Publishing Skewed Sensitive Microdata

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

Abstract

A *highly skewed* microdata contains some sensitive attribute values that occur far more frequently than others. Such data violates the "eligibility condition" assumed by existing works for limiting the probability of linking an individual to a specific sensitive attribute value. Specifically, if the frequency of some sensitive attribute value is too high, publishing the sensitive attribute alone would lead to linking attacks. In many practical scenarios, however, this eligibility condition is violated.

In this paper, we consider how to publish microdata under this case. A natural solution is "minimally" suppressing "dominating" records to restore the eligibility condition. We show that the minimality of suppression may lead to linking attacks. To limit the inference probability, we propose a randomized suppression solution. We show that this approach has the least expected suppression in a large family of randomized solutions, for a given privacy requirement. Experiments show that this solution approaches the lower bound on the suppression required for this problem.

1 Introduction

A person-specific microdata has the form T(QI, SA). QI is the quasi-identifier consisting of several public attributes (e.g., {birthdate, sex, Zip}) and SA is a sensitive attribute (e.g., Disease). Privacy is violated if it is possible to infer the value on SA of an individual via public knowledge on QI with a high probability [4][23]. To limit this inference, a common defence is imposing diversity on the values of SA, called the *l-diversity principle* [5]. Informally, an anonymized table T* is *l-diverse* if, for any individual I with a record in T, the maximum frequency of SA values in the anonymity group for I is no more than 1/*l*, where the anonymity group is defined as the set of candidate records for I in T*. Generalization [12] and bucketization [6] are two approaches for anonymizing T into *l-diverse* T*. Both approaches are based on partitioning the records into anonymity groups.

1.1 Motivations

However, not every table T has an *l*-diverse T*. For example, if 80% of the records in T have the disease HIV, there is no 2-diverse T*. A necessary and sufficient condition for having an *l*-diverse T* is the *eligibility* condition [6]: no single SA value occurs in more than 1/*l* of the records in T. Let us call this condition *l-eligibility*. All previous works based on *l*-diversity assume that T satisfies *l*-eligibility. On the other hand, there are practical scenarios in which *l*-eligibility is violated. Let us consider several scenarios.

- *Zipf's law distribution*: Many man made and naturally occurring phenomena are distributed according to Zipf's law [18]: the frequency of an event is inversely proportional to its rank in the frequency table, or a small number of events are responsible for a large portion of occurrences. The same principle applies to microdata; therefore, it is not surpresing that certain sensitive attribute values (such as diseeas) are far more common than others.
- *Incremental publishing*: Typically data are published incrementally over a period of time [19][20][22]. According to the law of large numbers, a small size data has a large variance in various statistical properties. Specifically, the small incremental data at each timestamp tends to have more skewed sensitive attribute value distribution than the combined data over a larger time interval. For example, on the real life adverse drug reaction database[1]: if D[i] is published monthly, weekly and daily, we observed no violation of 6-eligibility, 4 % violation of 6-eligibility, and 30% violation of 6-eligibility, respectively.
- *Small data size*: Typically the published microdata is a small subset of records of an underlying database as a result of queries or sampling. For example, in the period of April 16 2003 - May 8 2003, the average number of new Severe Acute Respiratory Syndrome (SARS) cases (including suspected cases) on each day is only 175. If each day's cases must be published separately, such as for temporal pattern analysis, the distribution on the sensitive attribute will be easily imbalanced.

[1]http://www.hc-sc.gc.ca/dhp-mps/medeff/databasdon/

Since previous work assumes *l*-eligibility, they do not provide a clue on how to publish *l*-eligibility violated T. In this paper, we consider the problem of restoring *l*-eligibility; once *l*-eligibility is restored, an existing method can be further applied to generate an *l*-diverse version. There are two key questions: how to restore *l*-eligibility while providing a privacy guarantee, how to restore *l*-eligibility with minimum modification to the data. Before answering these questions, we first consider several possible ways of restoring *l*-eligibility. In the following discussion, the term "non-dominant records" refers to records having infrequent SA values, and the term "dominant records" refers to records having the most frequent SA values.

Solution 1: Add fake records on non-dominant SA values One way is adding fake non-dominant records. Unfortunately, this approach leaves the most frequent sensitive values unchanged in the published data, which allows the attacker to infer the most frequent sensitive value in the original data T. In addition, this approach alters the statistics for non-dominant records. This may have a major impact on utility because non-dominant records often are the research target [16].

Solution 2: Delete non-dominant records An alternative is removing non-dominant records. This approach has a similar limitation to the first approach.

Solution 3: Suppress some dominant records The third way is suppressing some dominant records. In many applications, especially data mining, suppressing some dominant records has very limited impact on the research target, which is usually about non-dominant records. For example, for a data set with 90% records having "Flu" and 10% records having "H1N1", the focus of the research is on classifying H1N1 patients. To obtain good classification results, often dominant records are under-sampled (i.e., suppressed) [15].

In this paper, we consider suppressing dominant records to restore *l*-eligibility. At this point, it seems that we can minimally suppress dominant records until *l*-eligibility is restored. However, the next example shows that such suppression does not protect privacy.

1.2 Eligibility Attacks

Example 1 Figure 1 shows T with five sensitive values $\{S_1{:}10, S_2{:}4, S_3{:}2, S_4{:}1, S_5{:}1\}$, ranked by the frequency. T violates 3-eligibility because $10>|T|/3=18/3$, where $|T|$ denotes the number of records in T. To restore 3-eligibility while minimizing the number of suppressed records, we iteratively suppress the records having the most frequent sensitive value. After suppressing 6 records for S_1, T_P in the center satisfies 3-eligibility *for the first time*. S_i' denotes the i^{th} frequent sensitive value in T_P. With T_P satisfying 3-eligibility, T_P can now be anonymized by any existing method to achieve 3-diversity. The set of suppressed records T_S is withheld from publication. ■

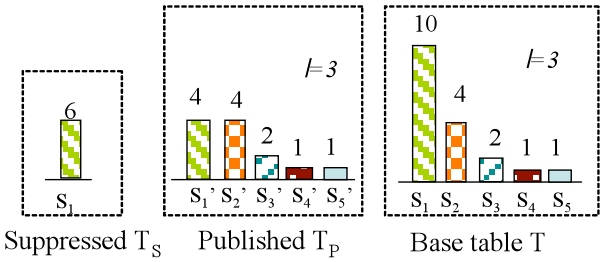

Figure 1 An example of minimum suppression

However, given T_P and the knowledge about the suppression algorithm used, an adversary can infer that only the two most frequent values S_1' and S_2' in T_P can possibly be the most frequent value S_1 in the original T. Suppose that the adversary also knows that T violates 3-eligibility, for example, by being told that some records are suppressed (for the purpose of validating the usefulness of data), which happens only if T violates 3-eligibility. Therefore, the adversary learns that one of S_1' and S_2' must have a frequency higher than 1/3 in T. This imposes a privacy threat for all individuals with a record in T.

