Practical Inference Control for Data Cubes

Haibing Lu and Yingjiu Li, *Member*, *IEEE*

Abstract—The fundamental problem for inference control in data cubes is how to efficiently calculate the lower and upper bounds for each cell value given the aggregations of cell values over multiple dimensions. In this paper, we provide the first practical solution for estimating exact bounds in two-dimensional irregular data cubes (that is, data cubes in which certain cell values are known to a snooper). Our results imply that the exact bounds cannot be obtained by a direct application of the Fréchet bounds in some cases. We then propose a new approach to improve the classic Fréchet bounds for any high-dimensional data cube in the most general case. The proposed approach improves upon the Fréchet bounds in the sense that it gives bounds that are at least as tight as those computed by Fréchet yet is simpler in terms of time complexity. Based on our solutions to the fundamental problem, we discuss various security applications such as privacy protection of released data, fine-grained access control, and auditing, and identify some future research directions.

Index Terms—Data cube, bound, inference problem.

1 Introduction

Since its introduction, the data cube model [1] has found widespread applications in decision support systems such as online analytic processing (OLAP), data warehousing [2], and data mining [3]. A data cube can be considered a high-dimensional data abstraction that allows one to view aggregated data at different levels.

Fig. 1 illustrates a data cube example with three feature dimensions: agent, time, and stock. The aggregation measure of the data cube is the stock volume. In the core cuboid, each cell has a nonnegative value indicating the stock volume bought by a particular agent at a particular time. Besides the core cuboid, the data cube consists of three two-dimensional (2D) cuboids (denoted by "by stock and agent," "by agent and time," and "by time and stock," respectively), three one-dimensional (1D) cuboids (denoted by "by stock," "by agent," and "by time," respectively), and one zero-dimensional cuboid (the grand total). These cuboids can be computed by aggregating the cell values in the core cuboid across one or more dimensions. In general, an n-dimensional cube is associated with 2^n cuboids. The various cuboids, except the core cuboid, are called star cuboids in this paper.

We consider the following inference problem in data cubes. Assuming that the core cuboid contains sensitive information about each cell but that none of the star cuboids contain sensitive information, can a data snooper infer accurate sensitive information about any cell using the nonsensitive information provided in the star cuboids?

In the above data cube example, each cell in the core cuboid shows which agent has bought which stock at what time and in what volumes. Such information can be considered sensitive, as it reveals each agent's strategy for stock investment. In many cases, the cell values in a core cuboid reveal private information about individuals. For example, in a patient-treatment cube, each cell indicates the number of times that a patient undergoes a certain treatment (for example, for AIDS), which is highly sensitive in real life. In student record management, each cell in the data cube shows the grade a student received for a particular course. The sensitive information in these cases should not be released to the public. However, although the data in a core cuboid must be protected, the aggregated information in a star cuboid is considered nonsensitive in most cases. Thus, the star cuboids can usually be provided to the public for statistical analysis, data mining, and OLAP services.

The inference problem exists since aggregations do not completely protect the sensitive information [4]. It is possible for data snoopers to use the remaining vestiges, together with external knowledge, to infer sensitive information in a core cuboid. Traditional access control [5] cannot capture these inferences, as the aggregations themselves are seemingly innocent. However, limiting such malicious inference of sensitive information is a realistic concern in practice, especially when large data cube products such as a national census or survey are released. This concern is demonstrated by the US Department of Commerce requirement that national statistical offices prevent unauthorized disclosure of sensitive subject-level data when releasing aggregations.

To limit possible disclosure of sensitive information in a data cube, we need to know how accurately a data snooper can estimate the sensitive information. In particular, we need to know how to calculate the lower and upper bounds for each cell value given the aggregation values in the star cuboids. This is the fundamental problem for inference control in data cubes. The lower and upper bounds induced by some fixed set of aggregations are of great importance in measuring the disclosure risk associated with the release of aggregations [6]. In recent years, the problem has become more acute in that applications of the data cube model enable online and query-based accesses to large-scale data sets. Even national statistical offices are moving from periodic releases of static tabulations to online services that provide a large number of users with dynamically updated

- *H. Lu is with the Management Science and Information Systems Department, Rutgers University, 180 University Avenue, Newark, NJ 07102. E-mail: haibing@cimic3.rutgers.edu.*
- *Y. Li is with the School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore 120348. E-mail: yjli@smu.edu.sg.*

Manuscript received 28 June 2006; accepted 17 June 2007; published online 31 July 2007.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-0089-0606.
Digital Object Identifier no. 10.1109/TDSC.2007.70217.

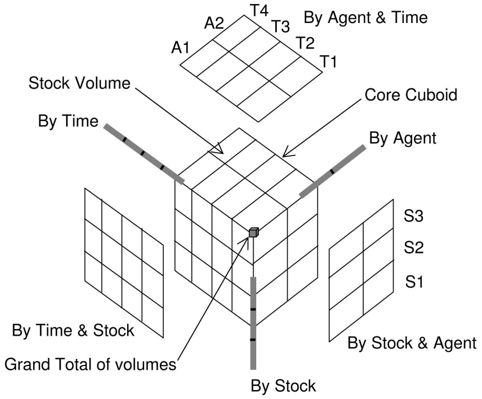


Fig. 1. A data cube example.

data sets. The traditional linear programming approach to inference control would not be efficient in such a scenario.

In this paper, we revisit the Fréchet bounds [7] to solve the inference problem for 2D regular data cubes. The Fréchet bounds of a cell value are first proven to be exact lower and upper bounds. We then propose the first practical solution for estimating exact bounds in 2D irregular data cubes, which are data cubes that contain cell values that are known to a snooper. Our results imply that the exact bounds cannot be obtained by a straightforward extension of the Fréchet bounds in some cases.

We then propose a new approach to improve the classic Fréchet bounds for any high-dimensional data cube in the most general case. The proposed approach improves upon the Fréchet bounds in the sense that it gives no-looser bounds yet is simpler in terms of time complexity. Based on our solutions to the inference problem, we discuss various security applications of our results including privacy protection for released data, fine-grained access control, and auditing.

The rest of this paper is organized as follows: Section 2 formulates the inference problem in data cubes. Section 3 argues why traditional linear programming is highly impractical for solving the inference problem. Section 4 solves the inference problem in two dimensions based on the Fréchet bounds. Section 5 presents our new approach to improve the Fréchet bounds in high-dimensional data cubes. Section 6 discusses various security applications of our results. Section 7 reviews related work. Finally, Section 8 concludes this paper and identifies some future research directions. The appendices provide formal proofs for the theorems proposed in this paper. A six-page extended abstract of this paper appeared in the 2006 IEEE Symposium on Security and Privacy [8]. Added in this complete version are all of the formal proofs and detailed discussions, which are important and represent a major contribution of this paper.

2 Problem Formulation

An n-dimensional data cube C is a collection of cuboids, including a core cuboid and star cuboids, across the spectrum of $n-1$ dimensions to zero dimension. Each dimension i $(1 \leq i \leq n)$ has d_i index values $1, 2, \ldots, d_i$. The core cuboid is an n-dimensional array with $\Pi_{i=1}^{n} d_i$ cell values. Let $a_{t_1 t_2 \cdots t_n}$ be the value at cell $(t_1, t_2, \ldots, t_n)$, where $1 \leq t_i \leq d_i$.

There are $\binom{n}{m}$ $(n-m)$-dimensional star cuboids for data cube C, where $1 \leq m \leq n$. Each $(n-m)$-dimensional star cuboid is an $(n-m)$-dimensional array derived from the core cuboid by aggregating the cell values along m dimensions. The aggregation function is SUM.[1] Let $\{a_{+t_2 \cdots t_n}\}$ be the $(n-1)$-dimensional star cuboid derived by aggregating the cell values along the first dimension. For any meaningful $t_2, \ldots, t_n$, we have $a_{+t_2 \cdots t_n} = \sum_{t_1=1}^{d_1} a_{t_1 t_2 \cdots t_n}$. (There is no ambiguity when "+" is used in subscript; it does not mean a literal addition operation.) Other star cuboids can be denoted similarly.

The inference problem in data cube C is stated as follows: given all $(n-1)$-dimensional *star cuboids, compute the lower and upper bounds for each cell value* $a_{t_1 t_2 \cdots t_n}$ *in the core cuboid.* In mathematical terms, this can be framed as follows: compute the lower and upper bounds for each cell $a_{t_1 \ldots t_n}$ such that $\sum_{t'_i=1}^{d_i} a_{t'_1 \ldots t'_n} = a_{t'_1 \ldots t'_{i-1} + t'_{i+1} \ldots t'_n}$ holds for any $1 \leq i \leq n$ and for any meaningful combination of $t'_1, \ldots, t'_{i-1}, t'_{i+1}, \ldots, t'_n$.

In the formulation of the inference problem, only the aggregations in $(n-1)$-dimensional star cuboids are considered. The reason is that the aggregations in other star cuboids (that is, aggregations of the cell values along two or more dimensions) can be easily derived from those aggregations provided in the $(n-1)$-dimensional star cuboids.

A value $\underline{a}_{t_1 t_2 \cdots t_n}$ is said to be a lower bound of cell value $a_{t_1 t_2 \cdots t_n}$ in data cube C if for any possible core cuboid $\{a'_{t_1 t_2 \cdots t_n}\}$ from which the star cuboids of C can be derived, the inequality $a'_{t_1 t_2 \cdots t_n} \geq \underline{a}_{t_1 t_2 \cdots t_n}$ holds. A value $\underline{a}_{t_1 t_2 \cdots t_n}$ is said to be the exact lower bound of cell value $a_{t_1 t_2 \cdots t_n}$ in data cube C if 1) it is a lower bound and 2) there exists a core cuboid $\{a'_{t_1 t_2 \cdots t_n}\}$ from which the star cuboids of C can be derived and the equality $a'_{t_1 t_2 \cdots t_n} = \underline{a}_{t_1 t_2 \cdots t_n}$ holds. An upper bound or exact upper bound $\overline{a}_{t_1 t_2 \cdots t_n}$ can be defined similarly.

An upper/lower bound a' of cell value a is said to be tighter (no tighter, respectively) than another upper/lower bound a'' of the same cell value a if a' is closer (no closer, respectively) to the exact upper/lower bound of a than a''; otherwise, a' is said to be a no-looser (looser, respectively) bound in comparison with a'', meaning it is at least as tight as a''.

