Inference Analysis in Privacy-Preserving Data Re-publishing*

Guan Wang, Zutao Zhu, Wenliang Du, and Zhouxuan Teng
Department of Electrical Engineering and Computer Science
Syracuse University, Syracuse, NY 13244, USA
{gwang07, zuzhu, wedu, zhteng}@syr.edu

Abstract

Privacy-Preserving Data Re-publishing (PPDR) deals with publishing microdata in dynamic scenarios. Due to privacy concerns, data must be disguised before being published. Research in privacy-preserving data publishing (PPDP) has proposed many such methods on static data. In PPDR, multiple appeared records can be used to infer private information of other records. Therefore, inference channels exist among different releases. To understand the privacy property of data re-publishing, we need to analyze the impact of these inference channels. Previous studies show such analysis when data are updated or disguised in special ways, however, no general method has been proposed.

Using the Maximum Entropy Modeling method, we have developed a general solution. Our method can conduct inference analysis when data are arbitrarily updated or arbitrarily disguised using either generalization or bucketization, two most common data disguise methods in PPDR. Through analysis and experiments, we demonstrate the advantage and the effectiveness of our method.

1. Introduction

Privacy-preserving data publishing draws great attention of the community in recent years because of the concerns about privacy breaching issues in data publication process. To prevent linking attack, a primary attack in data publishing, quite a few PPDP methods have been proposed, including Bucketization [15], Generalization [3, 9, 11, 12], and Randomization [1, 2]. Most of them focus on static one-time dataset publishing and will disclose sensitive information when data is re-published. For example, Figures 1(a) and 1(c) depict a dataset D_1 and its updated version D_2. Records (7,10,13) are deleted while (14,15) are added. Figures 1(b) and 1(d) are the published versions (D'_1 and D'_2) using bucketization. Due to data disguising, neither D'_1 nor D'_2 itself provides adversaries absolute certainty on the disease of each patient. However, when putting them together, we know that Patient 10 has `Lung Cancer`.

Therefore, by studying two disguised data sets together, one can discover extra knowledge that is unavailable from each individual data set, even if each version of microdata is well disguised. Such extra knowledge among multiple disguised data sets is referred to as the *inference channels* [5]. Understanding how these inference channels affect the privacy of each individual in the published data sets is called *inference analysis*, which is a very challenging work. We use an example to show the difficulty.

Example 1 From the first two buckets in D'_2 of Figure 1(d), one cannot identify the diseases of Patients 14 and 15. However, with both disguised data sets D'_1 and D'_2, we can learn more. Let us assume that Patient 14 has Flu, then from D'_2, we can tell that Patient 1 and 2 should have $Pneumonia$ and $Diabetes$; therefore, from the first bucket of D'_1, we can tell that Patient 3 and 4 should both have Flu. This conclusion conflicts with the information in the second bucket of D'_2, because there is only one Flu in that bucket, and thus only one person among Patients 3, 4, and 15 in the bucket can have Flu. Therefore, our assumption on Patient 14 is incorrect, so the disease of Patient 14 should be either $Pneumonia$ or $Diabetes$, not Flu. We can conduct this kind of inference in a simple data set; however, when there are thousands of records in hundreds of buckets or QI groups, such an analysis becomes quite difficult.

Outline of Our Approach. The goal of inference analysis in PPDR is to find out potential privacy breaches from all the published data sets. From statistical perspective, inference analysis basically tries to identify if the inference channels among the published data sets can reduce the uncertainty of certain individuals' SA values to a level that can lead to privacy breaches. We model such uncertainty by a conditional probability $P(S \mid \mathcal{I})$. Therefore, if we can derive $P(S \mid \mathcal{I})$ for any $\mathcal{I}$ and S from all the published data sets, inference analysis becomes straightforward: simply examinate which conditional probabilities reach a dangerous level.

*This work was supported by Awards No. 0312366, 0430252, and 0618680 from the United States National Science Foundation.

