DOI 10.1007/s10115-004-0193-2
Springer-Verlag London Ltd. © 2005
Knowledge and Information Systems (2006) 9: 38–61

Knowledge and Information Systems

Privacy preservation for data cubes

Sam Y. Sung[1], Yao Liu[1], Hui Xiong[2], Peter A. Ng[3]

[1]Department of Computer Science, National University of Singapore, Singapore
[2]MSIS Department, Rutgers University, USA
[3]Department of Computer Science, University of Texas—Pan American, Edinburg, USA

Abstract. A range query finds the aggregated values over all selected cells of an online analytical processing (OLAP) data cube where the selection is specified by the ranges of contiguous values for each dimension. An important issue in reality is how to preserve the confidential information in individual data cells while still providing an accurate estimation of the original aggregated values for range queries. In this paper, we propose an effective solution, called the zero-sum method, to this problem. We derive theoretical formulas to analyse the performance of our method. Empirical experiments are also carried out by using analytical processing benchmark (APB) dataset from the OLAP Council. Various parameters, such as the privacy factor and the accuracy factor, have been considered and tested in the experiments. Finally, our experimental results show that there is a trade-off between privacy preservation and range query accuracy, and the zero-sum method has fulfilled three design goals: security, accuracy, and accessibility.

Keywords: Privacy preservation; OLAP; Random data distortion; Range query

1. Introduction

Background: A *data warehouse* defined in Chaudhuri and Dayal (1997) is a collection of data from multiple sources, integrated into a common repository and extended by summary information (such as aggregate views), that is primarily used in organisational decision making. A class of queries that typically involves group-by and aggregation operators is called an online analytical processing or OLAP. OLAP software enables analysts and managers to gain insight into the performance of an enterprise and help decision making.

OLAP applications are dominated by ad hoc, complex queries. There are two approaches used in its implementation. The first approach uses the relational feature of conventional databases, which is called the relational-OLAP (ROLAP). The

Received 30 September 2003
Revised 29 January 2004
Accepted 24 October 2004
Published online 29 July 2005

other approach uses a *data cube* (Gray et al. 1996), known as MOLAP (Agrawal et al. 1997). Data-cube systems support a query style in which the data is best thought of as a multidimensional array, which is influenced by end-user tools such as spreadsheets, in addition to database query language. In the data-cube model, a data cube is constructed from a subset of attributes in the database. Certain *attributes* are chosen to be *measure attributes*, i.e. the attributes of which values are of interest. The remaining attributes are referred to as *functional attributes* or *dimensions*. The measure attributes of records with the same functional attributes values are combined (i.e. summed up) into aggregate values. Thus, a data cube could be viewed as a d-dimensional array, indexed by the values of the d dimension attributes, the cells of which contain the values of the measure attributes for the corresponding combination of dimension attributes.

Range queries apply a given aggregation operation over selected cells where the selection is specified as contiguous ranges in the domains of some of the attributes. Two types of queries, *range sum* and *range max*, have been extensively studied in recent years (Ho et al. 1997; Lee et al. 2000).

In this paper, we focus on the range-sum type of OLAP queries. An example is *sales data*. The measure attribute is *sales* and the dimensions are *item*, *day*, and *branch*. An example of a range-sum query is as follows: Find the total sales of stationery items that have an item code ranging from 1,201 to 1,300, between day 130 and 159 in the western outlets, with a branch number ranging from 45 to 89. This is a range-sum query, which can be realised using the following SQL-like statement: SELECT SUM(amount) FROM sales WHERE $(1{,}201 \leq item \leq 1{,}300)$ AND $(130 \leq day \leq 159)$ AND $(45 \leq branch \leq 89)$.

Another summary information query closely related to the range sum is the *range avg*, which is equal to range sum divided by the total count. For example, the average sales per stationery item, the average stationery sales per branch, etc., are all derived from the range sum of the above SQL query.

Scenario: There are three distinctive features that make our problem different from other applications in this field.

1. Our model is a client/server model in which the server is the company or the holder of data/information and the clients are OLAP users. If we compare our model with other models, where the clients are usually close to the server, our model is like a distributed system rather than the centralized system of the other models.
2. The data needs to be accessed quickly, as its response time is important. Therefore, data cubes are materialised and possibly precomputed on the server site and delivered to the client sites.
3. The only attributes that need to be protected are the measure attributes, whereas other functional attributes are not connected to the measure attributes and so usually do not need protection. Also, the order of functional attributes values needs to be preserved.

Objective: Data privacy refers to the issue of how to preserve the confidential information in individual data cells while still being able to provide an accurate estimation of the original summation values for range queries. There are three major goals in data privacy:

1. Security—Any sensitive data must be protected from being revealed.
2. Accuracy—Results of any analyses are valuable in a business's decision making.

3. Accessibility—A data warehouse is primarily built as an open system. In particular, exploratory OLAP analysis requires this open nature and so there should be no unnecessary restriction or control on the access to data.