Without loss of generality, we assume throughout this paper that *all cell values* $a_{t_1 t_2 \cdots t_n}$ *in a data cube are nonnegative real numbers.* If this is not the case, one can add an appropriate constant positive value to all cell values so as to transform the data cube to a data cube with nonnegative cell values. After a bound is computed for a transformed cell value in the new data cube, one can subtract the constant value from it in order to get the bound for the original cell value. Note that in the statistical data protection literature, a core cuboid with nonnegative integer cell values is often called a contingency table.

3 The Impracticality of Using Linear Programming

The exact bounds $[\underline{a}_{t_1 t_2 \cdots t_n}, \overline{a}_{t_1 t_2 \cdots t_n}]$ for any cell value $a_{t_1 t_2 \cdots t_n}$ in our inference problem are the solutions to the following two linear programming problems (LPs): 1) $\underline{a}_{t_1 t_2 \cdots t_n} = \min a_{t_1 t_2 \cdots t_n}$, and 2) $\overline{a}_{t_1 t_2 \cdots t_n} = \max a_{t_1 t_2 \cdots t_n}$. Both are subject to linear constraints $\sum_{t'_i=1}^{d_i} a_{t'_1 \ldots t'_n} = a_{t'_1 \ldots t'_{i-1} + t'_{i+1} \ldots t'_n}$ for any $1 \leq i \leq n$ and for any meaningful combination of $t'_1, \ldots, t'_{i-1}, t'_{i+1}, \ldots, t'_n$. The $\sum_{i=1}^{n} d_1 \times \cdots d_{i-1} \times d_{i+1} \times \cdots d_n$

[1] SUM can be extended to AVG provided that the number of cells involved in aggregation is known.

constraints define a nonempty convex feasibility set for the two LPs. According to linear programming theory, there exist optimal solutions $\underline{a}_{t_1t_2\cdots t_n}$ and $\overline{a}_{t_1t_2\cdots t_n}$, and these solutions can be computed in polynomial time.[2]

We argue that linear programming does not scale sufficiently for solving the inference problem for realistic data cubes. One of the most efficient algorithms for linear programming is Karmarkar's algorithm [9], whose time complexity is $O(N^{3.5}L)$, where N is the number of variables, and L is the number of bits required to store the LP in a computer. In the LPs given above, $N = \Pi_{i=1}^{n} d_i$, and $L = O(n\Pi_{i=1}^{n} d_i)$. Thus, the time complexity of solving each LP is $O(n(\Pi_{i=1}^{n} d_i)^{4.5})$, which is prohibitive for processing realistic data cubes. This conclusion has also been drawn by Dobra et al. in [10] by showing a realistic data cube (14-dimensional public survey table) with 4.5 billion cells.

4 Two-Dimensional Data Cubes

In this section, we consider the inference problem for 2D data cubes. Based on the early work of Fréchet [7], it is well known that the following Fréchet bounds are exact for solving the inference problem.

Statement 1 (2D Fréchet bounds). *Given two star cuboids $\{a_{+j}\}$ and $\{a_{i+}\}$ of 2D data cube C, the 2D Fréchet bounds for any cell value a_{ij} in C are*

$$\max\{0, a_{i+} + a_{+j} - a_{++}\} \leq a_{ij} \leq \min\{a_{i+}, a_{+j}\}.$$

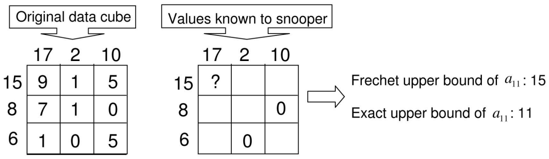
Theorem 4.1 (solution to the inference problem in two dimensions). *Two-dimensional Fréchet bounds are exact.*

A proof sketch of the above theorem was outlined by Cox in [11] via a stepping-stone algorithm. A formal construction proof is presented in Appendix A.

Compared with LP, the Fréchet bounds reduce the time complexity of computing the exact bounds of a cell value from $O((d_1d_2)^{4.5})$ to two addition/subtraction operations and two comparison operations.

Unfortunately, Theorem 4.1 may not hold if some of the cell values are known to a data snooper. This is shown by a counterexample in Appendix B. A snooper may know some of the cell values either because these values are nonsensitive and, thus, not protected or because the snooper has some external knowledge about these cells. For example, in a patient-treatment data set in which each cell indicates the number of times that a patient undergoes a certain treatment, a snooper who is also a patient would know his or her own cell value and may also know some of the other cell values for his or her patient friends. We investigate how to estimate the exact bounds in such a scenario.

Assume that one or more *subcore-cuboids* are known to a snooper. A subcore cuboid is a subset of the cell values defined as $\{a_{ij} \mid i \in S_1, j \in S_2\}$, where $S_1 \subseteq \{1, \ldots, d_1\}$, and $S_2 \subseteq \{1, \ldots, d_2\}$. Then, the inference problem becomes the following: *given all* $(n-1)$-dimensional *star cuboids and a collection of subcore-cuboids, calculate the lower and upper bounds for each cell value in the core cuboid in excess of the union of the given subcore-cuboids.*


Original data cube
Values known to snooper
Frechet upper bound of a_{11}: 15
Exact upper bound of a_{11}: 11

	17	2	10
15	9	1	5
8	7	1	0
6	1	0	5

	17	2	10
15	?		
8			0
6		0	

Fig. 2. Fréchet upper bound is not the exact upper bound in an irregular data cube.

This problem is more generic due to the modeling of unprotected cells and/or a snooper's external knowledge. From the linear programming point of view, this problem is simpler than the original one as it has fewer variables (the number of constraints is not necessarily smaller). However, from the Fréchet bounds point of view, this problem is more difficult to solve since we need to consider what additional information a data snooper may obtain from the known cells.

To solve this problem, we transform it into a normalized form. Let $\{A^k\}$ be the set of subcore-cuboids whose cell values are known to a snooper. Let the core cuboid in excess of the union of the subcore-cuboids be called the *irregular core cuboid.* Let the star cuboids derived from the irregular core cuboid be $\{a_{i+} - \sum_{a_{ij} \in \cup_k A^k} a_{ij}\}$, $\{a_{+j} - \sum_{a_{ij} \in \cup_k A^k} a_{ij}\}$, and $a_{++} - \sum_{a_{ij} \in \cup_k A^k} a_{ij}$. Let the union of the irregular core cuboid and the star cuboids derived from it be called the *irregular data cube.* The *normalized form of the inference problem* is described as follows: Given all $(n-1)$-dimensional star cuboids in an irregular data cube, calculate the lower and upper bounds for each cell value in the irregular core cuboid.

It is clear that the normalized form is equivalent to the original inference problem. (After normalization, the known values can be marked as zeros in the original core cuboid. With no ambiguity, we still use $\{a_{i+}\}$, $\{a_{+j}\}$, and a_{++} to represent the star cuboids in an irregular data cube after normalization.) Consequently, the Fréchet bounds of a_{ij} are still in the form of $\max\{a_{i+} + a_{+j} - a_{++}, 0\} \leq a_{ij} \leq \min\{a_{i+}, a_{+j}\}$ in an irregular data cube. It is easy to verify that the Fréchet bounds after the normalization are no looser than the Fréchet bounds before the normalization. Below, the Fréchet bounds in an irregular data cube always refer to those *after* the normalization.

Lemma 4.2 (exact lower bound in a particular irregular data cube). *Given a 2D irregular data cube, if no cell values in row i or column j are known to a snooper, then the Fréchet lower bound of a_{ij} is exact.*

A construction proof of this lemma is provided in Appendix C. Note that the Fréchet upper bound of a_{ij} may not be exact. Fig. 2 illustrates a counterexample in which two zero values are known to a snooper. In this example, the Fréchet upper bound of a_{11} is 15, whereas the exact upper bound of a_{11} is 11.

In reality, certain cell values in row i or column j may be known to the snooper. Then, Lemma 4.2 cannot be applied for computing the exact lower bound. To improve the lower bound given in Lemma 4.2, we define the *companion cuboid* of a_{ij} to be subcore cuboid $A_{ij} = \{a_{t_1t_2} \mid a_{t_1j}, a_{it_2} \notin \cup_k A^k\}$, where A^k is a collection of subcore-cuboids that are known to a snooper. Then, we have the following:

[2] If the core cuboid consists of integer counts, the LPs become integer programming problems (IPs). Since the feasibility set of IPs is nonempty and finite, there exist optimal solutions in this context as well. An IP usually takes a much longer time to solve than the corresponding LP.

Theorem 4.3 (exact lower bound in an irregular data cube). *Given a 2D irregular data cube, if the sum of all cell values in the companion cuboid of a_{ij} is known to a snooper, then the Fréchet lower bound of a_{ij} in the companion cuboid is exact.*

A construction proof is provided in Appendix D. Note that Theorem 4.3 cannot be proven by a direct application of the Fréchet bounds for two reasons: 1) in the companion cuboid of a_{ij}, one may not know all aggregations except a_{i+} and a_{+j}, and 2) there could be some cells inside the companion cuboid that are known to a data snooper. It might be possible that the first reason gives a snooper less information, whereas the second reason gives more. Regardless of these reasons, Theorem 4.3 asserts that the Fréchet lower bound is still exact.

In the case that the sum of all of the cell values in the companion cuboid of a_{ij} is not known to a snooper, the Fréchet lower bound in the companion cuboid cannot be computed by the snooper. We will show that the Fréchet lower bound in the companion cuboid is at least as tight as the exact lower bound of a_{ij}. An inference auditor who has access to all of the cell values can always calculate the grand total of the companion cuboid and use the Fréchet lower bound in the companion cuboid to estimate the exact lower bound.