Name	Pseudonym	Gender	Zip Code	Disease
Allen	1	male	13115	Flu
Brian	2	male	13120	Pneumonia
Cathy	3	female	13210	Diabetes
Daliy	4	female	13228	Flu
Ethan	5	male	13315	Flu
Frank	6	male	13471	Pneumonia
Grace	7	female	13520	Diabetes
Helen	8	female	13347	HIV
Irwan	9	male	13428	Flu
James	10	male	13451	Lung Cancer
Katey	11	female	13359	Pneumonia
Liman	12	male	13427	HIV
Milin	13	female	13530	Diabetes

(a) The original data set D_1

Pseudonym	Gender	Zip Code	Disease
1	male	13115	{Diabetes, Flu Flu, Pneumonia}
2	male	13120	
3	female	13210	
4	female	13228	
5	male	13315	{Diabetes, Flu Pneumonia}
6	male	13471	
7	female	13520	
8	female	13347	{Flu, HIV, Lung cancer}
9	male	13428	
10	male	13451	
11	female	13359	{Pneumonia, Diabetes, HIV}
12	male	13427	
13	female	13530	

(b) D_1': The disguised version of D_1

Name	Pseudonym	Gender	Zip Code	Disease
Allen	1	male	13115	Flu
Brian	2	male	13120	Pneumonia
Cathy	3	female	13210	Diabetes
Daliy	4	female	13228	Flu
Ethan	5	male	13315	Flu
Frank	6	male	13471	Pneumonia
Helen	8	female	13347	HIV
Irwan	9	male	13428	Flu
Katey	11	female	13359	Pneumonia
Liman	12	male	13427	HIV
Nikon	14	male	13119	Diabetes
Olice	15	female	13244	Pneumonia

(c) The original data set D_2

Pseudonym	Gender	Zip Code	Disease
1	male	13115	{Diabetes, Flu Pneumonia}
2	male	13120	
14	male	13119	
3	female	13210	{ Diabetes, Flu Pneumonia}
4	female	13228	
15	female	13244	
5	male	13315	{Flu, HIV Pneumonia}
8	female	13347	
11	female	13359	
6	male	13471	{Flu, HIV, Pneumonia}
9	male	13428	
12	male	13427	

(d) D_2': The disguised version of D_2

Figure 1. The data example used throughout this paper

Deriving a complete distribution of $P(S \mid \mathcal{I})$ from a single published data sets is not so difficult; most of the existing work on PPDP have provided methods to do so [12, 13, 15]. However, no existing work has shown how to derive $P(S \mid \mathcal{I})$ from multiple (related) published data sets. This task is quite challenging (see Example 1), especially when the original data sets can be arbitrarily updated and disguised.

We develop a general method to derive $P(S \mid \mathcal{I})$ in PPDR. We consider $P(S \mid \mathcal{I})$ for each combination of S and $\mathcal{I}$ as a variable. Each published data set provides certain "clues" about these variables. We model these clues as linear equations of these variables. We pool together the equations from all the published databases and try to find a solution to them. In most cases, the variables outnumber the equations, meaning that we will have many solutions for these variables; we have to choose one. Knowing that these variables represent probabilities, according to the *principle of maximum entropy* [10], the most unbiased solution is the one that maximizes the entropy. Therefore, we reduce our problem to a maximum entropy estimation problem, a well-established problem that has been extensively studied.

Related Work. PPDR problem has been studied by several groups recently [5, 6, 8, 16]. They focus on how to conduct PPDR to minimize the data disclosure risk. To be able to conduct inference analysis, they either put restrictions on how data can be updated, or on how data can be disguised. We lift these restrictions in our work. Our goal is not to propose another PPDR method; instead, we focus on providing a general inference analysis method for PPDR.

Wang and Fung propose another inference analysis approach [14]; it focuses on a different data re-publishing scenario. They suppose that data owners have a static data set with *QI* and *SA* values. Initially, the owners publish a subset of *QI* with *SA* values. Later, they release another subset of *QI* values, but without *SA* values. They show that in order to prevent inference attacks using the two releases, the second version should be anonymized properly.

In this paper, we use the maximum entropy (ME) estimation method to conduct inference analysis. The ME method has been used by Du et al. in integrating background knowledge in the quantification of privacy [7]. Although our proposed scheme also uses the ME method, we use it to solve a different problem.