However, data warehouses by their very nature create a security conflict as described in Kimball (1997). On one hand, the goal is to make data as accessible as possible. On the other hand, this valuable data is usually very sensitive. So security controls may hinder the analytical discovery process (refer to Priebe and Pernul (2000)). Another conflict is between security and accuracy of the query results. With the increase in the number of data warehouses and OLAP users, the misuse of data warehouse is steadily growing, which has led to the need for proper techniques to support all three goals of data privacy.

Justifications: Typically, people think of the issue of privacy as the need to protect individual data. Most effort has been in this direction. In a data-cube model, this privacy problem can be scaled up to aggregated or *collective* information. That is, the release of a block (subcube) of information should not enable *identification* of cell information. Such collective information is generally needed in the decision-making process, but its release should not cause any concerns. Therefore, the objective of privacy preserving for data cubes is to develop a technique that guarantees no cell data being revealed when collective information is released. Data privacy for data warehouses/data cubes is important for companies. An investment company can collect and hold the following data and information:

- Over a million customer accounts.
- In each account, the stocks a customer bought and the quantity, purchase_date and price of each purchase.
- The stocks each customer sold and the quantity, sale_date and price.

Much interesting and useful knowledge is contained in such an Investment-data warehouse, but individual investors will lose trust in the company if their data are revealed. The company may be willing to participate in a collaboration project but only if the protection of its own data can be ensured.

Contributions: The main contributions of this paper are summarised as follows:

1. We propose a simple but effective solution, called the zero-sum method, for providing an accurate estimation of the original values for range queries in a data cube while preserving the confidential information in individual data cells. Our method is based on random-data-distortion techniques and relative random-data distortion is applied.
2. We derive theoretical formulas to analyse the performance of our method. Also, we demonstrate that why a random matrix-based spectral filtering technique proposed by Kargupta et al. (2003) cannot be effectively applied to compromise our method, even if this filtering technique was designed to challenge the privacy-preserving approaches based on random data perturbation.
3. We conduct extensive experiments using a APB benchmark dataset from the OLAP Council (1998). Various parameters, such as the *privacy factor* and the *accuracy factor*, have been considered and tested in the experiments. Our experimental results show that there is a trade-off between privacy preservation and range-query accuracy and our method has fulfilled three design goals: security, accuracy and accessibility.

Overview: The remainder of this paper is organised as follows. Section 2 reviews related works. In Sect. 3, we introduce our distortion-based data-privacy-preserving

approach. Theoretical formulas and some related matters are also discussed in this section. Experiment results are described in Sect. 4. Finally, in Sect. 5, conclusions and future work are presented.

2. Related work

Compared with literature on accelerating response time in OLAP, OLAP privacy receives little attention. In contrast, there has been extensive research in the statistical databases community on the privacy problem (an excellent survey is in Adam et al. (1989)). For instance, there has been work on preserving privacy by intentionally distorting original data. The data-distortion method for security control has been extensively studied in the statistical database literature.

2.1. Security-control methods in statistical databases

Data privacy is a concern common to both OLAP and statistical database (SDB) communities. This concern has been recently extended to the data-mining area (Agrawal and Srikant 2000). The similarities and differences between OLAP and statistical databases were given in detail in Shoshani (1997). The similarities are that both OLAP and SDB deal with multidimensional datasets and both are concerned with statistical summarisations over the dimensions of the data set.

There are two types of techniques proposed by the statistical databases community:

1. *Query restriction*—The query-restriction category includes restricting the size of query results (Fellegi 1972; Denning et al. 1979) and controlling the overlap of successive queries (Dobkin et al. 1979).
2. *Data perturbation*—The data-perturbation category includes swapping values between records (Denning 1982), swapping attribute values (Conway and Strip 1976; Estivill-Castro and Brankovic 1999), replacing the original database with a sample from the same distribution (Lee et al. 2000), adding noise to the values in the database (Traub et al. 1984), adding noise to the results of a query (Beck 1980) and sampling the results of a query (Denning 1980).

Because our goal is to allow data access with minimum restriction, query-restriction methods are undesirable. Furthermore, data security may still be in danger from a set of smartly designed probes accessing the data cube.

Because of the fast-response requirement and distributed environment of our model, it is not feasible to access directly the original data cube. Therefore, many data-perturbation methods that are based on the use of the original data cube or queried results are also not feasible.

Note that the functional attributes do not usually need protection And, as the order of their attribute values needs to be preserved, swapping techniques are undesirable and unnecessary.