Theorem 4.4 (no-looser estimate of the exact lower bound in an irregular data cube). *Given a 2D irregular data cube, if the sum of all cell values in the companion cuboid of a_{ij} is unknown to a snooper, then the Fréchet lower bound of a_{ij} in the companion cuboid is at least as tight as the exact lower bound of a_{ij}.*

A formal proof is given in Appendix E. Although Theorem 4.4 gives a no-looser estimate of the exact lower bound in any irregular data cube, we now propose a no-tighter estimate of the exact lower bound. First, we define two *companion sums* of a_{ij} to be $c_{ij}^1 = \sum_{t_2}\{a_{+t_2} \mid a_{it_2} \notin \cup_k A^k\}$ and $c_{ij}^2 = \sum_{t_1}\{a_{t_1+} \mid a_{t_1 j} \notin \cup_k A^k\}$, where $\{A^k\}$ is a collection of subcore-cuboids that are known to a snooper (note that a snooper can calculate the companion sums from released aggregations). Then, we have the following:

Theorem 4.5 (no-tighter estimate of the exact lower bound in an irregular data cube). *Given a 2D irregular data cube, if the sum of all cell values in the companion cuboid of a_{ij} is unknown to a snooper, then* $\max\{a_{+j} + a_{i+} - c_{ij}^1, a_{+j} + a_{i+} - c_{ij}^2, 0\}$ *is a lower bound of a_{ij}.*

A formal proof is given in Appendix F. Since the companion sums are less than the grand total, the lower bound proposed in Theorem 4.5 is no looser than the Fréchet lower bound (after normalization). Thus, the bounds proposed in Theorems 4.3 and 4.4 are also no looser than the Fréchet lower bound.

So far, our study has been primarily focused on estimating the exact lower bound in an irregular data cube. In inference control, a frequently asked question is whether a particular cell value is greater than zero (for example, a patient is HIV positive) or greater than a threshold (for example, an agent buys a large-enough volume of stock). Estimating the exact lower bound of a cell value is the most useful way to answer such questions. To estimate the exact upper bound in an irregular data cube, one can use the Fréchet upper bound (after normalization) and further improve it by using the shuttle algorithm (see Section 5.2) based on the estimates of the exact lower bound provided above.

5 High-Dimensional Data Cubes

For regular data cubes, the Fréchet bounds have been extended to n dimensions [12].

Statement 2 (*n-dimensional* Fréchet bounds). *In an* n-dimensional *data cube, the Fréchet lower bound for any cell $a_{t_1\cdots t_n}$ equals the maximum of zero and the $\binom{n}{2}$ possible 2D Fréchet lower bounds:*

$$\max\left\{\begin{array}{l} 0, a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} + a_{t_1\cdots t_{j-1}+t_{j+1}\cdots t_n} - \\ a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_{j-1}+t_{j+1}\cdots t_n} \mid 1 \le i < j \le n \end{array}\right\},$$

and the Fréchet upper bound of $a_{t_1\cdots t_n}$ is the minimum of n aggregation values in the $(n-1)$-dimensional *star cuboids to which the cell value contributes*

$$\min\{\, a_{t_1} \cdots t_{i-1} + t_{i+1} \cdots t_n \mid 1 \le i \le n \,\}.$$

In particular, the three-dimensional Fréchet bounds for cell value a_{ijk} are

$$\max\left\{\begin{array}{l} 0 \\ a_{+jk} + a_{i+k} - a_{++k} \\ a_{+jk} + a_{ij+} - a_{+j+} \\ a_{i+k} + a_{ij+} - a_{i++} \end{array}\right\} \le a_{ijk} \le \min\left\{\begin{array}{c} a_{+jk} \\ a_{i+k} \\ a_{ij+} \end{array}\right\}.$$

The time complexity of computing the n-dimensional Fréchet bounds for each cell value is $O(\binom{n}{2}) = O(n^2)$. This complexity is significantly lower than the complexity $O((\sum_{i=1}^n d_i)^{4.5})$ of using linear programming to compute the exact bounds.

Unfortunately, the Fréchet bounds may not be exact for any high-dimensional data cube in general (there are special cases based on decomposibility into graph structures, as discussed in Section 7). This has been proven by Cox [11] with counterexamples. Below, we propose a formulation of new bounds that are no looser than the Fréchet bounds in the most general case and whose time complexity is simpler than that of the Fréchet bounds. Our bounds are also no looser than the recent improvements on the Fréchet bounds in high dimensions (see Section 5.2).

Statement 3 (*n-dimensional* new bounds). *Given the star cuboids of an* n-dimensional *data cube C, the new lower bound for any cell value $a_{t_1\cdots t_n}$ in C is*

$$\max\left\{\begin{array}{l} 0, a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \\ \sum_{t\neq t_i} \min\{a_{+t_2\cdots t_{i-1}tt_{i+1}\cdots t_n}, a_{t_1+t_3\cdots t_{i-1}tt_{i+1}\cdots t_n}, \\ \cdots a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_{n-1}+}\} \mid 1 \le i \le n \end{array}\right\}.$$

Let $\underline{a}_{t_1\cdots t_n}$ be the new lower bound of $a_{t_1\cdots t_n}$. The new upper bound of $a_{t_1\cdots t_n}$ is

$$\min\left\{\begin{array}{l} a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \\ \sum_{t\neq t_i} \underline{a}_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n} \mid 1 \le i \le n \end{array}\right\}.$$

In particular, the 3D new bounds for cell value a_{ijk} are

$$\underline{a}_{ijk} = \max\left\{\begin{array}{l} 0 \\ a_{+jk} - \sum_{t\neq i} \min\{a_{t+k}, a_{tj+}\} \\ a_{i+k} - \sum_{t\neq j} \min\{a_{+tk}, a_{it+}\} \\ a_{ij+} - \sum_{t\neq k} \min\{a_{+jt}, a_{i+t}\} \end{array}\right\},$$

$$\overline{a}_{ijk} = \min\left\{\begin{array}{l} a_{+jk} - \sum_{t\neq i} \underline{a}_{tjk} \\ a_{i+k} - \sum_{t\neq j} \underline{a}_{itk} \\ a_{ij+} - \sum_{t\neq k} \underline{a}_{ijt} \end{array}\right\}.$$

Theorem 5.1 (comparing new bounds with Fréchet bounds). *The new bounds are at least as tight as the Fréchet bounds in n dimensions.*

The proof is given in Appendix G. It is not difficult to prove that the new bounds are the same as the Fréchet bounds in two dimensions. We leave this as an exercise.

Note that the new bounds can be directly applied to irregular data cubes following the normalization process shown in Section 4. An alternative approach is to resort to the high-dimensional Fréchet bounds. Recall that in n dimensions, the Fréchet bounds are derived from $\binom{n}{2}$ 2D Fréchet bounds, each of which can be computed using the method proposed in Section 4. However, this approach is more complex than the application of our new bounds.

5.1 Complexity Reduction

At first glance, computing our new bounds appears to be more complex than computing the Fréchet bounds. If computed in a straightforward manner, the new lower bound for each cell requires $n-2$ comparison operations to compute each "min," $(d_i-1)(n-2)$ comparison operations and (d_i-2) addition operations to get each "$\sum$"; thus, it requires $(\sum_i d_i - n)(n-2)+n$ comparison operations and $\sum_i d_i - n$ addition and subtraction operations to get the final "max." The time complexity[3] of computing the new lower bound in this way is $O(n\sum_{i=1}^{n} d_i)$. After all of the lower bounds are obtained, the new upper bound for each cell can be computed in $\sum_{i=1}^{n} d_i - n$ addition and subtraction operations and $n-1$ comparison operations. Since the computation of the upper bound depends on the lower bounds, its complexity is also $O(n\sum_{i=1}^{n} d_i)$. In comparison, the time complexity of computing the Fréchet bounds (which is dominated by computing the lower bound) is $O(n^2)$.

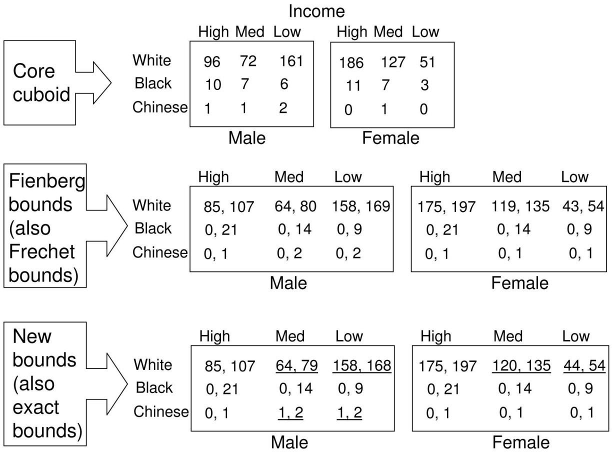
However, one can reduce the time complexity of computing the new bounds by precomputation and transformation. Let $\breve{a}_{t_1\cdots t_n} = \min\{a_{+t_2\cdots t_n}, a_{t_1+t_3\cdots t_n}, \ldots, a_{t_1\cdots t_{n-1}+}\}$. We have the following:

Theorem 5.2 (transformation of the new lower bound). *The new lower bound for cell value $a_{t_1\cdots t_n}$ can be transformed as*

$$\max\{0, a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \sum_{t\neq t_i} \breve{a}_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n} \mid 1 \leq i \leq n\}.$$

The proof is given in Appendix H. According to this theorem, one can precompute all $\breve{a}_{t_1\cdots t_n}$ before computing the new bounds. During this process, each cell requires at most $n-1$ comparison operations. The computation of the new lower bound for each cell requires $\sum_{i=1}^{n} d_i - n$ addition and subtraction operations and n comparison operations. After all of the lower bounds are obtained, each new upper bound requires $\sum_{i=1}^{n} d_i - n$ addition and subtraction operations and $n-1$ comparison operations. The time complexity of computing the new bounds in this manner is thus $O(\sum_{i=1}^{n} d_i)$.

This complexity $O(\sum_{i=1}^{n} d_i)$ is not only much simpler than that of linear programming $O(n(\Pi_{i=1}^{n} d_i)^{4.5})$ but also simpler than that of the Fréchet bounds $O(n^2)$ in the case that d_i is bounded. In real-world applications, a data cube is usually built from a database relation with a large number of attributes (it is common to see tens or hundreds of attributes in applications); however, the number of categories (that is, d_i) for each attribute is usually bounded (certain attributes such as binary attributes have very small d_i). In such cases, the time complexity of our new bounds is linear to n, whereas the Fréchet bounds are quadratic.