The root of the above attack is the *minimality of suppression* for producing T_P, which leaves a small number of candidates for the most frequent sensitive value S_1. The term *eligibility attack* refers to such inference of S_1 from a published subset T_P.

1.3 Contributions

We consider the following *l-dichotomy problem*. Given an *l*-eligibility violated base table T, let S_1 denote the most frequent SA value in T. We want to determine a partition $\{T_P, T_S\}$ of T, where T_P is for publication and T_S is suppressed, such that the following conditions hold.

- *Privacy*: T_P is *l*-eligible and, for every sensitive value S_i' in T_P, the posterior probability of $S_i'=S_1$, given T_P and K, is bounded by $1/l$. K denotes the knowledge about the algorithm for producing $\{T_P, T_S\}$.
- *Utility*: The number of records in T_P is maximized while satisfying the above privacy requirement. Since our approach is suppressing dominant records, this optimality minimizes the suppression of dominant records.

The above problem has several novelties. First, it considers an *l*-eligibility violated base table T. To our knowledge, anonymzing such microdata has not been considered previously. Second, it considers the knowledge about the algorithm that produces T_P. With such knowledge, *l*-eligibility of T_P alone is not sufficient, as illustrated by Example 1. Third, once *l*-eligibility is enforced on T_P, existing methods such as [5][6] can be applied to render T_P *l*-

diverse. In this sense, our work makes existing works applicable to skewed data.

The contributions of this work are as follows. We formulate the *l*-dichotomy problem (Section 2). We introduce two deterministic solutions (Section 3): one provides a lower bound on suppression but no privacy guarantee, and one guarantees privacy but suppresses too much data. Then, we propose a randomized solution (Section 4). We show that this solution has the least expected suppression in a large family of randomized algorithms (Section 5), and has a suppression approaching the lower bound (Section 6).

1.4 Related Work

We adopt the *l*-diversity [5] as the privacy notion. Previous works assume that the global distribution satisfies the special *l*-eligibility condition [6], and the focus is on limiting sensitive inference arising from the local distribution of anonymity groups. In the case that *l*-eligibility condition is not satisfied, no *l*-diverse table is produced. The work in [3] measures the privacy by the difference between local distribution and the global distribution. Thus a skewed global distribution is not considered sensitive. We consider a skewed global distribution on SA a privacy threat and have to deal with how to publish data with such distribution.

Our work is related to privacy threats due to background knowledge such as [5][9][10]. The background knowledge previously considered is primarily additional information about QI and SA. We consider background knowledge about the algorithm used for producing T_P and the knowledge about the violation of *l*-eligibility of the base table T. Such knowledge cannot be easily handled by previous works.

The recent work in [7] dealt with attacks arising from minimality principles of anonymization algorithms. The motivations, assumptions, attacks, and problems in their work are different from ours. For example, [7] assumes that the global distribution satisfies the *l*-eligibility condition and deals with minimality attacks that arise from the anonymization for achieving *l*-diversity. We assume that the *l*-eligibility is violated and deal with eligibility attacks that arise from the skewness of global distribution on SA.

Record suppression has been used to protect privacy previously [11][12][13][14]. In those works, records are suppressed because they are "outliers" for *k*-anonymity. In our works, records are suppressed because their sensitive values are too frequent.

2 Problem Statement

The base table T has several quasi-identifying attributes (QI) and one sensitive attribute SA. SA has *m* distinct sensitive values $S_1,\ldots,S_m$ with the frequency order $F_1\geq\ldots\geq F_m>0$ (in the number of records). S_i is the i^{th} frequent SA value in T and has the i^{th} rank. Table 1 summarizes the notations used in this paper.

T is *l-eligible* if no more than $|T|/l$ of the records shares a common SA value, i.e., $F_1 \leq |T|/l$. A *violating value* refers to S_i with $F_i>|T|/l$. We assume $l\leq m$. Anonymized table T* consists of several anonymity groups, obtained by generalization [12] or bucketization [6]. T* satisfies *l-diversity*, or is *l-diverse* [5], if for every anonymity group g in T*, the maximum frequency of any SA value in g is no more than $1/l$. Note that *l*-eligibility of T coincides with *l*-diversity of T* if T* has a single anonymity group.

l-eligibility of T is both necessary and sufficient for T to have an *l*-eligible T* [5][6]. If T violates *l*-eligibility, existing works cannot be applied to produce *l*-eligible T*. For this reason, existing works assume that T satisfies *l*-eligibility. In this paper, we consider T that violates *l*-eligibility.

Table 1. Notations

m	The number of distinct sensitive values on SA.
l	The parameter for *l*-eligibility, $l\leq m$.
T	The base table. $\|T\|$ is the number of records in T.
T_P (T_S)	Published (Suppressed) subsets of T.
S_i, F_i	S_i - the i^{th} frequent sensitive value in T; F_i - the frequency of S_i in T.
S_i', F_i'	S_i' - the i^{th} frequent sensitive value in T_P; F_i' – the frequency of S_i' in T_P.

Suppose that T violates *l*-eligibility, i.e., $F_1>|T|/l$. We want to determine a partition $\{T_P,T_S\}$ of T (i.e., $T= T_P\cup T_S$ and $T_P\cap T_S=\varnothing$) such that T_P is *l*-eligible and is for publication and T_S is withheld from publication. Let $S_1',\ldots,S_m'$ denote the SA values in T_P with their frequency order $F_1'\geq\ldots\geq F_m'$. Note that $F_i \geq F_i'$, $i=1,\ldots,m$. We represent the frequency of SA values in T by $\{S_1:F_1,\ldots,S_m:F_m\}$ and represent the frequency of SA values in T_P by $\{S_1':F_1',\ldots,S_m':F_m'\}$.

