ belong to QIT, and $(r_1[SA], i),\ldots,(r_k[SA], i)$ belong to ST. Let g(QIT) denote the set of records for g in QIT, and g(ST) denote the set of records for g in ST.

Given a query Q in Equation (18), a group g_i *matches* Q if some record in g_i(QIT) satisfies the query condition on the non-sensitive attributes in Q (i.e., $A_1 = a_1$ AND … AND $A_d = a_d$) and g_i(ST) contains the SA-value x_i. Let $g_1,\ldots,g_k$ be all the groups that match the query. Let $c(g_i, SA = x_i)$ be the count of x_i in g_i(ST) and let $c(g_i, A_1 = a_1,\ldots, A_d = a_d)$ be the number of records in g_i(QIT) satisfying the query condition on non-sensitive attributes. Then the query answer is estimated by

$$est = \sum_i c(g_i, A_1 = a_1,\ldots,6\ A_d = a_d) \times c(g_i, SA = x) / |g_i|.$$

See more details in [23].