Income
Core cuboid
Fienberg bounds (also Frechet bounds)
New bounds (also exact bounds)
Male
Female

Core cuboid

	Male High	Male Med	Male Low	Female High	Female Med	Female Low
White	96	72	161	186	127	51
Black	10	7	6	11	7	3
Chinese	1	1	2	0	1	0

Fienberg bounds (also Frechet bounds)

	Male High	Male Med	Male Low	Female High	Female Med	Female Low
White	85, 107	64, 80	158, 169	175, 197	119, 135	43, 54
Black	0, 21	0, 14	0, 9	0, 21	0, 14	0, 9
Chinese	0, 1	0, 2	0, 2	0, 1	0, 1	0, 1

New bounds (also exact bounds)

	Male High	Male Med	Male Low	Female High	Female Med	Female Low
White	85, 107	64, 79	158, 168	175, 197	120, 135	44, 54
Black	0, 21	0, 14	0, 9	0, 21	0, 14	0, 9
Chinese	0, 1	1, 2	1, 2	0, 1	0, 1	0, 1

Fig. 3. Comparison of bounds using Fienberg's example [13, Table 1].

5.2 Comparisons with Other Solutions

In recent years, rigorous efforts have been made to improve the Fréchet bounds in high dimensions. Most of the improvements take place in three dimensions, although some of them can be extended to n dimensions. In this section, we compare our new bounds with the recent improvements, including Fienberg's bounding approach [13], Chowdhury et al.'s network models for bounds [14], Buzzigoli and Giusti's shuttle algorithm [15], and Dobra and Fienberg's generalized Fréchet bounds [6], [10], [16]. A review of most of these methods was given by Cox in [12].

5.2.1 *Fienberg's Bounding Approach*

Fienberg's bounding approach works in three dimensions [13]. As correctly pointed out by Cox in [12], the lower bound provided by Fienberg is equivalent to the Fréchet lower bound, whereas the upper bound (also called the Bonferroni upper bound of Fienberg) is no looser

$$\begin{aligned} a_{ijk} &\leq (\text{Fienberg bound}) \min\{a_{+jk}, a_{i+k}, a_{ij+}, \\ &\quad a_{+++} - a_{i++} - a_{+j+} - a_{++k} + \\ &\quad a_{ij+} + a_{i+k} + a_{+jk}\} \\ &\leq (\text{Fréchet bound}) \min\{a_{+jk}, a_{i+k}, a_{ij+}\}. \end{aligned}$$

Theorem 5.3 (comparing new bounds with Fienberg bounds). *The new bounds are at least as tight as the Fienberg bounds in three dimensions.*

The proof is given in Appendix I. The above theorem can be illustrated using the example shown in Fig. 3. The core cuboid in this example is a $3\times 3\times 2$ table of sample counts taken from the 1990 Decennial Census Public Use Sample. This example has also been used by Fienberg [13] and Cox [12] for comparing bounds. In this example, the Fienberg bounds are exactly the same as the Fréchet bounds.[4] In comparison,

[3] The time complexity is derived solely based on the number of addition, subtraction, or comparison operations. We do not address issues such as data structure, memory cost, and I/O cost in this paper.

[4] Certain numeric errors in [13] regarding this example have been pointed out and corrected by Cox in [12].

our new bounds are the same as the exact bounds and tighter than the Fienberg bounds for certain cells.

5.2.2 Network Models for Bounds

Chowdhury et al. [14] presented network models for computing the exact bounds for integer cells in three dimensions. The network models provide a natural language to express 2D tables (or 2D star cuboids) and an efficient mechanism to compute the exact bounds.

The problem addressed in [14] is, assuming that one 3D core cuboid and one of its three 2D star cuboids are protected, how to calculate the exact lower bound and upper bound for each aggregation value in the protected star cuboid, given the other two star cuboids. Chowdhury et al. constructed networks for expressing the connections between the star cuboids and proposed two simple matrix operators for obtaining the exact bounds. Although the method is very efficient, it deals with 2D star cuboids only. Cox's comments [12] on this method are that "most generalizations beyond two dimensions are apt to fail" and that the problem can be solved directly using the Fréchet bounds without recourse to networks.

5.2.3 Shuttle Algorithm

The shuttle algorithm is an iterative algorithm proposed by Buzzigoli and Giusti [15]. The basic idea is that for each cell value in three dimensions and each 2D aggregation containing the cell, a candidate lower bound is computed by subtracting from the aggregation the sum of the current upper bounds of all of the other cells contained in the aggregation. If the candidate lower bound improves the current lower bound, it replaces it. A similar procedure is used to improve the current upper bound with a candidate computed from the sum of the current lower bounds. The two-step procedure is repeated until the lower bounds or upper bounds for all of the cells are stationary. The shuttle algorithm can be easily extended to n dimensions.

The shuttle algorithm can work with any initial lower and upper bounds. The initial lower and upper bounds could be chosen from the Fréchet bounds, the Fienberg bounds, or our new bounds. In this sense, the shuttle algorithm is complementary to our work. Cox has correctly pointed out in [12] that the shuttle algorithm converges in a finite number of iterations if all of the cell values are integers. However, it is not clear how fast the algorithm converges. The time complexity of this algorithm is at least as high as the algorithm used for providing the initial bounds. A generalized version of this algorithm was developed by Dobra et al. [10].

5.2.4 Dobra and Fienberg's Generalized Fre´chet Bounds

Dobra and Fienberg [6], [10], [16] studied exact lower and upper bounds, which they called generalized Fréchet bounds, for a specific type of high-dimensional statistical tables. A statistical table can be considered a data cube in which a nonnegative random variable is assigned to each cell. They assumed that the released set of marginals (that is, values in star cuboids) is the set of minimum sufficient statistics of a decomposible or reducible log-linear model. Under such an assumption, the exact lower and upper bounds of each cell can be expressed as explicit functions of relating marginals.

The difference between our work and Dobra and Fienberg's is clear. Since we do not make any assumption about the distribution of cell values, our results can be applied to *any* data cube in the most general case, regardless of the distribution of cell values. In contrast, Dobra and Fienberg's results pertain only to the reducible log-linear models with minimal sufficient statistics. Their results "represent only a small part of those needed to allow the computation of upper and lower bounds [...]" [16]. In a recent development, Dobra et al. [10] presented a hash-table-based structure and a generalized shuttle algorithm to exploit the extreme sparsity of large data sets.

6 Discussions on Security Applications

In this section, we discuss some security applications based on the solutions to the inference problem in data cubes.

6.1 Privacy Protection for Released Data

Privacy protection for released data has been a major concern in many applications such as statistical data publication, survey, and data mining. This concern is about how to preserve an individual's privacy in subject-level data when aggregation data is released.

We consider data cubes in this application scenario (for example, data cube products such as a national census or survey are released). When data aggregations are released, it is critical to ensure that the released data cannot be utilized by data snoopers to obtain privacy information. We classify the disclosure of privacy information into the following types based on what the privacy information means:

- *Existence disclosure.* The lower bound of a cell value is greater than zero (for example, a patient visits a doctor at least one time for a certain disease).
- *Threshold upward disclosure.* The lower bound of a cell value is greater than a certain threshold (for example, an agent buys a large-enough volume of certain stock).
- *Threshold downward disclosure.* The upper bound of a cell value is less than a certain threshold (for example, an agent does not buy a large-enough volume of certain stock).
- *Approximation disclosure.* The difference of the upper bound and lower bound of a cell value is less than a certain threshold (for example, a professor's salary falls into a small-enough range).

The existence and threshold upward disclosures are determined by the lower bounds that a snooper can infer, whereas the threshold downward disclosure and the approximation disclosure involve the upper bounds of protected cell values.

For any type of disclosure, we can determine which cells are subject to disclosure according to the exact bounds that a snooper may obtain (for example, through LP). There will be no mistakes in determining the cells if we use the exact bounds. If the no-tighter bounds are used instead, there might be false negatives (cells subject to disclosure are considered subject to no disclosure) but no false positives (cells subject to no disclosure are considered subject to disclosure). If we use no-looser bounds, it may lead to false positives but no false negatives.

Given a set $A^0 = \{a_{t'_1 \cdots t'_n}\}$ of cells that might be subject to disclosure, we now propose a generic approach, called *k-anonymity partition*, to limit the disclosure of those cells. Define the projection of A^0 to each dimension i as $P_i = \{t'_i\}$. Assume that $|P_i| = \min\{|P_1|, \ldots, |P_n|\}$, and $0 < k \leq d_i$. The *k-anonymity partition* from dimension i is defined by the following procedures:

- Partition the values in P_i into groups of k values. If $|P_i| \geq k$, then the last group may consist of more than k values (for simplicity, we describe our method only for the groups of k values). If $|P_i| < k$, then $k - |P_i|$ values from $D_i - P_i$ are combined with the values in P_i to form a group, where $D_i = \{1, \ldots d_i\}$ is the set of index values for dimension i.
- For each group of k values $t_i^1, \ldots t_i^k$ and for each dimension $j \neq i$ (without loss of generality, assume $j > i$), release the aggregations of sum values $a_{t_1 \cdots t_i^1 \cdots t_{j-1} + t_{j+1} \cdots t_n} + \ldots + a_{t_1 \cdots t_i^k \cdots t_{j-1} + t_{j+1} \cdots t_n}$ instead of individual sums $a_{t_1 \cdots t_i^1 \cdots t_{j-1} + t_{j+1} \cdots t_n}, \ldots, a_{t_1 \cdots t_i^k \cdots t_{j-1} + t_{j+1} \cdots t_n}$ in the star cuboid $\{a_{t_1 \cdots t_{j-1} + t_{j+1} \cdots t_n}\}$. In other words, any k values $a_{t_1 \cdots t_i^1 \cdots t_n}, \ldots, a_{t_1 \cdots t_i^k \cdots t_n}$ are summed together in all $(n-1)$-dimensional star cuboids. Other star cuboids can be processed similarly if they are released to the public.

From the released star cuboids only, a snooper cannot differentiate among any k values $a_{t_1 \cdots t_i^1 \cdots t_n}, \ldots, a_{t_1 \cdots t_i^k \cdots t_n}$. Now, consider any cell $a_{t'_1 \cdots t'_n}$ that might be subject to disclosure before k-anonymity partition (that is, $a_{t'_1 \cdots t'_n} \in A^0$). Since $t'_i \in P_i$, there exists a set of k values in the form of $a_{t_1 \cdots t_i^1 \cdots t_n}, \ldots, a_{t_1 \cdots t_i^k \cdots t_n}$ such that 1) $a_{t'_1 \cdots t'_n}$ is one of these k values and 2) these k values are always summed together in all star cuboids. Therefore, $a_{t'_1 \cdots t'_n}$ cannot be differentiated from a group of k cells after k-anonymity partition. A k-anonymity protection is thus achieved for those cells that might be subject to disclosure at the price of reducing the number of aggregated values that are released in the star cuboids.