Adversary Knowledge The attacker has access to the published T_P, therefore, knows the rank $S_1',\ldots,S_m'$ in T_P with $F_1'\geq\ldots\geq F_m'$. But the attacker has no access to T or T_S. The attacker has the following knowledge, denoted by K.

- **K1**: The algorithm used to produce T_P.
- **K2**: $|T|$. In practice, the researcher may be told the number of records suppressed, i.e., $|T_S|$, for validating the usefulness of data. Therefore, the original size $|T|$ is likely public. From $|T|>|T_P|$, the adversary also knows that the original data T is not *l*-eligible (i.e., $F_1>|T|/l$), otherwise, $|T|=|T_P|$.

As for the knowledge on the most frequent sensitive value S_1 in T, there are two cases. In the first case, the adversary knows S_1 in T *before* any data is published. In this case, even when no data is published, the adversary can infer S_1 with a probability higher than $1/l$; thus, no method can provide

protection. In the second case, the attacker does not know the most frequent sensitive value S_1 in T before seeing T_P. The attack occurs when the attacker can infer $S_i'=S_1$ (i.e., some value S_i' in T_P is S_1) *after* seeing the published T_P. This is the case we consider in this paper. We denote this posterior probability by $Pr(S_i'=S_1|T_P, K)$. Our goal is to limit $Pr(S_i'=S_1|T_P, K)$ while publishing as many non-dominant records as possible. We formalize this problem below.

Definition 1 (*l*-Dichotomy) Consider an *l*-eligibility violated T and a partition $\{T_P, T_S\}$ of T. Let S_1 denote the most frequent SA value in T. We say that (T_P, K) satisfies *l-dichotomy* if both of the following conditions hold:

- *P-Eligibility*: $F_1' \le |T_P|/l$, that is, T_P is *l*-eligible. The term "P-Eligibility" comes from the fact that this condition is on the published table T_P.
- *S-Ambiguity*: $Pr(S_i'=S_1|T_P,K) \le 1/l$, $i=1,\ldots,m$. The term "S-Ambiguity" comes from the fact that this condition is on the suppression algorithm.

The *l-dichotomy problem* is to find a partition $\{T_P,T_S\}$ such that (T_P,K) satisfies *l*-dichotomy, and as many non-dominant records as possible are published in T_P.■

The definition of S-Ambiguity does not suggest an efficient test of the condition. To address this problem, we consider an implied condition that can be tested efficiently. Consider an *l*-eligibility violated T, i.e., $F_1>|T|/l$, and a partition $\{T_P, T_S\}$ of T. If S_i' is S_1, $F_i'+|T_S|\ge F_1$, because the frequency of S_1 in T_S is no more than $|T_S|$ and the frequency of S_1 in T_P is equal to F_i'. Therefore, to provide S-Ambiguity, there must exist at least *l* candidates S_i' such that $F_i'+|T_S|>|T|/l$; otherwise fewer than *l* values S_i' can possibly be S_1, which implies $Pr(S_i'=S_1| T_P, K)>1/l$. Since $F_1' \ge \ldots \ge F_l' \ge \ldots \ge F_m'$, this condition is stated as the *l*-candidacy condition below.

Definition 2 (*l*-Candidacy) We say that (T_P, K) satisfies *l-candidacy* if $F_l'+|T_S|>|T|/l$. ■

From the above discussion, S-Ambiguity implies *l*-candidacy. The converse is not true. (T_P,K) in Example 1 with $l=3$ provides a counter-example. (T_P,K) satisfies *l*-candidacy because $F_l'+|T_S|=2+6>|T|/l=18/3$. However, as in Example 1, (T_P,K) violates S-Ambiguity.

Corollary 1 S-Ambiguity implies *l*-candidacy. The converse is not true. ■

P-Eligibility states that the most frequent sensitive value S_1' in T_P is not too frequent, whereas *l*-candidacy says that the next *l*-1 frequent values in T_P, i.e., $S_2',\ldots,S_l'$, are not too infrequent (otherwise they will be disqualified for being S_1). We will show that, for the randomized algorithm in Section 4, *l*-candidacy and P-Eligibility together will guarantee S-Ambiguity.

3 Deterministic Suppression

The *deterministic* suppression in Example 1 *always* suppresses a record for the most frequent value S_1' in T_P, which minimizes the number of suppressed records for satisfying P-Eligibility. On the other hand, it is this minimality that compromises S-Ambiguity, as illustrated in that example. Nevertheless, this deterministic suppression serves an important building block in our randomized algorithm and provides a lower bound on suppression for achieving S-Ambiguity. In this section, we present a formal analysis of this deterministic suppression.

Consider a record *r* in T_P having the value S_i'. The *level* of *r* refers to the frequency F_i' of S_i' in T_P (in number of records). The *level* of T_P refers to the highest level of the records in T_P, i.e., F_1'. $top(T_P)$ denotes the set of distinct sensitive values in T_P that have the highest level in T_P.

3.1 D-suppression

Figure 2 presents the general form of the deterministic suppression, called *D-suppression*. Starting with T_P=T, D-suppression iteratively suppresses a record for a sensitive value in $top(T_P)$, the lowest rank first if there are more than one value. In other words, it always suppresses a record having the highest frequency in T_P. The suppressed records are contained in T_S. We will consider several stop conditions shortly.

D-suppression:
Input: *l*-eligibility violated T.
Initialize T_P to T and initialize T_S to ∅.
While Stopping condition not true **do**
  Choose S_j from $top(T_P)$
  Move one record for S_j from T_p to T_S.