2. Problem Formulation

Let D_i be a dataset to be published. We call $D_1, \ldots, D_t$ a series of *sequentially updated data sets* if D_i is an updated version of D_{i-1} through deletions and additions. We make some basic assumptions here:

We consider modification to the data set as a combination of deletion and addition. If the same person appears multiple times in different releases with different SA values, we treat them as different persons.

Definition 2.1 *(Privacy-Preserving Data Republishing) Let $D_1, \ldots, D_{t-1}$ be a series of sequentially updated data sets; they have already been published using certain*

data disguising schemes, such as generalization and bucketization. Let D_t be an updated version of D_{t-1}. The Privacy-Preserving Data Republishing (PPDR) problem is to publish the data set D_t in a way that satisfies the predetermined privacy requirements.

Definition 2.2 *(Inference Channel) Let S represent SA values and I be pseudonym. Let $P_u(S \mid I)$ be the conditional probability derived from a published data set D'_u, and Let $P_{u,v}(S \mid I)$ be the conditional probability derived from the published data sets D'_u and D'_v. If $P_u(S \mid I) \neq P_{u,v}(S \mid I)$, we say that there are inference channels between D'_u and D'_v. Finding inference channels or the impact of inference channels is called inference analysis.*

If we can derive all the conditional probabilities $P(S \mid \mathcal{I})$ for each person $\mathcal{I}$ and for each SA value S, we can accurately measure the privacy impact of data re-publishing. Therefore, we formulate the inference analysis problem as the following probability estimate problem:

Problem 2.1 *(Inference Analysis) Let D_1, ..., D_t be a series of sequentially updated data sets. Let D'_1, ..., D'_t be their respective disguised versions that have been published. Let $D = D_1 \cup D_2 \cup \ldots \cup D_t$ be the union of the original data set, and $D' = D'_1 \cup D'_2 \cup \ldots \cup D'_t$ be the union of the published data sets. Let variable S represent SA attributes, and let $\mathcal{I}$ represent the pseudonym attribute. Given D', derive $P(S \mid \mathcal{I})$ for all the combinations of $\mathcal{I}$ and S values.*

This is the main problem that we are going to solve in this paper. We have developed a general method to derive $P(S \mid \mathcal{I})$. Our method allows data to be arbitrarily updated. More importantly, our method allows data publishers to use any arbitrary combination of the two popular data disguise methods, bucketization and generalization; moreover, data publishers do not need to follow any specific pattern when they conduct data disguising. For the sake of simplicity, we only focus our discussions on bucketization in this paper, understanding that the same method that we have developed also applies to generalization.

3. A Complete Inference Analysis in PPDR

As we have discussed in the previous section, our inference analysis task is to derive $P(S \mid \mathcal{I})$ for each person $\mathcal{I}$ and each SA value S. In this section, we describe a systematic method to estimate $P(S \mid \mathcal{I})$ in data re-publishing.

3.1. Directly deriving $P(S \mid \mathcal{I})$

Deriving $P(S \mid \mathcal{I})$ for one-time data publishing is easy. For example, from the first bucket of Figure 1(b), We can easily infer that $P(Flu \mid \mathcal{I} = 1) = \frac{1}{2}$, because there are two out of four people in that bucket have `Flu`. [1]

In data re-publishing, we cannot use the same strategy to derive $P(S \mid \mathcal{I})$. Let us see an example. From Figure 1(d) alone, we can see that $P(Pneumonia \mid \mathcal{I} = 11) = \frac{1}{3}$, and $P(Flu \mid \mathcal{I} = 11) = \frac{1}{3}$, too. However, if we put these two published data sets together, we know that $P(Flu \mid \mathcal{I} = 11) = 0$, and $P(Pneumonia \mid \mathcal{I} = 11) = \frac{1}{2}$. Namely, without the first data set, we only know that $P(Pneumonia \mid \mathcal{I} = 11) = \frac{1}{3}$; however, with the information provided in the first data set, our inference can become more accurate.