The fixed-data perturbation for numerical attributes (also called *value distortion*) in SDB may be useful for OLAP privacy preservation. This method was developed by Traub et al. (1984). The idea is to return a perturbed value by adding a random noise drawn from some distribution to the true value. Suppose, for example, that the true value of a given attribute (e.g. sales) of an entity k is x_k. The answer to the sum

query, under this method, will be $= \sum_{k=1}^{n} y_k$, where $y_k = x_k + z_k$, z_k is a random-perturbation variable with $E(z_k) = 0$ and $Var(z_k) = \sigma^2$, and z_k is independent for different k's.

2.2. Privacy preserving in data mining

Privacy-preserving data mining means getting valid data-mining results without learning the underlying data values. In most applications, the privacy issue is somehow related to an individual or groups of individuals sharing some common characteristics in a given context. Sometimes the patterns detected by a data-mining system may be used in a counterproductive manner that violates the privacy of an individual or group of individuals. Therefore, it is important to protect the privacy of the data and its context while mining.

In recent developments of privacy preserving in data mining (refer to Agrawal and Srikant (2000)), the idea has been dealt for scrambling a customer's personal data by randomisation and simultaneously reconstructing the original distributions of the values of the confidential attributes. Note that the goal is to reconstruct distributions, not values in individual records. Thus, both the objectives of privacy protection and statistical-based rule accuracy can be achieved at the same time. In Agrawal and Srikant (2000), it has been shown that it is possible in decision-tree classification. Their work is also extended by Evfimievski et al. (2002) to the application of association rules. The work in Kantarcioglu and Clifton (2002) addresses the problem of association rule mining where transactions are distributed across data sources.

Privacy preservation in data mining concentrates on the reconstruction of *data distribution*, whereas our concern is the reconstruction of *aggregates*. The underlying database used by data mining is not necessarily a data cube. Sparsity and response time are more critical in our problem and distribution sensitivity is not a concern. Therefore, the methods developed for data mining are not adaptable because they are different application domains with different concerns and working environments.

2.3. Access control

Access control is an important security technique and is commonly used in operational data sources to control the access to data sources (data warehouse and source databases). However, the relational model predominates in operational systems while OLAP systems make use of the nontraditional multidimensional model. In OLAP systems, protection is not defined in terms of tables, but in terms of dimensions, hierarchical paths and granularity levels (refer to Priebe and Pernul (2000)). Thus, it is not easy to map the traditional access control into OLAP systems.

In Priebe and Pernul (2000), different OLAP access-control requirements are proposed. These requirements are related to OLAP operations mentioned in the previous section. The main objective of access control is to hide information in the data cubes. The advanced requirements satisfy more security requirements, but also create more difficulties in implementation because of the increasing complexity.

With complex security requirements, information hiding will cause several problems. For example, if certain slices are hidden, how should the higher level summary data be decided? Should the hidden slices be included? If including the hidden slices to reflect the real totals, the report will be left in an inconsistent state (the displayed total is not the summary of the displayed data). If the hidden slices are not included, only the total of the displayed values being shown, then the total will have to change

from one query to another even though the queried domain remains the same. In fact, both approaches can be found in today's commercial systems.

In OLAP applications, complex access control is too slow. In addition, allowing outsiders to directly access the inner database may result in the leakage of sensitive information even with perfect access control. For example, some companies allow their customers or partners to access their data cubes for viewing aggregated data, such as monthly or yearly data. However, the highly sensitive individual data should be strictly protected.

In the OLAP literature, approximating queries on subcubes from higher level aggregations is investigated in Barbara and Sullivan (1997). The idea is to divide the cube into regions and to use a statistical model to describe each region, with an additional estimation procedure being introduced to estimate the missing entries with a certain level of accuracy based on the incomplete specified cube (the quasi-cube). The quasi-cube is designed to save storage, but it does hide some individual entries. Queries of the quasi-cube could provide approximate answers by estimating the missing entries with a reasonable level of accuracy using linear regression. However, processing the estimation procedure must be faster than computing the data from the underlying relations and a certain portion of the storage has to be used for the description.

The idea of fixed-data perturbation for numerical attributes (*value distortion*) in SDB seems more attractive to us. The advantages of a distortion-based algorithm are as follows:

1. The method is simple.
2. The distorted data looks very different from the original data and it is almost impossible to accurately estimate the original data.
3. The distorted data can be open to many users without introducing any access restrictions.
4. Unlike the quasi-cube, which needs an additional estimation procedure, all existing commercial OLAP applications can use such distorted data without any change.

However, negative results show that these techniques cannot provide High-quality statistics and prevent low disclosure of individual information at the same time (refer to Adam et al. (1989)). This is mainly because the goal to provide high-quality estimates is at a point level, which greatly conflicts with the objective of preventing information disclosure, which is also set at the same point level.

3. Distortion-based data privacy-preserving methods

In this section, we introduce our privacy-preserving method. The method we proposed is also based on random data distortion, but it is specifically aimed at solving the *data-cube summation problem*. Several terminologies will be defined before the privacy-preserving method is described.