Figure 2 D-suppression

Example 2 Consider T in Figure 1. Initialize T_P to T and $top(T_P)=\{S_1\}$. After suppressing 6 records for S_1, T_P is at level 4 and $top(T_P)=\{S_1,S_2\}$. Then D-suppression suppresses one record for S_2 because S_2 has a lower rank than S_1 in T, and then suppresses one record for S_1 because $top(T_P)=\{S_1\}$. ■

We say that an algorithm *stops at a level* F if T_P produced is at the level F, i.e., $F_1'=F$; an algorithm *stops at the start of a level* F if it stops at the level F without suppressing any record at the level F.

Property 1 Consider T_P during the course of D-suppression,

1. (*Order Preservation*) D-suppression preserves the "≥" order of frequency in T. As a result, S_1 is *always* contained in $top(T_P)$. Therefore, for every $S_i' \in top(T_P)$, $Pr(S_i'=S_1|T_P,K)=1/|top(T_P)|$, and for every $S_i' \notin top(T_P)$, $Pr(S_i'=S_1|T_P,K)=0$.
2. (*Least Suppression*) D-suppression stopping at the start of the level F has the *least* suppression among *all* algorithms stopping at the level F.

From Property 1(1), S-Ambiguity is compromised if D-suppression stops with $|top(T_P)|<l$. From Property 1

Property 1(2), D-suppression has the least suppression among all algorithms stopping at the same level. This property allows us to obtain a lower bound on the number of suppressed records for the deterministic suppression. Below, we identify two interesting stopping conditions of D-suppression.

3.2 Safe D-Suppression

Safe D-suppression refers to D-suppression that stops when the combination of P-Eligibility and S-Ambiguity is achieved *for the first time*. Let F_{Safe} and X_{Safe} denote the level of T_P and the suppression when Safe D-suppression stops. The next theorem tells that Safe D-suppression always stops at the start of the level F_l.

Theorem 1 (Safe D-suppression) Safe D-suppression stops at the start of the level F_l, i.e., $F_{Safe}= F_l$.■

Proof: We show that the start of the level F_l is the first time where the combination of P-Eligibility and S-Ambiguity is achieved. From

Property **1**(1), $Pr(S_i'=S_1|T_P,K) =1/|top(T_P)|$. Consider the T_P produced when D-suppression stops at the start of the level F_l. At this point, all of $S_1,\ldots,S_l$ are suppressed to the level F_l, thus $top(T_P)$ contains $S_1,\ldots, S_l$ and $Pr(S_i'=S_1|T_P,K)\leq 1/l$, so S-Ambiguity is satisfied. Also, $|T_P|= l\times F_l+\sum_{i>l} F_i$, thus $F_l \leq |T_P|/l$, so P-Eligibility is satisfied. At any level *above* F_l, $top(T_P)$ contains at most $S_1,\ldots, S_{l-1}$, so $|top(T_P)|<l$, from

Property **1**(1), S-Ambiguity is not satisfied. □

Safe D-suppression achieves S-Ambiguity by suppressing *all* of the l most frequent values $S_1,\ldots,S_l$ to the frequency F_l. For a small F_l and a large l, this will suppress too many records. In Example 1, with l=3, this means suppressing 8 records for S_1 and 2 records for S_2, which is more than half of the data. Although Safe D-suppression provides a solution, it loses too much information.

3.3 Unsafe D-suppression

Unsafe D-suppression refers to D-suppression that stops when the combination of P-Eligibility and l-candidacy is achieved *for the first time*. Let F_{Unsafe} denote the level when Unsafe D-suppression stops and X_{Unsafe} denote the suppression by Unsafe D-suppression.

Example 3 Consider D-suppression on T in Figure 1 with l =3. After suppressing 5 records for S_1, l-candidacy is achieved for the first time: $F_3'+|T_S|=2+5>|T|/l=18/3$. After suppressing one more record for S_1, P-Eligibility is achieved for the first time at the start of level 4. Unsafe D-suppression stops right here with F_{Unsafe} = 4 and X_{Unsafe}=6. S-Ambiguity is not satisfied until D-suppression continues to the start of the level F_3 =2. This is where Safe D-suppression stops, with F_{Safe}=2 and X_{Safe}=10. ■

Theorem 2 (Unsafe D-suppression) The suppression by *any* algorithm that achieves both P-Eligibility and l-candidacy is at least X_{Unsafe}.■

Proof: D-suppression always suppresses the record at the highest level in T_P in each iteration. Such suppression most effectively reduces the level F_1' of T_P for achieving P-Eligibility $F_1'\leq|T_P|/l$. Also, such suppression leaves F_l' at the highest possible level for a given suppression size $|T_S|$, thus requires the least suppression $|T_S|$ to achieve l-candidacy $F_l'+|T_S|>|T|/l$.□

Recall that S-Ambiguity implies l-candidacy (Corollary 1). Therefore, from Theorem 2, Unsafe D-suppression provides a *lower bound* on suppression for achieving l-dichotomy, that is, no algorithm for achieving l-dichotomy can have less suppression than Unsafe D-suppression. A solution is considered interesting if its suppression approaches this lower bound and is much less than the suppression of Safe D-suppression. Note that Unsafe D-suppression does not provide a solution because it does not guarantee S-Ambiguity (Example 3). In Section 4, we shall show that Unsafe D-suppression provide S-Ambiguity when it is used by a randomized algorithm.

4 Randomized Suppression

Safe D-suppression guarantees l-dichotomy at the cost of suppressing all of $S_1,\ldots,S_l$ to the level F_l. Unsafe D-suppression has a small suppression but fails to achieve S-Ambiguity. The problem with both is their deterministic nature of suppression, which leads to either excessive suppression or a clue left on T_P to infer S_1. We now address this problem addressed by introducing randomness into the suppression of S_1. In this section, we present such a randomized algorithm, R-suppression. In Section 5, we show that R-suppression has the least suppression in a large family of randomized algorithms.

4.1 R-suppression

R-suppression has two steps. In the first step, it suppresses S_1 to a random level. In the second step, it suppresses records deterministically to achieve l-dichotomy like Unsafe D-suppression. This algorithm is given in Figure 3.