In data re-publishing, anything published has a potential ability to affect the derivation of $P(S \mid \mathcal{I})$. When data sets are not so complicated, like the one in Figure 1, we might be able to figure out all the dependence among all the records, and derive $P(S \mid \mathcal{I})$ directly. However, when there are many versions of the published data, and when the bucketization (or generalization) used in different versions are quite different from one data set to another, deriving $P(S \mid \mathcal{I})$ directly becomes an infeasible task.

3.2. Indirectly deriving $P(S \mid \mathcal{I})$

We switch to an indirect approach to derive $P(S \mid \mathcal{I})$. For each combination of S and $\mathcal{I}$, we assign a variable to $P(S \mid \mathcal{I})$. Therefore, if we have m different S values, and n different people (i.e. $\mathcal{I}$ values), we have $m \cdot n$ different variables. We use a vector $\vec{x}$ to represent these variables.

These variables are not independent to each other. Actually, we can formulate their relationships as equations. For example, from the second bucket in Figure 1(b), we can derive the following: $P(Diabetes \mid \mathcal{I} = 5) * P(\mathcal{I} = 5) + P(Diabetes \mid \mathcal{I} = 6) * P(\mathcal{I} = 6) + P(Diabetes \mid \mathcal{I} = 7) * P(\mathcal{I} = 7) = \frac{1}{13}$, where 1 is the total number of $Diabetes$ and 13 is the total number of records in D'_1. We can formulate a number of equations like this; they are linear equations of the variables in $\vec{x}$.

If we can formulate all the existing knowledge from the published data sets as linear equations, deriving $P(S \mid \mathcal{I})$ basically becomes finding an assignment for all the variables in $\vec{x}$, such that all these linear equations are satisfied. Our task now becomes solving those linear equations.

Unfortunately, in most cases, we have more variables than equations, i.e., we will end up having many solutions. The question is which solution we should choose.

3.3. Using Maximum Entropy Principle

To decide which solution to choose, we have to step back and understand the meaning of these variables in $\vec{x}$. They are not arbitrary variables but probabilities. By solving these equations, we are trying to derive an *inference*

[1] Here, we assume that there is no background knowledge; with background knowledge, we cannot say that these four people have the same probability of having `Flu`.

for these probabilities. When deriving inference, the most important criterion that we need to follow is to be unbiased. Although there are many solutions for those equations, some are biased. Being biased means assuming some extra information that we do not possess; therefore, the least biased assignment is the most desirable [10].

This is the *Maximum Entropy (ME)* principle, applying which, our problem becomes deriving the distribution of $P(S \mid \mathcal{I})$, such that the following conditional entropy $H(S \mid \mathcal{I})$ is maximized:

$$H(S \mid \mathcal{I}) = -\sum_{\mathcal{I},S} P(\mathcal{I})P(S \mid \mathcal{I}) \log P(S \mid \mathcal{I}) \quad (1)$$

We assume that $\mathcal{I}$ is unique across the data set, and each person $\mathcal{I}$ only has one entry in the original data set (the same entry may be published several times, and thus appear multiple times in the published data). Therefore, $P(\mathcal{I})$ is a constant for all $\mathcal{I}$ values. As results, maximizing $H(S \mid \mathcal{I})$ is equivalent to maximizing the following:

$$-\sum_{\mathcal{I},S} P(S \mid \mathcal{I}) \log P(S \mid \mathcal{I}). \quad (2)$$

Without any constraint, $H(S \mid \mathcal{I})$ is maximized when $P(S \mid \mathcal{I})$ has an uniform distribution. However, the values of $P(S \mid \mathcal{I})$ are indeed subject to many constraints contained in the data sets. To apply the ME method, we need to convert all the available knowledge into equations (or inequalities) based on $P(S \mid \mathcal{I})$. Let these constraints be h_1, ..., h_w. Our problem is formally defined as the following:

Definition 3.1 *(Maximum Entropy Modeling) Finding an assignment for $P(S \mid \mathcal{I})$ for each combination of S and $\mathcal{I}$, such that the entropy $H(S \mid \mathcal{I})$ is maximized, while all the constraints $h_1, \ldots, h_w$ are satisfied.*

3.4. Constraints from each data set

To treat the inference analysis as a maximum entropy problem, we need to formulate all the knowledge that can be derived from the published data sets as linear equations. We refer to these equations as constraints. Before we describe the constraints, we define the following term:

Definition 3.2 *(Maximum Common Subset) Let $SA_i(id)$ represent the set of possible SA values that might be assigned to id from the published data set D'_i. In bucketization, this is the set of sensitive attributes contained in id's bucket. The Maximum Common Subset (MCS) of id is the intersection of $SA_1(id)$, ..., $SA_t(id)$.*

(a) Zero Constraints. $MCS(id)$ includes all possible SA values that are likely associated with this id. For any SA value s outside $MCS(id)$, the probability for this id to have s is zero. We call the following equation a *Zero constraint*:

$$P(s \mid \mathcal{I} = id) = 0, \text{ for } \forall s \notin MCS(id). \quad (3)$$

For example, from D'_1 in Figure 1(b), we can derive $SA_1(5) = \{Diabetes, Pneumonia, Flu\}$; from D'_2 in Figure 1(d), we can derive $SA_2(5) = \{HIV, Pneumonia, Flu\}$. Combining D'_1 and D'_2, we get $MCS(5) = SA_1(5) \cap SA_2(5) = \{Pneumonia, Flu\}$. Therefore, we have the following Zero constraints: $P(HIV \mid \mathcal{I} = 5) = 0$, $P(Diabetes \mid \mathcal{I} = 5) = 0$, and $P(LungCancer \mid \mathcal{I} = 5) = 0$.

To simplify the computation of maximum entropy estimation, we actually remove the variable $P(S \mid \mathcal{I})$ from our variable set if $P(S \mid \mathcal{I}) = 0$. This can reduce the total number of variables in the computation, and can thus improve the computation.

(b) One Constraints. From the properties of conditional probability, we know that the sum of $P(s \mid id)$ should be one for all possible SA values that might be associated to id. Therefore, we call the following equation a *One constraint*:

$$\sum_{s \in MCS(id)} P(s \mid \mathcal{I} = id) = 1. \quad (4)$$

For example, from the previous example, we know that $MCS(5) = \{Pneumonia, Flu\}$; even though we do not know how likely Patient 5 gets $Pneumonia$ or Flu, we do know that the total probability of getting these diseases is 1, i.e., $P(Pneumonia \mid \mathcal{I} = 5) + P(Flu \mid \mathcal{I} = 5) = 1$.

(c) Relation Constraints. Zero constraints and One constraints only depict the relationship of the conditional probabilities of each single id. Relationships among different id's are not captured. However, putting several people's records in the same bucket does somehow make them related. For example, from the first bucket of Figure 1(b), we know two people among $\{1, 2, 3, 4\}$ have Flu, although we do not know which two from this bucket. This information basically makes Patients $1, 2, 3$, and 4 related, and such relationship should be formulated into linear equations in our maximum entropy modeling.

Let $\mathcal{I}(b)$ be the set of id's in bucket b, $P_b(id)$ the probability of id in bucket b, and $P_b(s)$ the probability of SA value s in bucket b. We have following equation:

$$\sum_{id \in \mathcal{I}(b)} P(s \mid \mathcal{I} = id) * P_b(id) = P_b(s), \text{ for } \forall s \text{ in bucket } b. \quad (5)$$

Because each id appears in each published data set only once, we know $P_b(id) = \frac{1}{|b|}$, where $|b|$ is the number of records in bucket b. We also know $P_b(s) = \frac{\#_b(s)}{|b|}$. where, $\#_b(s)$ represents the number of s in bucket b. Therefore, Eq. (5) becomes the following:

$$\sum_{id \in \mathcal{I}(b)} P(s \mid \mathcal{I} = id) = \#_b(s), \text{ for } \forall s \text{ in bucket b.} \quad (6)$$

We call the above equation a *Relation constraint*. For each bucket b in a published data set, we can formulate as many constraints as the number of SA values in the bucket. An example of such a constraint for the first bucket in Figure 1(b) is $P(Flu \mid \mathcal{I} = 1) + P(Flu \mid \mathcal{I} = 2) + P(Flu \mid \mathcal{I} = 3) + P(Flu \mid \mathcal{I} = 4) = 2$.