3.1. Terminologies

Definition 1. A data cube Ω of d dimension is a d-dimensional array. For each dimension i, the size is n_i, which represents the number of distinct values for that dimension. Thus, the data cube consists of $n_1 \times n_2 \times ... \times n_d$ cells, and each cell can be represented as $\Omega[X_1, X_2, ..., X_d]$ where $0 \leq X_i < n_i$.

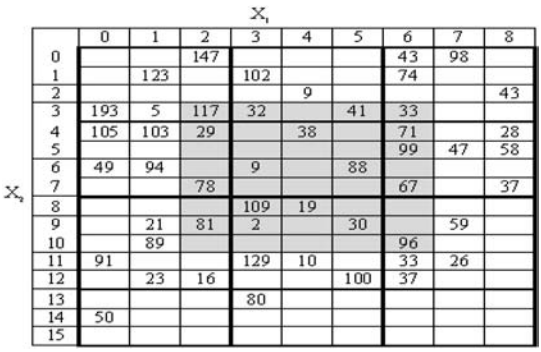


Fig. 1. Data cube, blocks and query range

Definition 2. Given a data cube Ω of d dimensions and the size of each dimension being $n_i (1 \leq i \leq d)$, with d partition factor $b_1, \ldots, b_d$, the data cube can be partitioned into $\prod_{i=1}^{d} (\lceil n_i / b_i \rceil)$ disjoint subregions known as *blocks* or *partitions*.

Definition 3. The input to a range-sum query can be expressed as $(l_1 : h_1, ..., l_d : h_d)$, where l_i and $h_i (1 \leq i \leq d)$ denote the lowest and highest bound of the range query in the ith dimension of the data cube. Usually, the data cube is not fully occupied. Some cells are empty, i.e. they contain NULL value because they actually correspond to no record. The density of a data cube is defined as $\frac{number\ of\ effective\ cells}{total\ number\ of\ cells}$.

Generally speaking, density ranges from 10 to 40% in the real dataset.

Example 1. Figure 1 shows a simple 2-dimensional data cube. The size of its dimension is 9×16. $\Omega[3, 11] = 129$, $\Omega[6, 4] = 71$. Density $= \frac{49}{9 \times 16} = 34\%$. The partition factors are 3 and 4 for the dimension X_1 and X_2, respectively. The shaded area indicates the *query range*.

There are four typical OLAP operations:

1. *Drill up*—Data is summarised by climbing up the hierarchy or by dimension reduction. For example, drill up from the current view of a weekly report to a monthly report.
2. *Drill down*—This is the reverse operation of drill up. It enables the user to navigate the data cube from a higher level summary to a lower level summary or detailed data. Through drill up/down, users can navigate among levels of data ranging from the most summarised to the most detailed.
3. *Slice and dice*—A slice is a subset of a multidimensional array that corresponds to a single value for one or more members of the dimensions not in the subset. A dice is a selection of some ranges over the dimensions. The operation of slice and dice helps the user to select one or more dimensions.
4. *Rotate*—Allows users to change the dimensional orientation of a report or page display. An example of the rotate operation is swapping the rows and columns.

3.2. Privacy-preserving method

From statistics literature, two popular scrambling methods, *discretisation* and *value distortion*, can be used:

1. *Discretization*—The method used most often for hiding individual values. In this method, the values in a block are averaged and the average value is used for every cell in the partition.
2. *Value distortion*—The basis of this method is to use a value, $x_i + z_i$, instead of the true value, x_i, where z_i is a random value drawn from some distribution. For example, using a uniform distribution between $[-\alpha, +\alpha]$, the mean of the random variable is 0. A more realistic method, however, is to choose the distorted value z_i from a domain of $[-\alpha|x_i|, +\alpha|x_i|]$. In other words, the distortions are defined as relative values with respect to the original data, x_i, rather than absolute values.

As stated in Sect. 2, a value distortion-based algorithm has certain attractive features. However, a distortion-based method cannot be effectively applied to OLAP directly. Because the OLAP cube is usually sparse, to simply apply the value distortion, fixed-data perturbation will lead the distorted cube's density to increase dramatically, to almost 100%. With the multidimensional characteristic of data cubes, changing one cell will affect several summation values. The distortion method applied to OLAP must guarantee a certain degree of accuracy of range-sum queries. This is not considered in SDB. If the query size is very large, the perturbed database tends to give a large error. Thus, our solution to this problem is to adjust the initially distorted data so that the accuracy of range-sum queries is close to 100%, especially when dealing with large queries.

3.2.1. The zero-sum method

First, we start with an initial distortion in each cell. The initially distorted data is then adjusted so that all the *marginal sums* of each block are zeroes. These adjusted distortions are the final distortions. This adjustment process is called the *zero-sum* method. For illustration purpose, consider a case of a 2-dimensional data cube. The process of *zero sum* is to make adjustments on each row such that its (new) distortions are totalled to equal zero. Subsequently, each column's distortion values are also summed to zero.