Let us consider what a snooper can infer for each group of k values after the k-anonymity partition. Assume that the snooper can infer $a_{t'_1 \cdots t'_n} > \tau$ for existence or threshold upward disclosure before the k-anonymity partition, where $\tau \geq 0$ is a predetermined threshold. After the k-anonymity partition, the snooper can, at best, infer that $a_{t_1 \cdots t_i^1 \cdots t_n} + \ldots + a_{t_1 \cdots t_i^k \cdots t_n} > \tau$. The snooper cannot infer any of the k values having a nonzero lower bound. Thus, all k values are safe from existence and threshold disclosures.

For threshold downward disclosure and approximation disclosure, however, the inference of a group of k values is determined by its upper bound. Generally, assume that a snooper can obtain $\tau_1 \leq a_{t_1 \cdots t_i^1 \cdots t_n} + \ldots + a_{t_1 \cdots t_i^k \cdots t_n} \leq \tau_2$ after the k-anonymity partition; then, the snooper can infer that all of these k values are in the range of $[0, \tau_2]$. If τ_2 is small enough, it may be considered a disclosure. In such case, one can choose large-valued sums in the partition or increase k so as to increase the upper bounds.

6.2 Fine-Grained Access Control and Auditing

If the aggregations in a data cube are not to be released for public access, fine-grained access control and auditing can be applied for protecting privacy information when users query the data cube on the server side. In this scenario, a user may be granted to access certain cell values and/or aggregations values provided that no privacy information is revealed from these values.

We assume that appropriate authentication is enforced when a user queries the data cube. For each user, a subset of cell values is defined as privacy information. The three types of disclosure defined in the above section can still be used to describe the leakage of privacy information.

To ensure that the server only answers those queries that do not reveal any privacy information, an auditing monitor is implemented to keep recording all of the queries that have been asked by and answered to each user. The auditing monitor should not be bypassed or tampered with for the integrity of auditing records. When a user constructs a new set of queries, fine-grained access control is implemented to check whether the answers to this set of queries, combined with historical auditing records, reveal any privacy information. If not, grant the access request; otherwise, deny it.

The fine-grained access control can be easily performed with an application of our results. The first reason is that the bounds in our results can be computed in the presence of known cell values (thus, we do not need to resort to LP). The second reason is that the bounds of a cell can be computed with a minimum number of aggregation values (instead of all aggregation values in LP). As a result, the server can quickly locate the relevant cells and compute their bounds given a set of known cell values and aggregation values. The last reason is that the high efficiency of our method is critical for enforcing the access control in an online environment.

This access control is fine grained because it deals with ad hoc sets of cell/aggregation values. In comparison, the previous access/inference control method proposed for data cubes [17] deals with cuboids or slices of data as authorization objects. The previous method [17] derives privacy breaches based on the logical relationships among authorization objects, rather than the bounds of underlying cell values. Due to these differences, their method is complementary to ours.

7 Related Work

Although the need for security protection in data cubes has been identified [18], the fundamental problem of inference control, which is how to efficiently calculate the lower and upper bounds for each cell given the aggregations, has not yet been fully addressed. A special case of this problem, the inference of exact values (that is, the lower bounds and upper bounds are the same) in data cubes, has been studied recently [19], [20], [21]. In [19], Brankovic et al. gave the maximum number of queries that can be answered without compromising any previously unknown values in a data cube. In [20], Wang et al. gave a tight upper bound for the number of known values such that a data cube is inference-free. In [21], it is proven that even queries (that is, where an even number of cell values are involved in multidimensional axis-parallel cuboids) are not subject to exact inferences. In comparison, we address a more generic and practical problem regarding the inference of bounds rather than exact values in data cubes.

In the context of statistical databases, inference control (or privacy protection) has been extensively studied [22], [23], [24]. The proposed techniques can be roughly classified into perturbation based and restriction based. The perturbation-based techniques protect data against possible disclosure by adding random noises to source data [25], [26], [27], [28], [29],

[30], query answers [31], or database structures [32]. Since these techniques inevitably introduce errors, they may not be appropriate for certain applications.

The restriction-based techniques limit possible disclosure by posing restrictions on queries and/or source data. The advantage of this approach is that it does not introduce any errors. The trade-off is that it may reduce the amount of information that is provided for data services. Our k-anonymity partition method falls into this category. It borrows the k-anonymity concept proposed by Samarati and Sweeney [33], [34] for protecting microdata (individual respondent data). We extend it for protecting cell values in data cubes. Our k-anonymity partition also borrows ideas from the partition approach [35], [36], in which individual entities are clustered into a number of mutually exclusive subsets (called atomic populations). The difference is that our method partitions the sum values in the star cuboids rather than the individual values at the lowest level. Our method is similar to the microaggregation approach [23], [37] in the sense that certain sum values are clustered into mutually exclusive groups prior to publication. The difference is that our method clusters only a selected set of sum values, whereas the microaggregation approach clusters all individual records and then publishes the average over each group instead of the individual records. Another related work is cell suppression [11], [38], [39], [40], in which all cell values that might cause disclosure are suppressed either fully or partially from a released table(s). In comparison, we do not suppress any cell values but aggregate selected sum values in the star cuboids.

8 Conclusions and Future Directions

Data cubes, including those related to data warehouses, data mining, and OLAP, are important decision-support tools for business and scientific applications. Data cubes can be used to discover trends and patterns in a multidimensional and multilevel manner. Although data cubes restrict user access to predefined aggregations, an inappropriate inference of sensitive or private information about cell values may still occur. To protect the data, it is critical to discover such disclosure effectively and efficiently.

The main purpose of this paper is to provide practical solutions for calculating the lower and upper bounds for each cell value given the aggregations in a data cube. The lower and upper bounds tell us to what extent a data snooper can compromise the protected values. Although this problem can be solved using linear programming, the time complexity of this solution makes it prohibitive in practice.

The same problem has been studied using different forms and terms in statistical data protection and statistical databases. The best method for finding practical solutions to this problem is one that was formulated by Fréchet in 1940, providing exact lower and upper bounds (Fréchet bounds) in the 2D case. We advance the concept of Fréchet bounds by contributing the following:

- We provide the first practical solution for estimating the lower and upper bounds in 2D irregular data cubes. Our results can be considered a nontrivial extension of the Fréchet bounds in irregular data cubes. In particular, we give the exact lower bound for each cell value and no-tighter and no-looser estimates of the exact lower bound, all of which are at least as tight as a straightforward extension of the Fréchet lower bound (after normalization) in irregular data cubes. The upper bound for each cell value is the same as the Fréchet upper bound (after normalization), and it may be improved through the application of the shuttle algorithm based on our lower bounds.
- We provide the first improvement of the Fréchet bounds in arbitrary n dimensions for any nonnegative data cubes. We prove that our new bounds for each cell in n dimensions are at least as tight as the n-dimensional Fréchet bounds and that the time complexity of our approach can be reduced to be linear in terms of the total number of indices in all dimensions. In contrast, the Fréchet bounds are quadratic in terms of the total number of dimensions. We also compare our new bounds with recent improvements of the Fréchet bounds. In particular, we prove that our bounds are at least as tight as the Fienberg bounds, that they provide a good starting point for the shuttle algorithm, and that they are more generic than the network models for bounds and the generalized Fréchet bounds.
- Based on the bounds that a data snooper can obtain for each cell, we discuss two security applications including privacy protection for released data and fine-grained access control and auditing. We classify the disclosure of privacy information into three types and propose a k-anonymity partition method to protect the privacy information.

Our ongoing work includes an extension to dynamic data cubes in which the cell values may be frequently updated over time. For dynamic data cubes, new issues arise, including but not limited to disclosure about which cells have been updated and to what extent they have been updated. It would also be interesting to develop practical algorithms for computing exact bounds for large sparse data cubes.

Appendix A

Proof of Theorem 4.1

Lemma 1.1. *Given two sets of nonnegative values $\{a_{+j}\}$ and $\{a_{i+}\}$ that satisfy the consistency condition $\sum_j a_{+j} = \sum_i a_{i+} = a_{++}$, there exists a 2D (nonnegative) core cuboid $\{a_{ij}\}$ such that $\{a_{+j}\}$ and $\{a_{i+}\}$ are star cuboids of it.*

Proof. A construction proof is provided. Consider two cases: 1) $a_{+1} + a_{1+} - a_{++} \geq 0$ and 2) $a_{+1} + a_{1+} - a_{++} < 0$.

For Case 1, choose $a_{11} = a_{+1} + a_{1+} - a_{++}$ and

$$\begin{cases} a_{1j} = a_{+j} & (j \neq 1), \\ a_{i1} = a_{i+} & (i \neq 1), \\ a_{ij} = 0, & (i \neq 1, j \neq 1). \end{cases}$$

The 2D (nonnegative) core cuboid $\{a_{ij}\}$ constructed this way can derive star cuboids $\{a_{+j}\}$ and $\{a_{i+}\}$.

For Case 2, choose $a_{11} = 0$. Due to the consistency condition, there must exist $\{a_{1j}\}_{j \neq 1}$ and $\{a_{i1}\}_{i \neq 1}$ such that $a_{1j} \leq a_{+j}$, $a_{i1} \leq a_{i+}$, and

$$\begin{cases} \sum_{j \neq 1} a_{1j} = a_{1+}, \\ \sum_{i \neq 1} a_{i1} = a_{+1}. \end{cases}$$

Thus, the cell values in the first row and the first column are determined in the core cuboid that is to be

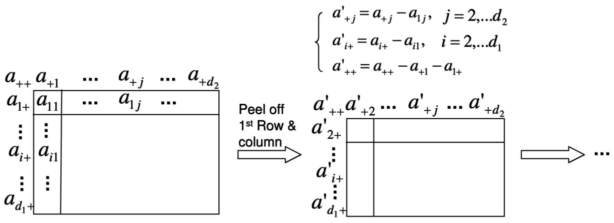

$a'_{+j} = a_{+j} - a_{1j}, \quad j = 2, \ldots d_2$
$a'_{i+} = a_{i+} - a_{i1}, \quad i = 2, \ldots d_1$
$a'_{++} = a_{++} - a_{+1} - a_{1+}$
a_{++} a_{+1} ... a_{+j} ... a_{+d_2}
a_{1+} a_{11} ... a_{1j} ...
a_{i+} a_{i1}
a_{d_1+}
Peel off 1st Row & column
a'_{++} a'_{+2} ... a'_{+j} ... a'_{+d_2}
a'_{2+}
a'_{i+}
a'_{d_1+}
...