3.5. Combining constraints

We have discussed how to derive constraints from each individually published data set. We now show how to combine them together to derive $P(S \mid \mathcal{I})$ for a series published data sets $D'_1, \ldots, D'_t$. We let C_i represent the set of constraints derived from D'_i.

If an individual $\mathcal{I}$ appears in a data set D'_i, all his/her conditional probabilities $P(S \mid \mathcal{I})$ must satisfy the constraints in C_i. Intuitively speaking, these constraints rule out many values for $P(S \mid \mathcal{I})$, because they cannot satisfy the constraints. When the same individual appears in another data set, the additional constraints from the new data set might rule out more values for $P(S \mid \mathcal{I})$. Therefore, all the inference channels are actually already embedded in these constraints. Based on this observation, to estimate the value of $P(S \mid \mathcal{I})$ for a series published data sets $D'_1, \ldots, D'_t$, we just need to pool all the constraints together, and create a joint constraint set $C = C_1 \cup \ldots \cup C_t$. Then, we need to find the assignment for $P(S \mid \mathcal{I})$ for each combination of S and $\mathcal{I}$, such that the entropy $H(S \mid \mathcal{I})$ is maximized, while all the constraints in C are satisfied.

4. Evaluation

We use the categorical attributes of Adults data set from the UC Irvine Machine Learning Repository to perform our numerical analysis. All experiments were run on an Intel Pentium-D machine with 3.00 GHz CPU and 4GB physical memory. We use the Knitro software package [4] as nonlinear programming problem solver.

4.1. Inference analysis in various scenarios

In our experiments, we use two data sets D_1 and D_2, each with 7200 records. D_2 is an updated version of D_1. We let $D_1^{'}$ be the bucketized version of D_1, and $D_2^{'}$ be the bucketized version of D_2. D_1 is already published using the conventional bucketization method [15]. We use two different strategies to publish D_2.

(1) Independent re-publishing. When we bucketize D_2 here, we do not take the bucketized result of D_1 into consideration. As results, records in the same bucket in $D_1^{'}$ may be in different buckets in $D_2^{'}$. This "independent" republishing creates the most complicated inference channels. None of the existing work can conduct inference analysis for this scenario. We apply our method on D_1 and D_2 to show the privacy impact of data re-publishing. For each individual data set, we bucketize it to achieve 2-, 3-, and 5-diversity, respectively. For each diversity, we compare the maximum entropy obtained from the single release with that from all the published data. Figure 2 shows the result.

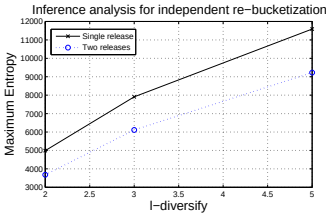

Figure 2. Independent re-bucketization

Compared to the results from single release, the maximum entropy from the combined data sets is 20% to 25% lower, which indicates that the uncertainty of $P(S \mid \mathcal{I})$ is lower. This is the consequence caused by the inference channels between D'_1 and D'_2. The inference channels are introduced by the common records between D'_1 and D'_2.

(2) History-guided re-publishing. To compare, we let the bucketization of D_2 depend on that of D_1. We call it *history-guided data re-publishing*. Most existing republishing schemes fall into this category. Among many ways to conduct history-guided data re-publishing, we try to preserve the similarity of buckets in D'_1 and D'_2.

We use the bucket structure of D'_1 as our basis. If a record is deleted from D'_1, we create an empty slot in its previews position. We then randomly pick a newly added record with the same SA value to fill this slot. However, if we cannot find such a record, we leave the empty slot there. Finally, for the rest of the new records, we use independent bucketization to generate new buckets in $D_2^{'}$.

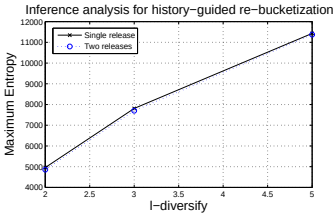

Figure 3. History-guided re-bucketization

Figure 3 depicts the results of history-guided method. From the figure, we can see that the maximum entropy results derived from the combined data are just slightly lower than the ones derived from a single release. This indicates that there are not many inference channels between D'_1 and D'_2. This slight difference is caused by our failing to fill the empty slots when we cannot find a record with the matching SA value. If we happen to fill every empty slot with a new record (or, like m-invariance, with counterfeit records), the maximum entropy from the combined data sets will be exactly the same as that from the single release.