In reality, OLAP systems often use existing languages, such as SQL, to express the OLAP operations. From this angle, the slice and dice and the drill up/down operations can be viewed as range queries.

Figure 2 illustrates how the zero-sum method preserves the confidential information in each individual data cell while still being able to provide an accurate estimation of the original summation values for range queries. The data set is a 2-dimensional data cube with 9 blocks. The gray rectangle in Fig. 2 is a range query. This range query covers 9 blocks: $A_1, A_2, \ldots, A_9$ and it fully covers A_5 only. According to the zero-sum method, the cells in each block have been adjusted, so the marginal sums of a distorted block is equal to that of the corresponding original block. Let S be the sum of the range query over the original data and S' be the sum of the distorted data. Let s_i be the sum of the gray area in the block A_i before distortion and s'_i be the sum of the gray area in the block A_i after distortion.

$$S = s_1 + s_2 + \ldots + s_9,\ S' = s'_1 + s'_2 + \ldots + s'_9\,.$$
$$S' - S = (s'_1 - s_1) + (s'_3 - s_3) + (s'_7 - s_7) + (s'_9 - s_9)\,.$$

It can be seen that $S' - S$ is a small fraction of S. That's why the zero-sum method can help to provide an accurate estimation for range-sum queries.

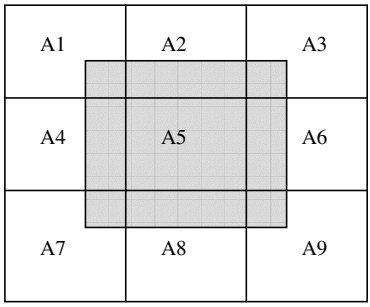


Fig. 2. Data cube, blocks and query range

There are several ways to enforce *marginal sums* to be zeroes. One of the ways is given as follows: for each row, redistribute the negative sum of the row back to each cell in the row such that the new sum becomes zero. This redistribution can be done uniformly so that each cell receives the same amount of adjustment. After the row adjustments are done, we repeat the same process for column adjustments. This process converts the original distortions to zero-sum form, and a proof is given in Theorem 1.

Theorem 1. A block (partition) of k-dimension can be converted to zero-sum form with k iterations.

Proof. Let $d_{i_1,i_2,\ldots,i_k}$ = initial distortion value for cell $\langle i_1, i_2, \ldots, i_k\rangle$, $1 \le i_j \le n_j$, $1 \le j \le k$. Let $S_{i_1,i_2,\ldots,i_{k-1},*}$ be the (marginal) sum of distortions at marginal point $(i_1, i_2, \ldots, i_{k-1}, *)$. Then, we have $S_{i_1,i_2,\ldots,i_{k-1},*} = \sum_{i_k=1}^{n_k} d_{i_1,i_2,\ldots,i_{k-1},i_k}$.

Similarly,

$$S_{i_1,i_2,\ldots,i_{k-2},*,*} = \sum_{i_{k-1}=1}^{n_{k-1}} \sum_{i_k=1}^{n_k} d_{i_1,i_2,\ldots,i_{k-2},i_{k-1},i_k} = \sum_{i_{k-1}=1}^{n_{k-1}} d_{i_1,i_2,\ldots,i_{k-1},*}$$

is defined as the (marginal) sum of distortions at marginal point $(i_1, i_2, , i_{k-2}, *, *)$. In the same way, we can define the (marginal) sum of distortions for all other marginal points.

Let $\Delta_{i_1,i_2,\ldots,i_{k-1},*} = -\frac{S_{i_1,i_2,\ldots,i_{k-1},*}}{n_k}$.

In the first iteration, we have $d_{i_1,i_2,\ldots,i_k} \leftarrow d_{i_1,i_2,\ldots,i_k} + \Delta_{i_1,i_2,\ldots,i_{k-1},*}$.

Now, the (new) value $S_{i_1,i_2,\ldots,i_{k-1},*}$ becomes zero. Similarly, $S_{i_1,i_2,\ldots,i_{k-2},*,*}$ also becomes zero, and so on, for the sum of distortions at all other marginal points.

Repeat the process, and after the second iteration, the value $S_{i_1,i_2,\ldots,i_{k-2},*,i_k}$ becomes zero.