STEP-1 (Randomization Step) suppresses S_1 randomly so that, after the suppression, any of the most frequent values $S_1',\ldots,S_l'$ on T_P can possibly be S_1 with a bounded probability. Specifically, it picks a random number h from $1..l$ and then suppresses S_1 to a random level F chosen uniformly from the interval $[F_{h+1}, F_h]$. After the suppression, S_1 becomes S_h' on T_P. The probability for picking h is equal to $1/l$. The uniform distribution of F in $[F_{h+1}, F_h]$ ensures that the attacker gains no new information from where S_1 is likely to be suppressed to in the interval $[F_{h+1}, F_h]$.

STEP-2 (Deterministic Step) calls for D-suppression to suppress more records from T_P to achieve P-Eligibility and *l*-candidacy.

R-suppression:
Input: *l*-eligibility violated T, and *l*.
Output: T_P.
STEP-1: Let T_P be T. (1) Pick an integer *h* from 1..*l* at random with equal probability 1/*l*. (2) Pick a level F at random uniformly from $[F_{h+1}, F_h]$. (3) Suppress S_1 to the level F on T_P.
STEP-2: Apply D-suppression to T_P and stop when (T_P,K) satisfies P-Eligibility and *l*-candidacy for the first time. Return T_P.

Figure 3 R-suppression

Example 4 Consider the example in Figure 1. Recall $T=\{S_1:10, S_2:4, S_3:2, S_4:1, S_5:1\}$ and $l=3$. Suppose that the random choice in STEP-1 of R-suppression is $h=2$ and F=3, that is, it picks the interval $[F_3,F_2]=[2,4]$ and the level F=3 from $[F_3,F_2]$. S_1 is suppressed to F=3 in STEP-1. Now $T_P=\{S_1':4, S_2':3, S_3':2, S_4':1, S_5':1\}$. STEP-2 calls for D-suppression to further suppress some records to satisfy P-Eligibility and *l*-candidacy. STEP-2 stops with $T_P=\{S_1':3, S_2':3, S_3':2, S_4':1, S_5':1\}$ where P-Eligibility and *l*-candidacy are satisfied for the first time. ■

Two questions about R-suppression must be answered. First, are P-Eligibility and *l*-candidacy in STEP-2 sufficient for achieving *l*-dichotomy? We will show in Section 4.2 that, with the randomization in STEP-1, the answer is yes. Second, does R-suppression provide the least suppression among randomized solutions? We will present a proof of this optimality in Section 5.

4.2 *l*-Dichotomy of R-suppression

We show that R-suppression achieves P-Eligibility and S-Ambiguity (therefore, *l*-dichotomy). P-Eligibility is enforced in STEP-2. We focus on S-Ambiguity. Prior to the proof, we first illustrate how $Pr(S_i'=S_1|T_P,K)$ is determined from the *adversary's point of view*, T_P and K.

Example 5 Continue with $T=\{S_1:10, S_2:4, S_3:2, S_4:1, S_5:1\}$ and $T_P=\{S_1':3, S_2':3, S_3':2, S_4':1, S_5':1\}$ in Example 4. Recall that the adversary has access to T_P, not T. With the knowledge K, the adversary knows $|T|-|T_P|=8$ records are suppressed. Any base table B that is consistent with the knowledge K and the observed T_P would be "plausible". From the knowledge on R-suppression, S_1 is among the three most frequent values in T_P. This means that the 8 suppressed records can have any distribution of S_1', S_2' and S_3'.

For example, one plausible base table is $B=\{S_1':11, S_2':3, S_3':2, S_4':1, S_5':1\}$, assuming that all 8 suppressed records have S_1'. For this B, $S_1'=S_1$. Given this B, R-suppression can produce the observed T_P by choosing $h=1$ and any $F\in[3,11]$. Similarly, by assuming that all 8 suppressed records have S_2', there is another plausible base table B in which $S_2'=S_1$. The third plausible base table B is $\{S_3':10, S_1':3, S_2':3, S_4':1, S_5':1\}$, assuming that all 8 suppressed records have S_3'. However, S_1 cannot be S_4' because S_1 must be among the three most frequent values in T_P.

Since all these reconstructions of the base table are plausible, the adversary cannot exclude any of S_1', S_2' and S_3' as the candidate of S_1. Without further information, the adversary cannot tell which reconstruction is more likely. Therefore, $Pr(S_i'=S_1|Tp, K) = 1/3$, where i=1,2,3. ■

Theorem 3 (*l*-Dichotomy of R-suppression) For T_P produced by R-suppression, (T_P,K) satisfies P-Eligibility and S-Ambiguity (thus, *l*-dichotomy).■

The proof is given in Appendix.

5 Optimality Of R-Suppression

We show that R-suppression has the least expected suppression in a large family of randomized algorithms, called *R*-suppression* family. This family contains all randomized algorithms that randomize the suppression of S_1 in the first step and deterministically suppress more records to satisfy *l*-dichotomy in the second step. The members vary in the specific procedures used in each step. R*-suppression is obtained from R-suppression in Figure 3 with the following modifications:

STEP-1: pick an integer *h* from 1..*m* (instead of 1..*l*) at random with probability $\leq 1/l$ (which is possible because $m\geq l$). This step bounds the probability of $S_1=S_h'$ on T_P by 1/*l*, for $h=1..m$. Any special case under this constraint gives rise to an instance of this step. For example, STEP-1 of R-suppression is the special case where *h* is picked from 1..*l* with the equal probability 1/*l*.

STEP-2: replace "D-suppression" with "any deterministic suppression algorithm" and replace "P-Eligibility and *l*-candidacy" with "*l*-dichotomy". Any choice of a deterministic algorithm gives rise to an instance. For example, R-suppression chooses D-suppression.

Each combination of the choices in STEP-1 and STEP-2 gives rise to a member in the R*-suppression family. Obviously, R-suppression is a member in this family.

Since R is a randomized algorithm, its execution is on an random instance. Let R_i denote the random instance of R-suppression with *i* being the initial suppression of S_1 at STEP-1, i.e. $i= F_1-F$ where F is the level for S_1 chosen in STEP-1. Let X_i be the total suppression by R_i in the two steps. Let X be the random variables for X_i, and E(X) be the expected value of X, respectively. Similarly, R_i*, X_i*, X and E(X*) denote the counterparts for R*-suppression.