Fig. 4. Constructing a 2D core cuboid in Lemma 1.1.

constructed. Peeling off the first row and column, a smaller 2D core cuboid is to be constructed with the revised star cuboids (see Fig. 4)

$$\begin{cases} a'_{+j} = a_{+j} - a_{1j}, & j = 2, \ldots, d_2, \\ a'_{i+} = a_{i+} - a_{i1}, & i = 2, \ldots, d_1, \\ a'_{++} = a_{++} - a_{+1} - a_{1+}. \end{cases}$$

These revised star cuboids still satisfy the consistency condition. Therefore, the above process can be applied recursively. In any recursive step, if Case 1 happens, the construction stops; otherwise, the construction process continues until the last row or column is peeled off. Finally, a 2D nonnegative core cuboid $\{a_{ij}\}$ is constructed, from which the star cuboids $\{a_{+j}\}$ and $\{a_{i+}\}$ can be derived. □

Proof of Theorem 4.1. Without loss of generality, the theorem is proven for cell value a_{11} only.

First, prove that $\min\{a_{+1}, a_{1+}\}$ is the exact upper bound of a_{11}. Since all cell values are nonnegative, $\min\{a_{+1}, a_{1+}\}$ is an upper bound of a_{11}. To prove that it is the exact upper bound, one needs to prove that there exists a core cuboid $\{a'_{ij} \geq 0\}$ such that 1) $a'_{11} = \min\{a_{+1}, a_{1+}\}$ and 2) the star cuboids $\{a_{i+}\}$ and $\{a_{+j}\}$ can be derived from it. Without loss of generality, assume that $a_{+1} \leq a_{1+}$. Let $a'_{11} = \min\{a_{+1}, a_{1+}\} = a_{+1}$ and $a'_{i1} = 0$ for $i \neq 1$. The first column in the core cuboid $\{a'_{ij}\}$ is thus constructed. Peeling off this first column, a smaller 2D core cuboid is to be constructed with revised aggregation values: $a'_{1+} = a_{1+} - a_{+1}$, $a'_{++} = a_{++} - a_{+1}$, $a'_{+j} = a_{+j}$ for $j = 2, \ldots, d_2$, and $a'_{i+} = a_{i+}$ for $i = 2, \ldots, d_1$. These aggregation values satisfy the consistency condition. From Lemma 1.1, a nonnegative core cuboid $\{a'_{ij}\}$ can be constructed with these aggregation values. Combining this core cuboid with the peeled-off column, one obtains the required core cuboid.

Then, prove that $\max\{0, a_{1+} + a_{+1} - a_{++}\}$ is the exact lower bound of a_{11}. From $a_{11} + a_{12} + \ldots + a_{1d_2} = a_{1+}$ and $a_{1i} \leq a_{+i}$, one can derive $a_{11} \geq a_{1+} - (a_{+2} + a_{+3} + \ldots + a_{+d_2}) = a_{+1} + a_{1+} - a_{++}$. Thus, $\max\{0, a_{1+} + a_{+1} - a_{++}\}$ is a lower bound of a_{11}. To prove that it is the exact lower bound, one needs to prove that there exists a core cuboid $\{a'_{ij} \geq 0\}$ such that 1) $a'_{11} = \max\{0, a_{1+} + a_{+1} - a_{++}\}$ and 2) the star cuboids $\{a_{i+}\}$ and $\{a_{+j}\}$ can be derived from it. The proof of this is exactly the same as that of Lemma 1.1. □

Appendix B

Theorem 4.1 May Not Hold in Irregular Data Cubes

We show that Theorem 4.1 may not hold in irregular data cubes. Consider the simple example shown in Fig. 5. In this example, a single subcore-cuboid A^0 is known to a snooper, whereas the other three subcore-cuboids A^1, A^2, and A^3 are protected. If the Fréchet bounds are directly applied to a cell value $a_{ij} \in A^1$, then

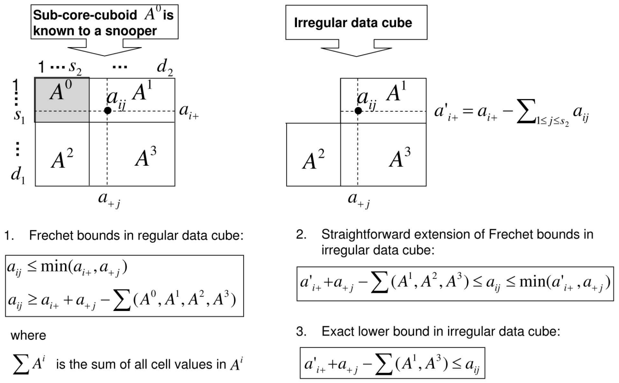

Sub-core-cuboid A^0 is known to a snooper
$1 \cdots s_2$ ... d_2
1
s_1
d_1
A^0
a_{ij}
A^1
a_{i+}
A^2
A^3
a_{+j}
Irregular data cube
a_{ij}
A^1
$a'_{i+} = a_{i+} - \sum_{1 \leq j \leq s_2} a_{ij}$
A^2
A^3
a_{+j}
1. Frechet bounds in regular data cube:
$a_{ij} \leq \min(a_{i+}, a_{+j})$
$a_{ij} \geq a_{i+} + a_{+j} - \sum(A^0, A^1, A^2, A^3)$
where
$\sum A^i$ is the sum of all cell values in A^i
2. Straightforward extension of Frechet bounds in irregular data cube:
$a'_{i+} + a_{+j} - \sum(A^1, A^2, A^3) \leq a_{ij} \leq \min(a'_{i+}, a_{+j})$
3. Exact lower bound in irregular data cube:
$a'_{i+} + a_{+j} - \sum(A^1, A^3) \leq a_{ij}$

Fig. 5. Example of an irregular data cube.

$$a_{i+} + a_{+j} - a_{++} = a_{i+} + a_{+j} - \sum(A^0, A^1, A^2, A^3)$$
$$\leq a_{ij} \leq \min\{a_{i+}, a_{+j}\},$$

where $\sum A^k$ denotes the sum of all cell values in sub-core cuboid A^k ($k = 0$, 1, 2, or 3). These bounds may not be the exact bounds due to the existence of no-looser bounds

$$a'_{i+} + a_{+j} - a_{++} = a'_{i+} + a_{+j} - \sum(A^1, A^2, A^3)$$
$$\leq a_{ij} \leq \min\{a'_{i+}, a_{+j}\},$$

where $a'_{i+} = a_{i+} - \sum_{i=1}^{s_1} a_{ij}$ can be computed by a snooper. Moreover, one can verify that the above lower bound of a_{ij} can be further improved by the following:

$$a'_{i+} + a_{+j} - \sum(A^1, A^3) \leq a_{ij}.$$

Appendix C

Proof of Lemma 4.2. Without loss of generality, consider a_{11} and assume that all a_{i1} and a_{1j} are not known to a snooper. It is clear that the Fréchet lower bound of a_{11} is a lower bound of a_{11}. To prove that it is the exact lower bound, we construct an irregular core cuboid $\{a'_{ij} \geq 0\}$ such that a'_{11} has the value of the Fréchet lower bound and that the star cuboids $\{a_{i+}\}$ and $\{a_{+j}\}$ can be derived from it.

First, consider the case where $a_{1+} + a_{+1} - a_{++} \leq 0$. From $a_{1+} + a_{+1} - a_{++} \leq 0$, we have $a_{11} \leq \sum_{i,j \neq 1} a_{ij}$. There exist $\{\delta_{ij}\}_{i,j \neq 1}$ such that $\sum_{i,j \neq 1} \delta_{ij} = a_{11}$ and $0 \leq \delta_{ij} \leq a_{ij}$. Let

$$\begin{cases} a'_{11} = 0, \\ a'_{1j} = a_{1j} + \sum_{i \neq 1} \delta_{ij} & (j \neq 1), \\ a'_{i1} = a_{i1} + \sum_{j \neq 1} \delta_{ij} & (i \neq 1), \\ a'_{ij} = a_{ij} - \delta_{ij} & (i \neq 1, j \neq 1). \end{cases}$$

From $\{a'_{ij}\}$, one can derive the star cuboids $\{a_{i+}\}$ and $\{a_{+j}\}$, because

$$\begin{cases} \sum_j a'_{1j} = \sum_{j \neq 1}(a_{1j} + \sum_{i \neq 1} \delta_{ij}) = a_{1+} \\ \text{for } i \neq 1 : \sum_j a'_{ij} = a'_{i1} + \sum_{j \neq 1} a'_{ij} = \\ a_{i1} + \sum_{j \neq 1} \delta_{ij} + \sum_{j \neq 1}(a_{ij} - \delta_{ij}) = a_{i+}. \end{cases}$$

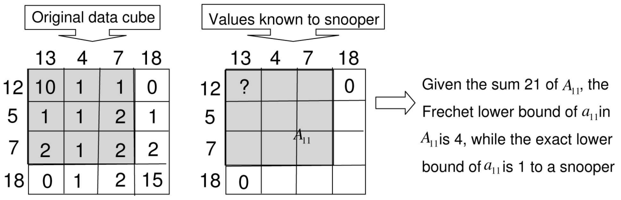

Original data cube
13 4 7 18
12 10 1 1 0
5 1 1 2 1
7 2 1 2 2
18 0 1 2 15
Values known to snooper
13 4 7 18
12 ? 0
5
7
18 0
A_{11}
Given the sum 21 of A_{11}, the Frechet lower bound of a_{11}in $A_{11}$is 4, while the exact lower bound of a_{11} is 1 to a snooper

Fig. 6. Fréchet lower bound in the companion cuboid is not the exact lower bound.

Similarly, one can verify $\sum_i a'_{ij} = a_{+j}$ for all j. The construction is complete.