However, the value $S_{i_1,i_2,\ldots,i_{k-1},*}$ still remains zero. This is because

$$\begin{aligned}(new)S_{i_1,i_2,\ldots,i_{k-1},*} &= (previous)S_{i_1,i_2,\ldots,i_{k-1},*} + (previous)\frac{S_{i_1,i_2,\ldots,i_{k-2},*,*}}{n_{k-1}} \\ &= 0\,.\end{aligned}$$

Similarly, after the third iteration, the values of $S_{i_1,i_2,\ldots,i_{k-1},*}$, $S_{i_1,i_2,\ldots,i_{k-2},*,i_k}$ and $S_{i_1,i_2,\ldots,i_{k-3},*,i_{k-1},i_k}$ are all zeroes. This can be seen from the following:

6	-4	4	6	-1	11
-5	-6	7	-7	-4	-15
-7	-1	-3	5	9	3
8	5	-8	-4	-3	-2
-5	-2	4	3	2	2
-3	3	-7	3	-2	-6
6	-4	6	-5	-3	0
0	-9	3	1	-2	-7

(a) Original distortions

3.8	-6.2	1.8	3.8	-3.2	0
-2	-3	10	-4	-1	0
-7.6	-1.6	-3.6	4.4	8.4	0
8.4	5.4	-7.6	-3.6	-2.6	0
-5.4	-2.4	3.6	2.6	1.6	0
-1.8	4.2	-5.8	4.2	-0.8	0
6	-4	6	-5	-3	0
1.4	-7.6	4.4	2.4	-0.6	0

(b) Distortions adjusted by row

3.6	-5.1	1.2	3.45	-3.1	0
-2.2	-1.9	9.4	-4.3	-0.9	0
-7.8	-0.5	-4.2	4.1	8.48	0
8.2	6.48	-8.2	-3.9	-2.5	0
-5.6	-1.3	3	2.3	1.7	0
-2	5.3	-6.4	3.9	-0.7	0
5.8	-2.9	5.4	-5.3	-2.9	0
0	0	0	0	0	0

(c) Distortions adjusted by column

Fig. 3. An example of using the iterative zero-sum method

For the value $S_{i_1,i_2,\ldots,i_{k-1},*}$, we have

$$(new)S_{i_1,i_2,\ldots,i_{k-1},*} = (previous)S_{i_1,i_2,\ldots,i_{k-1},*} + (previous)\frac{S_{i_1,i_2,\ldots,i_{k-3},*,i_{k-1},*}}{n_{k-2}}$$
$$= 0\,.$$

For the value $S_{i_1,i_2,\ldots,i_{k-2},*,i_k}$, we have

$$(new)S_{i_1,i_2,\ldots,i_{k-2},*,i_k} = (previous)S_{i_1,i_2,\ldots,i_{k-2},*,i_k} + (previous)\frac{S_{i_1,i_2,\ldots,i_{k-3},*,*,i_k}}{n_{k-2}}$$
$$= 0\,.$$

And the same is true for all subsequent iterations. □

Theorem 1 means that, for a k-dimensional block in the ith iteration, all marginal sums of dimension $i(1 \le i \le k)$ can be adjusted to zeroes while all marginal sums of dimension $j(1 \le j < i)$ remain zeroes. Thus, after k iteration, all marginal sums of the k-dimensional block are zeroes.

Example 2. An example of zero-sum iterations on a 2-dimensional data cube is given in Fig. 3. Suppose that there is an original block with 7 rows and 5 columns. Figure 3(a) shows the original distortions of the original block. Each cell contains a randomly generated number within $[-9, 9]$. The first iteration is to adjust the sum of each row to be zero. Each row has 5 cells and the sum of the first row is 11. The sum of the first row becomes zero by adding $-(11/5) = -2.2$ to each cell of the first row. Then we adjust all other rows in the same manner. After the first iteration, as shown in Fig. 3(b), the sum of each row is zero.

Next consider the sum of each column. Similarly, as shown in Fig. 3(b), the sum of the first column is 1.4. The sum of the first column is adjusted to be zero by adding $-(1.4/7) = -0.2$ to each of the seven cells in this column.

The cell values after the second iteration are shown in Fig. 3(c). It can be seen that the sum of each column is zero, and the sum of each row also remains as zero.

6	-4	4	6	-1	11
-5	-6	7	-7	-4	-15
-7	-1	-3	5	9	3
8	5	-8	-4	-3	-2
-5	-2	4	3	2	2
-3	3	-7	3	-2	-6
6	-4	6	-5	-3	0
0	-9	3	1	-2	-7

(a) Original distortions

3.6	-5.1	1.2	3.45	-3.1	0
-2.2	-1.9	9.4	-4.3	-0.9	0
-7.8	-0.5	-4.2	4.1	8.48	0
8.2	6.48	-8.2	-3.9	-2.5	0
-5.6	-1.3	3	2.3	1.7	0
-2	5.3	-6.4	3.9	-0.7	0
5.8	-2.9	5.4	-5.3	-2.9	0
0	0	0	0	0	0

(b) Adjusted distortions modification

4	-5	1	3	-3	0
-2	-2	9	-4	-1	0
-8	-1	-4	4	9	0
8	6	-8	-4	-2	0
-6	-1	3	2	2	0
-2	5	-6	4	-1	0
6	-2	5	-5	-4	0
0	0	0	0	0	0

(c) Reinstall to integer format

Fig. 4. Example of using formula (1) in the zero-sum method

The adjusted distortions in Fig. 3(c) is in zero-sum form. If we add the adjusted distortions to the corresponding cells in the original block, the value of all the cells in the original block will be changed but all marginal sums will remain the same. Therefore, after distortion, a certain degree of the accuracy of the range-sum query can be guaranteed to some extent.