Theorem 4 (Optimality of R-suppression) $E(X)\leq E(X^*)$**.** That is, R-suppression has the least expected suppression in the R*-Suppression family. ■

Proof: The intuition is that R-suppression has a higher probability for a smaller initial suppression of S_1 in STEP-1

and maximally relies on the most effective D-suppression in STEP-2 to achieve P-Eligibility and l-candidacy. More formally, for any initial suppression i of S_1 in STEP-1, if $i \le F_1 - F_{l+1}$, $Pr(R_i) \ge Pr(R_i^*)$, and if $i > F_1 - F_{l+1}$, $Pr(R_i)=0$. From Theorem 2, for every initial suppression i in STEP 1, we have $X_i \le X_i^*$. Note $\Sigma_i Pr(R_i) = \Sigma_i Pr(R_i^*)=1$. Together, these imply $\Sigma_i Pr(R_i) \times X_i \le \Sigma_i Pr(R_i^*) \times X_i^*$, thus $E(X) \le E(X^*)$. □

6 Empirical Study

In this section, we evaluated the suppression of R-suppression empirically on both real life data and typical data distributions. For a given solution $\{T_P, T_S\}$, we measure the distortion by *Suppression Rate* defined as $|T_S|/|T|$, i.e., the percentage of records suppressed, where $|T|=|T_P|+|T_S|$. Since our approach prefers suppressing records for dominant sensitive values (except for the random suppression of S_1 in STEP-1) and we assume that records for non-dominant sensitive values have more utility, the suppression rate reflects the amount of dominant records suppressed. The smaller the suppression rate, the more records for non-dominant records can be published. We compare the following suppression methods.

- **Safe D** – Safe D-suppression, which provides a valid solution to the l-dichotomy problem.
- **Unsafe D** – Unsafe D-suppression, which provides a lower bound on suppression for the l-dichotomy problem (Theorem 2).
- **R-sup** – R-suppression. This is the proposed solution to the l-dichotomy problem. We report the average of 100 random instances for R-suppression. We do not evaluate other members in the R*-suppression family as R-sup has the least expected suppression in this family (Theorem 4).

We use Safe D and Unsafe D as references. R-sup is effective if its suppression rate approaches the lower bound of Unsafe D and is much lower than the suppression rate of Safe D. We conducted two experiments, one on real life data (Section 6.1) and one on two representative data distributions (Section 6.2). All experiments were implemented in C++ and run on a PC with 2.4GHz CPU, 512M memory and Windows XP.

6.1 Sample based Publishing

The first experiment simulates incremental data publishing by a series of samples drawn from the base population adapted from the real life data set Adults from UC Irvine Machine Learning Repository [1]. This data contains person-specific records from the US Census, collected from real life US demographics, and has been used as a de facto benchmark [14]. After removing records with missing values, the resulting data contains 30,162 records. On the conservative side, we selected the "Occupation" as SA because it is the least skewed attribute (see Figure 4). SA has 14 distinct values with the most frequent value having the relative frequency of 13.4%. Thus, there is no violation of l-eligibility in the base population for l=6.

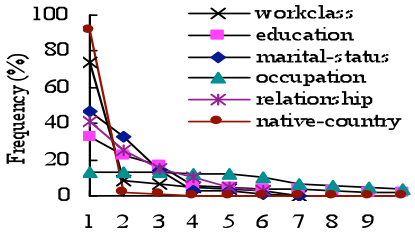

Figure 4 Attribute value distribution (value vs frequency)

Samples violating l-eligibility First, we examine how often l-eligibility is violated by a sample drawn from the Adult population. For a given *sample size* p (in fraction), we take 100 samples from the Adult data set. Each sample T has the size $|T| = p \times |Adult|$ and is taken by simple random sampling without replacement. T is a *violating sample* if T does not satisfy l-eligibility. Figure 5 plots the number of violating samples among 100 samples, for each pair of p and l values.

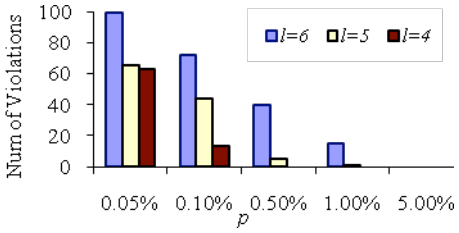

Figure 5 Eligibility violations vs *p* and *l*

For a fixed l (i.e., the same color), with a small sample size p, the distribution of values on SA is skewed and there are more violating samples. For a fixed sample size p, a larger l sets a harder eligibility condition to satisfy, thus leads to more violating samples. For example, at p=0.5% or 150 records, the percentage of violating samples is as high as 40% for l=6. Such small sample size is likely for the reasons discussed in Section 1.1. Interestingly, these violations happen on samples though the underlying Adult data set as a whole does not violate l-eligibility for l=6. This result verifies our claim in Section 1.1 that l-eligibility is violated as the sample size decreases.

We should mention that the Adult data set is only moderately skewed with the highest frequency being 13.4%. A typical distribution like those following Zipf's law can be much more skewed (see Section 6.2). In those cases, we expect a more frequent violation of l-eligibility.

Suppression Rate For each violating sample, we apply the above-mentioned methods to study suppression rate. For the purpose of reference, we also include the distortion of suppressing all records in a sample if the sample violates l-

eligibility, denoted by "Suppress-All". In fact, this is the "state-of-the-art" solution given that existing works consider only the case that *l*-eligibility is satisfied.

For each setting of *p* and *l*, we compute the averaged suppression rate, $(\Sigma_i SR_i)/n$, where SR_i represents the suppression rate of a violating sample i, and n is the total number of samples, which is 100 in our experiment. Intuitively, the averaged suppression rate is the percentage of data suppressed over all samples. Figure 6 and Figure 7 show the averaged suppression rate vs *p* and *l*, respectively.