Then, consider the case where $a_{1+} + a_{+1} - a_{++} > 0$. Let

$$\begin{cases} a'_{11} = a_{1+} + a_{+1} - a_{++}, & \\ a'_{1j} = a_{+j} & (j \neq 1), \\ a'_{i1} = a_{i+} & (i \neq 1), \\ a'_{ij} = 0 & (i \neq 1, j \neq 1). \end{cases}$$

It is clear that $\sum_j a'_{ij} = a_{i+}$ and $\sum_i a'_{ij} = a_{+j}$ for all i and j.□

Appendix D

Proof of Theorem 4.3. Without loss of generality, consider a_{11} and its companion cuboid A_{11} in an irregular cuboid A. Let c be the sum of all cell values in A_{11}. The Fréchet lower bound of a_{11} in the companion cuboid A_{11} is $\max\{0, a_{1+} + a_{+1} - c\}$ (note that a_{1+}, a_{+1}, and c are known to a snooper). It is clear that this bound is a lower bound of a_{11}. To prove that it is the exact lower bound, we need to construct another irregular cuboid $A' = \{a'_{ij}\}$ such that a'_{11} has the value of the Fréchet lower bound in the companion cuboid and that the star cuboids of A can be derived from it.

We first construct another companion cuboid A'_{11} such that a'_{11} has the value of the Fréchet lower bound in the companion cuboid and that the 1D sums of the companion cuboid A_{11} remain unchanged in A'_{11}. The construction of such A'_{11} follows the proof of Lemma 4.2.

Then, the irregular cuboid A' is constructed by combining A'_{11} with those cells in $A - A_{11}$. It is clear that the star cuboids of A can be derived from A'. □

Appendix E

Proof of Theorem 4.4. Without loss of generality, consider the Fréchet lower bound of a_{11} and its companion cuboid A_{11}. According to the proof of Theorem 4.3, there exists another companion cuboid A'_{11} such that a'_{11} has the value of the Fréchet lower bound and that the 1D sums of the companion cuboid A_{11} remain unchanged in A'_{11}. By combining A'_{11} with those cells in $A - A_{11}$, one obtains an irregular core cuboid from which the star cuboids in the original irregular data cube can be derived. Since the exact lower bound of a_{11} is the lowest possible value in any irregular core cuboid from which the original star cuboids can be derived, the Fréchet lower bound of a_{11} in its companion cuboid A_{11} is no less than the exact lower bound of a_{11}. □

Fig. 6 gives an example that shows that in certain cases, the Fréchet lower bound in the companion cuboid is indeed tighter than the exact lower bound.

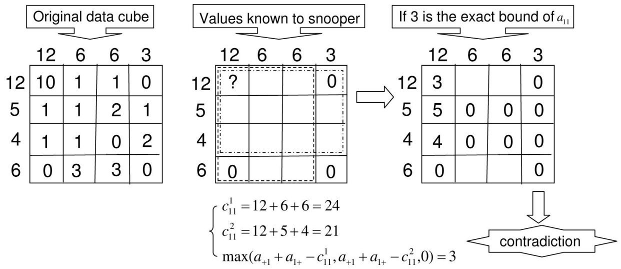


Fig. 7. $\max\{a_{+1} + a_{1+} - c^1_{11}, a_{+1} + a_{1+} - c^2_{11}, 0\}$ is not the exact lower bound.

Note that in Theorem 4.4, a snooper knows neither the grand total of the companion cuboid nor the Fréchet lower bound in the companion cuboid. The Fréchet lower bound in the companion cuboid is a lower bound from an auditor's perspective; it cannot be considered a lower bound from a snooper's perspective (as in the proof of Theorem 4.3).

Appendix F

Proof of Theorem 4.5. Without loss of generality, consider a_{11} and its companion sums c^1_{11} and c^2_{11}. For any irregular core cuboid $\{a'_{ij} \geq 0\}$ from which the star cuboids of the original cube can be derived, we have $a_{+1} + a_{1+} - c^1_{11} = a'_{11} - \sum_{t_1,t_2 \neq 1}\{a'_{t_1t_2} \mid a'_{1t_2} \notin \cup_k A^k\} \leq a'_{11}$ and $a_{+1} + a_{1+} - c^2_{11} = a'_{11} - \sum_{t_1,t_2 \neq 1}\{a'_{t_1t_2} \mid a'_{t_11} \notin \cup_k A^k\} \leq a'_{11}$; therefore, $\max\{a_{+1} + a_{1+} - c^1_{11}, a_{+1} + a_{1+} - c^2_{11}, 0\}$ is a lower bound of a_{11}. □

Fig. 7 gives an example that shows that in certain cases, $\max\{a_{+1} + a_{1+} - c^1_{11}, a_{+1} + a_{1+} - c^2_{11}, 0\}$ is indeed looser than the exact lower bound of a_{11}. In this example, $a_{41} = a_{14} = a_{44} = 0$ are known to a snooper. The snooper can compute $\max\{a_{+1} + a_{1+} - c^1_{11}, a_{+1} + a_{1+} - c^2_{11}, 0\} = 3$. If three is the exact lower bound of a_{11}, then a_{21} and a_{31} must be five and four, respectively, to satisfy $a_{+1} = 12$. Consequently, $a_{2j} = a_{3j} = 0$ for $j = 2, 3, 4$ for satisfying $a_{2+} = 5$ and $a_{3+} = 4$. A contradiction is committed since $a_{+4} = 3$ can never be satisfied. Therefore, $\max\{a_{+1} + a_{1+} - c^1_{11}, a_{+1} + a_{1+} - c^2_{11}, 0\}$ cannot be the exact lower bound in this example.

Appendix G

Proof of Theorem 5.1. First, prove that the new lower bound is indeed a lower bound for cell $a_{t_1\cdots t_n}$. From

$$a_{t_1\cdots t_n} = a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \sum_{t \neq t_i} a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n}$$

and

$$a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n} \leq \min\{a_{+t_2\cdots t_{i-1}tt_{i+1}\cdots t_n}, a_{t_1+t_3\cdots t_{i-1}tt_{i+1}\cdots t_n}, \cdots, a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_{n-1}+}\},$$

we have

$$a_{t_1\cdots t_n} \geq a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \sum_{t \neq t_i} \min\{a_{+t_2\cdots t_{i-1}tt_{i+1}\cdots t_n}, a_{t_1+t_3\cdots t_{i-1}tt_{i+1}\cdots t_n}, \cdots a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_{n-1}+}\}.$$

Thus, the new lower bound is indeed a lower bound.

Then, prove that the new lower bound is greater than or equal to the n-dimensional Fréchet lower bound. For any term in the max bracket in the formula of the n-dimensional Fréchet lower bound, one has

$$\begin{aligned}
&a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} + a_{t_1\cdots t_{j-1}+t_{j+1}\cdots t_n} - \\
&a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_{j-1}+t_{j+1}\cdots t_n} = \\
&a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \sum_{t\neq t_i} a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_{j-1}+t_{j+1}\cdots t_n} \leq \\
&a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \sum_{t\neq t_i} \min\{a_{+t_2\cdots t_{i-1}tt_{i+1}\cdots t_n}, \\
&a_{t_1+t_3\cdots t_{i-1}tt_{i+1}\cdots t_n}, \cdots a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_{n-1}+}\}.
\end{aligned}$$

Thus, for any of $\binom{n}{2}$ terms in the max bracket of the lower Fréchet bound, there exists one out of n terms in the max bracket of our new lower bound such that the latter is greater than or equal to the former. Therefore, the new lower bound is greater than or equal to the Fréchet lower bound.

Now, consider the new upper bound for $a_{t_1\cdots t_n}$. From $a_{t_1\cdots t_n} = a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \sum_{t\neq t_i} a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n}$ and $a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n} \geq \underline{a}_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n}$, where $\underline{a}_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n}$ is the new lower bound of $a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n}$, we have $a_{t_1\cdots t_n} \leq a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \sum_{t\neq t_i} \underline{a}_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n}$. Thus, the new upper bound is indeed an upper bound. Compared with the Fréchet upper bound, it is clear that the new upper bound is less than or equal to the Fréchet upper bound. □

APPENDIX H

Proof of Theorem 5.2. We prove that the transformed lower bound is the same as the new lower bound given before:

$$\max\left\{\begin{array}{l} 0, a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \\ \sum_{t\neq t_i} \min\{a_{+t_2\cdots t_{i-1}tt_{i+1}\cdots t_n}, a_{t_1+t_3\cdots t_{i-1}tt_{i+1}\cdots t_n}, \\ \cdots a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_{n-1}+}\} \mid 1 \leq i \leq n \end{array}\right\}.$$

If for all $t \neq t_i$, one has

$$\begin{aligned}
&\min\{a_{+t_2\cdots t_{i-1}tt_{i+1}\cdots t_n}, a_{t_1+t_3\cdots t_{i-1}tt_{i+1}\cdots t_n}, \\
&\cdots a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_{n-1}+}\} = \breve{a}_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n},
\end{aligned}$$

then the theorem is proven. Otherwise, there exists a $t \neq t_i$ such that the following equation holds

$$\begin{aligned}
&\min\{a_{+t_2\cdots t_{i-1}tt_{i+1}\cdots t_n}, a_{t_1+t_3\cdots t_{i-1}tt_{i+1}\cdots t_n}, \\
&\cdots a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_{n-1}+}\} > \\
&\breve{a}_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n} = a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n}.
\end{aligned}$$

Then,

$$\begin{aligned}
&a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \\
&\sum_{t\neq t_i} \min\{a_{+t_2\cdots t_{i-1}tt_{i+1}\cdots t_n}, a_{t_1+t_3\cdots t_{i-1}tt_{i+1}\cdots t_n}, \\
&\cdots a_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_{n-1}+}\} < 0 \\
&a_{t_1\cdots t_{i-1}+t_{i+1}\cdots t_n} - \sum_{t\neq t_i} \breve{a}_{t_1\cdots t_{i-1}tt_{i+1}\cdots t_n} \leq 0.
\end{aligned}$$

The theorem is proven. □

APPENDIX I

Proof of Theorem 5.3. Since the Fienberg lower bound is equivalent to the Fréchet lower bound, we only need to prove that the new upper bound $\overline{a}_{ijk}$ is less than or equal to the Fienberg upper bound.