Alternatively, using formula (1) below, a block can be converted to zero-sum form by one iteration.

$$\begin{aligned} d_{i_1,i_2,\ldots,i_k} \leftarrow d_{i_1,i_2,\ldots,i_k} &- \frac{S_{i_1,\ldots,i_{k-1},*}}{n_k} - \frac{S_{i_1,\ldots,i_{k-2},*,i_k}}{n_{k-1}} - \ldots + \frac{S_{i_1,\ldots,i_{k-2},*,*}}{n_k n_{k-1}} \\ &+ \frac{S_{i_1,\ldots,i_{k-3},*,i_{k-1},*}}{n_k n_{k-2}} + \frac{S_{i_1,\ldots,i_{k-3},*,*,i_k}}{n_{k-1} n_{k-1}} + \ldots - \ldots \\ &+ (-1)^k \frac{S_{*,*,\ldots,*}}{n_k n_{k-1} \ldots n_2 n_1} . \end{aligned} \tag{1}$$

Example 3. The same example in Fig. 3 can be converted to zero-sum form by the 1-iteration method as shown in Fig. 4. In Fig. 4(a), the original distortion value, $d_{11} = 6$, is converted to $d_{11} - \frac{S_{1,*}}{n_2} - \frac{S_{*,1}}{n_1} + \frac{S_{*,*}}{n_1 n_2} = 6 - \frac{11}{5} - \frac{0}{7} + \frac{-7}{35} = 3.6$ The result in Fig. 4(b) is the same as Fig. 3(c). If the integer format is desired, then distortions can be reinstalled by rounding the numbers. The boundary values can be used to make the marginal sums remain as zeroes. This is illustrated by Fig. 4(c).

The *time complexity* for k iterations is $k \times n_1 \times n_2 \times \ldots \times n_k$. The time complexity for 1-iteration is $2^k \times n_1 \times n_2 \times \ldots \times n_k$.

A special case is that all cells are initially distorted to a constant value. For example, when *discretisation* is used as the initial distortion, the values in a block are averaged and the average value is used for every cell in the block. In this special case, the (adjusted) final distorted values will be in a special format. For instance,

in the 2-dimensional case, the new value y_{ij} at cell *(i,j)* becomes

$$y_{ij} = x_{ij} + z_{ij} = \frac{\sum_{j=1}^{n} x_{ij}}{n} + \frac{\sum_{i=1}^{m} x_{ij}}{m} - \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}}{mn}.$$

In general, in the case of a k-dimensional data cube, we have

$$\begin{aligned} y_{i_1,i_2,\ldots,i_k} = {} & \frac{\sum_{i_k=1}^{n_k} x_{i_1,\ldots,i_{k-1},i_k}}{n_k} + \frac{\sum_{i_{k-1}=1}^{n_{k-1}} x_{i_1,\ldots,i_{k-1},i_k}}{n_{k-1}} + \ldots \\ & - \frac{\sum_{i_{k-1}=1}^{n_{k-1}}\sum_{i_k=1}^{n_k} x_{i_1,\ldots,i_{k-1},i_k}}{n_{k-1}n_k} - \frac{\sum_{i_{k-2}=1}^{n_{k-2}}\sum_{i_k=1}^{n_k} x_{i_1,\ldots,i_{k-1},i_k}}{n_{k-2}n_k} \\ & - \frac{\sum_{i_{k-2}=1}^{n_{k-2}}\sum_{i_{k-1}=1}^{n_{k-1}} x_{i_1,\ldots,i_{k-1},i_k}}{n_{k-2}n_k} - \ldots + \ldots - \ldots \\ & - (-1)^k \frac{\sum_{i_1=1}^{n_1}\ldots\sum_{i_k=1}^{n_k} xi_1, i_2, \ldots, i_n}{n_k n_{k-1}\ldots n_2 n_1}. \end{aligned} \tag{2}$$

3.2.2. Bound of distortions

In this section, we present theoretical analysis of the zero-sum method. We derive a bound on distortions that is useful when explained in probabilistic terms. A generalisation of the commonly known *Chernoff's* bounds (Motwani and Raghavan 1995) is used to derive the bound.

Theorem 2. Let X_i, $1 < i < n$, be mutually independent random variables with all $E[X_i] = 0$ and all $|X_i| < \alpha_i$. Let $S_n = X_1 + \ldots + X_i$. For $a > 0$, then

$$Pr[S_n > a] < e^{-a^2/2\sum\alpha_i^2}.$$

Proof. Let $\lambda = a/\sum\alpha_i^2$ and $h_i(x) = \frac{e^{\lambda\alpha_i}+e^{-\lambda\alpha_i}}{2} + \frac{e^{\lambda\alpha_i}-e^{-\lambda\alpha_i}}{2\alpha_i}x$.