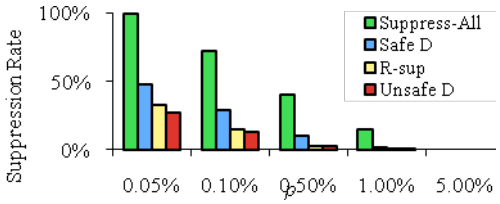

Figure 6 Suppression Rate vs *p* (*l*=6)

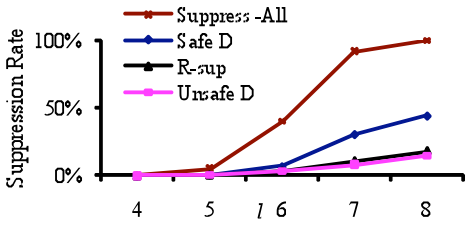

Figure 7 Suppression Rate vs *l* (*p*=0.5%)

In both Figure 6 and Figure 7, an outstanding trend is that R-sup suppresses only slightly more than the lower bound of Unsafe D for all *p* and *l* tested. Take the setting of *p*=0.5% and *l*=6 as an example. R-sup suppresses less than 3% records on each sample (on average), which is nearly identical to the suppression of Unsafe D. This result is encouraging in that Unsafe D serves a lower bound for achieving P-Eligibility and *l*-candidacy, which are necessary (but not sufficient) for achieving *l*-dichotomy. R-sup approaches this lower bound while guaranteeing the stronger *l*-dichotomy.

On the other hand, R-sup suppresses much less than Safe D. For example, for the same setting of *p*=0.5% and *l*=6, Safe D suppresses more than 10% records of each sample (on average). And suppression goes up quickly for a smaller *p* and a larger *l*. This result confirms our expectation that Safe D's naïve suppression to the level F_l loses too much data. Suppress-All is far worse than all other approaches.

6.2 Zipf Distributions

Besides real life data, we also consider some well-known data distributions. Many types of data studied in physical and social science follow Zipf's law. In this experiment, we assume that the frequency F_i of sensitive values on SA follows this law: $F_i=\beta/i^{\alpha}$, where $\beta>0$ and $\alpha>0$, i=1,…,*m*, where *m* is the number of distinct sensitive values. β, which is equal to F_1, determines the scale of frequency but has little impact on the (relative) suppression rate. α determines the decreasing rate of F_i and is a major factor for eligibility violation. We use the default setting β=1000, α=1.0 and *l*=5. *m*=20.

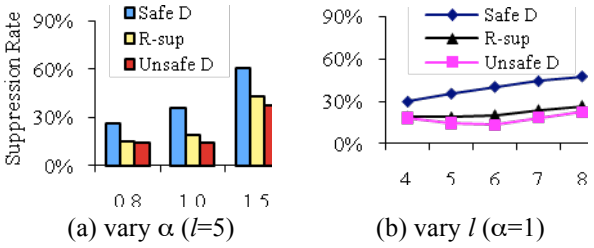

Figure 8 Suppression Rate under Zipf distribution

Figure 8(a) shows suppression rate vs α. For α below 0.8, there is no eligibility violation. For a larger α, F_i decreases faster, and thus, causes more eligibility violation and requires more suppression. This can be seen from the high lower bound of Unsafe D. Even in this case, R-sup has about 20% less suppression than Safe D and stays close to the lower bound of Unsafe D.

Figure 8(b) shows suppression rate vs *l*. Again, R-sup stays close to the lower bound of Unsafe D-sup and has much less suppression than Safe D. Interestingly, when *l* is increased from 4 to 6, suppression rate of Unsafe D decreases. This is because *l*-candidacy $|T_S|+F_l'>|T|/l$ requires less suppression when *l* increases, for a small *l*. However, when *l* is increased from 6 to 10, suppression rate increases because P-Eligibility $F_1' \le |T_P|/l$ dominates in this case.

We also conducted experiments on linear distributions, but due to the space, the results are omitted. In general, we observed a similar trend as that on Zipf distribution, except that there are less eligibility violations and consequently require less suppression than the case of Zipf distribution.

7. Conclusions

To provide the protection of *l*-diversity, existing works assumes that the global distribution of sensitive values is not skewed. In several real life scenarios, this assumption is violated. In this work, we show that publishing a maximum *l*-eligible subset may lead to the inference of the most frequent sensitive value in the original data, called eligibility attacks. We formulated the problem of preventing eligibility attacks while allowing more data for publication. We proposed a simple but effective randomized solution to this problem, i.e., R-suppression. R-suppression has the least expected suppression among a large family of randomized algorithms.

On both real life data and common data distributions, our experiments showed that R-suppression approaches the lower bound on suppression required for this problem.

Acknowledgements: The research of Raymond Chi-Wing Wong is supported by HKRGC GRF 621309 and Direct Allocation Grant DAG08/09.EG01.

References

[1] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.

[2] J. Milton and J. Arnold. Introduction to probability and statistics: Principles and applications for engineering and the computing sciences, McGraw Hill, 1995.

[3] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: Privacy beyond *k*-anonymity and *l*-diversity. ICDE 2007.

[4] L. Sweeney. Achieving *k*-anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002.

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

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

[7] R. Wong, A. Fu, K. Wang, and J. Pei. Minimality attack in privacy preserving data publishing. VLDB 2007.

[8] R. Wong, J. Li, A. Fu, and K. Wang. (α, *k*)-Anonymity: An enhanced *k*-anonymity model for privacy-preserving data publishing. SIGKDD 2006.

[9] D.J Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, J. Y. Halpern. Worst-case background knowledge for privacy-preserving data publishing. ICDE 2007.

[10] Wenliang Du, Zhouxuan Teng, Zutao Zhu. Privacy-MaxEnt: Integrating background knowledge in privacy quantification. SIGMOD 2008.

[11] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. ICDE 2005.

[12] P. Samarati. Protecting respondents' identities in microdata release. TKDE 13, 6, 1010-1027.

[13] F. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: efficient full-domain k-anonymity. SIGMOD 2005.

[14] V.S. Iyengar. Transforming data to satisfy privacy constraints. SIGKDD 2002.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 2002.

[16] P Domingos. MetaCost: A general method for making classifiers cost-sensitive. KDD 1999.

[17] N.V. Chawla, N. Japkowicz, A. Kotcz, SIGKDD Explorations, Special Issue on Class Imbalances, vol. 6(1), New York, June 2004.

[18] L. A. Adamic. Zipf, power-laws, and pareto – a ranking tutorial. http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html, 2002.