The above experiment shows that the history-guided re-publishing is better than independent re-publishing regarding privacy preserving. How to develop optimal history-guided data re-publishing methods is beyond the scope of this paper.

(3) The impact of overlapping records

The common records shared between two data releases are the causes of inference channels. Here, we study how the number of common records in D_1 and D_2 affect the inference analysis when datasets are independently disguised. We fix the number of records in both D_1 and D_2 to 7200. We set the number of records shared between D_1 and D_2 to 6000, 4800, 3600, 2400, 1200, and 0, respectively. To make the comparison of entropy meaningful, we let D_2 remain the same throughout this experiment and change D_1 according to the number of shared records between D_1 and D_2. Figure 4 plots the entropy of $H(S \mid \mathcal{I})$ for all $\mathcal{I}$'s that appear in D_2. The trend of entropy shows the impact of the number of shared records between D_1 and D_2.

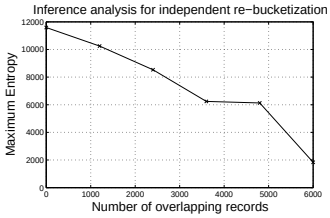

Figure 4. The impact of shared records between D_1 and D_2.

4.2. Performance

To demonstrate the scalability of our method, we record the performance data in Figure 5, including the running time and memory usages. We also show the relationship between the performance and the number of records, variables, and constraints.

For the sake of simplicity, we let the number of records in D_1 and D_2 be the same, i.e., the number of deleted records is the same as the number of new records. We publish the disguised data using independent bucketization. Finally, we conduct the inference analysis for these releases. Our results are shown in Figure 5.

Num of Records	1200	2400	3600	4800	6000	7200
Memory Usage (M)	33	73	116	76	103	134
Running Time (Min)	32	43	49	57	233	221
Num of Variable	11237	30437	49637	23617	29575	34852
Num of Constraints	2921	5841	9036	12101	15601	19121

Figure 5. Performance data

5. Conclusion and Future Work

We describe a generic method to conduct inference analysis across multiple published datasets in PPDR. We formulate the problem as a well-studied maximum entropy estimation problem, and use standard non-linear programming tool to solve it. Experimental results demonstrate the effectiveness of this approach.

The techniques we propose for privacy quantification is general. Following this line, we can develop practical PPDR algorithms to ensure that privacy requirements are satisfied at each releasing point. Also, we can integrate background knowledge (the prior information that adversaries might know about the original data) in privacy analysis of PPDR.

References

[1] D. Agrawal and C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *PODS'01*.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *ACM SIGMOD on Management of Data*, pages 439–450, 2000.

[3] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE'05*.

[4] R. Byrd, J. Nocedal, and R. Waltz. Knitro: An integrated package for nonlinear optimization. In *Large-Scale Nonlinear Optimization*, pages 35–59. Springer-Verlag, 2006.

[5] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li. Secure anonymization for incremental datasets. In *SDM'06*.

[6] C.Yao, X.S.Wang, and S.Jajodia. Checking for k-anonymity violation by views. In *VLDB'05*.

[7] W. Du, Z. Teng, and Z. Zhu. Privacy-MaxEnt: Integrating background knowledge in privacy quantification. In *SIGMOD'08*.

[8] B. C. M. Fung, K. Wang, A. W. C. Fu, and J. Pei. Anonymity for continuous data publishing. In *EDBT'08*.

[9] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *ICDE'05*.

[10] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, May 1957.

[11] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *ICDE'06*.

[12] A. Machanavajjhala, J. E. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. In *ICDE'06*.

[13] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, 1998.

[14] K. Wang and B. C. M. Fung. Anonymizing sequential releases. In *KDD'06*.

[15] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *VLDB'06*, pages 139–150.

[16] X. Xiao and Y. Tao. m-invariance: Towards privacy preserving re-publication of dynamic datasets. In *SIGMOD'07*.