On the one hand, one can verify that

$$\begin{aligned}
&a_{ijk} + \sum_{t_1\neq i,t_2\neq j,t_3\neq k} a_{t_1t_2t_3} = a_{+++} - \\
&a_{i++} - a_{+j+} - a_{++k} + a_{ij+} + a_{i+k} + a_{+jk}.
\end{aligned}$$

On the other hand, from the formula of $\overline{a}_{ijk}$, one can derive

$$\begin{aligned}
\overline{a}_{ijk} &\leq a_{+jk} - \sum_{t_1\neq i} \underline{a}_{t_1jk} \\
&\leq a_{+jk} - \sum_{t_1\neq i}(a_{t_1+k} - \sum_{t_2\neq j} a_{t_1t_2+}) \\
&= a_{+jk} - \sum_{t_1\neq i}(a_{t_1jk} - \sum_{t_2\neq j,t_3\neq k} a_{t_1t_2t_3}) \\
&= a_{ijk} + \sum_{t_1\neq i,t_2\neq j,t_3\neq k} a_{t_1t_2t_3}.
\end{aligned}$$

Combining this with (1) and given the obvious fact that $\overline{a}_{ijk} \leq \min\{a_{+jk}, a_{i+k}, a_{ij+}\}$, one has $\overline{a}_{ijk} \leq \min\{a_{+jk}, a_{i+k}, a_{ij+}, a_{+++} - a_{i++} - a_{+j+} - a_{++k} + a_{ij+} + a_{i+k} + a_{+jk}\}$. □

ACKNOWLEDGMENTS

The authors would like to thank the Editor-in-Chief, Professor Virgil Gligor, and the anonymous reviewers for their help in the review process. H. Lu would like to thank Dr. Shuhong Wang from Singapore Management University for his help for the proof of Lemma 4.2. This work was conducted when he was at Singapore Management University. Y. Li would like to thank Professor Ramayya Krishnan from the Heinz School of Public Policy and Management, Carnegie Mellon University, for his valuable comments. This work was partially supported by the SMU Office of Research under 04-C220-SMU-003.

REFERENCES

[1] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals," *Data Mining and Knowledge Discovery,* vol. 1, no. 1, pp. 29-53, 1997.

[2] S. Chaudhuri and U. Dayal, "An Overview of Data Warehousing and OLAP Technology," *SIGMOD Record,* vol. 26, no. 1, pp. 65-74, 1997.

[3] G. Dong, J. Han, J.M.W. Lam, J. Pei, and K. Wang, "Mining Multi-Dimensional Constrained Gradients in Data Cubes," *Proc. 27th Int'l Conf. Very Large Data Bases,* pp. 321-330, 2001.

[4] D.E. Denning, *Cryptography and Data Security.* Addison-Wesley, 1982.

[5] E. Bertino and R. Sandhu, "Database Security: Concepts, Approaches, and Challenges," *IEEE Trans. Dependable and Secure Computing,* vol. 2, no. 1, pp. 2-19, Jan.-Mar. 2005.

[6] A. Dobra and S.E. Fienberg, "Bounds for Cell Entries in Contingency Tables Induced by Fixed Marginal Totals with Applications to Disclosure Limitation," *Statistical J. United States,* vol. 18, pp. 363-371, 2001.

[7] M. Fréchet, *Les Probabilities, Associées a un Système d'Événments Compatibles et Dépendants,* vol. Premiere Partie, Hermann & Cie, 1940.

[8] Y. Li, H. Lu, and R.H. Deng, "Practical Inference Control for Data Cubes (extended abstract)," *Proc. IEEE Symp. Security and Privacy,* pp. 115-120, 2006.

[9] J.P. Ignizio and T.M. Cavalier, *Linear Programming.* Prentice Hall, 1994.

[10] A. Dobra, A. Karr, and A. Sanil, "Preserving Confidentiality of High-Dimensional Tabulated Data: Statistical and Computational Issues," *Statistics and Computing,* vol. 13, pp. 363-370, 2003.

[11] L. Cox, "On Properties of Multi-Dimensional Statistical Tables," *J. Statistical Planning and Inference,* vol. 117, no. 2, pp. 251-273, 2003.

[12] L. Cox, "Bounding Entries in 3-Dimensional Contingency Tables," *Inference Control in Statistical Databases: From Theory to Practice.* Springer, pp. 21-33, 2002.

[13] S. Fienberg, "Fréchet and Bonferroni Bounds for Multi-Way Tables of Counts with Applications to Disclosure Limitation," *Proc. Conf. Statistical Data Protection,* pp. 115-129, 1999.

[14] S. Chowdhury, G. Duncan, R. Krishnan, S. Roehrig, and S. Mukherjee, "Disclosure Detection in Multivariate Categorical Databases: Auditing Confidentiality Protection through Two New Matrix Operators," *Management Sciences,* vol. 45, pp. 1710-1723, 1999.

[15] L. Buzzigoli and A. Giusti, "An Algorithm to Calculate the Lower and Upper Bounds of the Elements of an Array Given Its Marginals," *Proc. Conf. Statistical Data Protection,* pp. 131-147, 1999.

[16] A. Dobra and S.E. Fienberg, "Bounds for Cell Entries in Contingency Tables Given Fixed Marginal Totals and Decomposable Graphs," *Proc. Nat'l Academy of Sciences,* vol. 97, no. 22, pp. 11885-11892, 2000.

[17] L. Wang, S. Jajodia, and D. Wijesekera, "Securing OLAP Data Cubes against Privacy Breaches," *Proc. IEEE Symp. Security and Privacy,* pp. 161-175, 2004.

[18] B.K. Bhargava, "Security in Data Warehousing (Invited Talk)," *Proc. Second Data Warehousing and Knowledge Discovery,* pp. 287-289, 2000.

[19] L. Brankovic, P. Norak, M. Miller, and G. Wrightson, "Usability of Compromise-Free Statistical Databases," *Proc. Ninth Int'l Conf. Scientific and Statistical Database Management,* pp. 144-154, 1997.

[20] L. Wang, D. Wijesekera, and S. Jajodia, "Cardinality-Based Inference Control in Sum-Only Data Cubes," *Proc. Seventh European Symp. Research in Computer Security,* pp. 55-71, 2002.

[21] L. Wang, Y. Li, D. Wijesekera, and S. Jajodia, "Precisely Answering Multi-Dimensional Range Queries without Privacy Breaches," *Proc. Eighth European Symp. Research in Computer Security,* pp. 100-115, 2003.

[22] N.R. Adam and J.C. Wortmann, "Security-Control Methods for Statistical Databases: A Comparative Study," *ACM Computing Surveys,* vol. 21, no. 4, pp. 515-556, 1989.

[23] L. Willenborg and T. de Walal, *Statistical Disclosure Control in Practice.* Springer, 1996.

[24] J. Domingo-Ferrer, "Advances in Inference Control in Statistical Databases: An Overview," *Inference Control in Statistical Databases: From Theory to Practice,* pp. 1-7, 2002.

[25] J.F. Traub, Y. Yemini, and H. Wozniakowski, "The Statistical Security of a Statistical Database," *ACM Trans. Database Systems,* vol. 9, no. 4, pp. 672-679, 1984.

[26] Y. Li, L. Wang, and S. Jajodia, "Preventing Interval-Based Inference by Random Data Perturbation," *Privacy Enhancing Technologies,* pp. 160-170, 2002.

[27] D. Agrawal and C.C. Aggarwal, "On the Design and Quantification of Privacy Preserving Data Mining Algorithms," *Proc. 20th ACM SIGACT-Sigmod-SIGART Symp. Principles of Database Systems,* 2001.

[28] K. Muralidhar and R. Sarathy, "A General Additive Data Perturbation Method for Database Security," *Management Sciences,* vol. 45, pp. 1399-1415, 1999.

[29] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the Privacy Preserving Properties of Random Data Perturbation Techniques," *Proc. Third IEEE Int'l Conf. Data Mining,* pp. 99-106, 2003.

[30] Z. Huang, W. Du, and B. Chen, "Deriving Private Information from Randomized Data," *Proc. ACM SIGMOD '05,* pp. 37-48, 2005.

[31] L.L. Beck, "A Security Mechanism for Statistical Databases," *ACM Trans. Database Systems,* vol. 5, no. 3, pp. 316-338, 1980.

[32] J. Schlörer, "Security of Statistical Databases: Multidimensional Transformation," *ACM Trans. Database Systems,* vol. 6, no. 1, pp. 95-112, 1981.

[33] P. Samarati and L. Sweeney, "Protecting Privacy When Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression," technical report, SRI Int'l, 1998.

[34] L. Sweeney, "Achieving k-Anonymity Privacy Protection Using Generalization and Suppression," *Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, no. 5, pp. 571-588, 2002.

[35] F.Y.L. Chin and G. Özsoyoglu, "Statistical Database Design," *ACM Trans. Database Systems,* vol. 6, no. 1, pp. 113-139, 1981.

[36] J. Schlörer, "Information Loss in Partitioned Statistical Databases," *Computer J.,* vol. 26, no. 3, pp. 218-223, 1983.

[37] J. Domingo-Ferrer and J.M. Mateo-Sanz, "Practical Data-Oriented Microaggregation for Statistical Disclosure Control," *IEEE Trans. Knowledge and Data Eng.,* vol. 14, no. 1, pp. 189-201, Jan. 2002.

[38] L.H. Cox, "Suppression Methodology and Statistical Disclosure Control," *J. Am. Statistical Assoc.,* vol. 75, no. 370, pp. 377-385, 1980.

[39] M. Fischetti and J.J. Salazar, "Solving the Cell Suppression Problem on Tabular Data with Linear Constraints," *Management Sciences,* vol. 47, pp. 1008-1026, 2000.

[40] M. Fischetti and J.J. Salazar, "Partial Cell Suppression: A New Methodology for Statistical Disclosure Control," *Statistics and Computing,* vol. 13, pp. 13-21, 2003.

Haibing Lu received the BSc and MSc degrees in mathematics from Xi'an Jiaotong University, China, in 2002 and 2005, respectively. He is currently working toward the PhD degree in information technology in the Management Science and Information Systems Department, Rutgers University. He was a research assistant in the School of Information Systems, Singapore Management University, from 2005 to 2006. His research interests include data security, data mining, access control model, and optimization.

Yingjiu Li received the PhD degree in information technology from George Mason University in 2003. He is currently an assistant professor in the School of Information Systems, Singapore Management University. His research interests include applications security, privacy protection, and data rights management. He has published 39 technical papers in the refereed journals and conference proceedings. He is a member of the ACM and the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.