[19] R. T. Greenlee. Measuring Disease Frequency in the Marshfield Epidemiologic Study Area (MESA). Clinical Medicine & Research. Volume 1, Number 4: 273-280, 2003

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

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

[22] B. Fung, K. Wang, A. Fu, J. Pei. Anonymity for Continuous Data Publishing. EDBT 2008.

[23] P. Samarati. Protecting respondents' identities in micro data release. *TKDE*, vol. 13, no. 6, pp. 1010–1027, 2001.

Appendix

Theorem 5 (*l*-Dichotomy of R-suppression) For T_P produced by R-suppression, (T_P,K) satisfies P-Eligibility and S-Ambiguity (thus, *l*-dichotomy).■

To prove Theorem 5, we need to show $Pr(S_i'=S_1|\ T_P,K) \le 1/l$, for $i=1,\ldots,m$, where T_P is produced by R-suppression. The intuition is as follows. STEP-1 first establishes the prior belief that any of S_i' with $i=1,\ldots,l$ has equal probability being S_1, i.e. $Pr(S_i'=S_1|K) = 1/l$. This is why *h* and F are picked in the specific ways described in STEP-1, Figure 3. In STEP-2, this belief remains unchanged because D-suppression preserves the frequency order (

Property **1**(1)) and *l*-candidacy ensures that the attacker cannot exclude any of S_i' with $i=1,\ldots,l$ as the candidates for S_1 even after observing T_P. As a result, $Pr(S_i'=S_1|\ T_P, K) = 1/l$ for $i=1,\ldots,l$. We now give a formal proof.

Recall that the attacker has no access to T and T_S, and infer $S_i'=S_1$ from the "view" (T_P, K), where K is the knowledge defined in Section 2. Borrowed from Bayesian Theorem [2], we have the rewriting

$$\Pr(S_i' = S_1 \mid T_P, K) = \frac{\Pr(T_P \mid S_i' = S_1, K) \times \Pr(S_i' = S_1 \mid K)}{\Pr(T_P \mid K)} \quad (1)$$

Here, $S_i'=S_1$ is the "hypothesis" the attacker tries to establish and T_P is the "evidence". $Pr(S_i'=S_1|K)$ is the attacker's prior belief in the hypothesis before observing the evidence T_P. $Pr(S_i'=S_1|T_P,\ K)$ is the attacker's posterior belief after observing the evidence T_P.

From STEP-1, the target interval $[F_{h+1},\ F_h]$ is randomly selected from the first *l* intervals with equal probability. So

the prior belief $Pr(S_i'=S_1|K)$ is equal to $1/l$ for $i \le l$, and 0 for $i>l$. Together with the uniform distribution of the level F in $[F_{h+1}, F_h]$, STEP-1 injects sufficient randomness to bound the attacker's prior belief on $S_i'=S_1$ by $1/l$ for $i \le m$.

We show $Pr(S_i'=S_1|T_P,K) = Pr(S_i'=S_1|K)$ for $i \le m$, i.e., attackers' belief is not changed after observing T_P produced by STEP-2. From Equation (1), it suffices to prove $Pr(T_P|K) = Pr(T_P|K, S_i'=S_1)$ for $i \le l$. That is, knowing that S_i' is most frequent in the base table does not alter the chance of T_P being observed. Therefore, the next lemma and the above discussion imply $Pr(S_i'=S_1|T_P, K) \le 1/l$, thus, Theorem 5.

Lemma 1 $Pr(T_P|K) = Pr(T_P|K, S_i'=S_1)$ for $i \le l$, where S_1 is the most frequent sensitive value in the base table T and T_P is produced by R-suppression. ■

Proof: Recall that K consists of the knowledge about R-suppression and the knowledge on |T|. Let $\Omega(K)$ be the set of all eligible sets w that are produced by R-suppression from an l-eligibility-violated base table with size |T|. All such w satisfy P-Eligibility and l-candidacy because they are produced by R-suppression. Let $\Omega(K, S_i'=S_1)$ be the set of all eligible sets w that are produced by R-suppression given $S_i'=S_1$ on w. Note that $\Omega(K, S_i'=S_1) \subseteq \Omega(K)$, and T_P is in both $\Omega(K)$ and $\Omega(K, S_i'=S_1)$. Without further knowledge, each eligible set in $\Omega(K)$ and $\Omega(K, S_i'=S_1)$ is equally likely, so $Pr(T_P|K) = 1/|\Omega(K)|$ and $Pr(T_P|K, S_i'=S_1)=1/|\Omega(K,S_i'=S_1)|$. If we can show $\Omega(K, S_i'=S_1) \supseteq \Omega(K)$, we have $Pr(T_P|K) = Pr(T_P|K, S_i'=S_1)$.

Consider any eligible set w in $\Omega(K)$. To show that w is in $\Omega(K, S_i'=S_1)$, we need to construct a "plausible" base table for w, say T*(w), where T*(w) violates l-eligibility and $|T^*(w)|=|T|$, such that (i) S_i' is the most frequent sensitive value in T*(w), and (ii) w can be produced by R-suppression from T*(w). The construction of T*(w) is similar to those illustrated in Example 5. T*(w) contains all the records in w, plus $|T|-|w|$ records all having the value S_i'. We claim that T*(w) has the properties (i) and (ii).

In T*(w), S_i' has the frequency $F_i'+|T|-|w|$ and all other values S_j', $j \ne i$, have the same frequency as in w. l-candidacy of w implies $F_i'+|T|-|w|>|T^*(w)|/l$, thus S_i' is a violating value in T*(w). Let $S_1(w)$ denote the most frequent value in T*(w). $S_1(w)=S_i'$. P-Eligibility of w implies that all other values S_j' are not violating in T*(w). This implies part (i) above.

For part (ii), consider the following instance of R-suppression applied to the input T*(w). STEP-1 suppresses $S_1(w)$ to the level $F=F_i'$. This is possible since $i \le l$, there is a non-zero probability of picking $F=F_i'$ in STEP-1 because $F_i' \ge F_{l+1}$. After STEP-1, the result is exactly w because all values in T*(w), except $S_1(w)$, have exactly the same frequency as in w. STEP-2 then returns w immediately because w satisfies both P-Eligibility and l-candidacy. This shows part (ii). □