For $x \in [-\alpha_i, \alpha_i]$, $e^{\lambda x} \le h_i(x)$. ($y = h_i(x)$ is a chord passing through the points $x = -\alpha_i$ and $x = +\alpha_i$ of a convex curve $y = e^{\lambda x}$.) Thus,

$$E[e^{\lambda X_i}] \le E[h_i(X_i)] = h_i(E[X_i]) = h_i(0) = \frac{e^{\lambda\alpha_i} + e^{-\lambda\alpha_i}}{2} = cosh(\lambda\alpha_i).$$

The inequality below is valid for all $\lambda > 0$. (The inequality can be shown by comparing the Taylor series of the two functions $e^{\lambda\alpha_i}$ and $e^{-\lambda\alpha_i}$ term wise.)

$$cosh(\lambda\alpha_i) < e^{\lambda^2\alpha_i^2/2}.$$

We have $e^{\lambda S_n} = \prod_{i=1}^{n} e^{\lambda X_i}$.
Because the X_i are mutually independent, so are $e^{\lambda X_i}$, and

$$E[e^{\lambda S_n}] = \prod_{i=1}^{n} E[e^{\lambda X_i}] = \prod_{i=1}^{n} cosh(\lambda\alpha_i) \le e^{\lambda^2\sum\alpha_i^2/2}.$$

Note that $S_n > a$ if and only if $e^{\lambda S} > e^{\lambda a}$. Apply the following Markov's inequality:

$$Pr[Y > aE[Y]] < \frac{1}{\alpha}$$

and we get

$$Pr[S_n > a] = Pr[e^{\lambda S_n} > e^{\lambda a}] < E[e^{\lambda S}]/e^{\lambda a} \leq e^{\lambda^2 \sum \alpha_i^2/2 - \lambda a}.$$

Because $\lambda = a/\sum \alpha_i^2$, the inequality becomes

$$Pr[S_n > a] < e^{-a^2/2\sum \alpha_i^2}.$$

Hence, the claim holds. □

From Theorem 2, we can derive the following two corollaries.

Corollary 1. Under the same conditions as Theorem 2, we have $Pr[|S_n| > a] < 2e^{-a^2/2\sum \alpha_i^2}$.

Corollary 2. Under the same conditions as Theorem 2, if all α_i are equal, we have $Pr[|S_n| > a] < 2e^{-a^2/2n\alpha^2}$.

When we describe the accuracy measurement in Sect. 4.1.2, Theorem 2 will be further discussed.

3.3. Security and other issues

Security: Because the (original) cubes are not allowed for probing, it's not possible for snoopers to improve their estimation of the value of a field in a record by repeatedly placing queries (Adam et al. 1989).

It is shown in Faloutsos et al. (1997) that it is possible to fully recover original distribution from nonoverlapping, contiguous partial sums. However, it requires the original distribution to be smooth enough. (That is, there will only be a small difference between successive elements of the vector.) This smooth assumption is not true in general for data cubes.

Research work on estimating attribute distributions from partial information can be found in Barbara et al. (1997). Work on approximating queries on subcubes from higher level aggregations can be found in Barbara and Sullivan (1997). However, these works did not deal with information that has been deliberately distorted. Furthermore, their estimations require some additional statistical information of actual data.

Another security attack can come from the known marginal information. For example, in the case of 2-dimensional data cubes, the correct values of the sums of rows and the sums of columns are known. However, this known data is relatively much smaller than the unknown information. For example, in the case of 2-dimensional data cubes, it is impossible to solve a system of $(m + n - 1)$ equations using $m \times n$ variables, where m is the number of rows and n is the number of columns.

Recently, Kargupta et al. (2003) proposed a random matrix-based spectral filtering technique to challenge the privacy-preserving approaches based on random data perturbation. Although random data distortion is used in our paper, this filtering technique cannot be effectively applied to compromise our approach for the

following three reasons. First, the effectiveness of this filtering technique is based on the assumption that the eigenvectors of the covariance matrix of the perturbed data are orthogonal to the eigenvectors of the covariance matrix of the original data. Consider that, in our approach, different blocks in a data cube can be distorted by random data from different distributions with different ranges. It is not difficult to generate the perturbed data such that the eigenvectors of the covariance matrix of the perturbed data are not orthogonal to the eigenvectors of the covariance matrix of the original data. In other words, one of the major assumptions in this filtering approach is invalid in our paper. Second, if there is no prior knowledge of perturbed-data distributions, which is true in our approach, this filtering approach needs to estimate the variance of the perturbed data. Even if this variance can be correctly estimated, the process for estimating the variance adds complexity and increases computation cost significantly. Finally, the filtering approach requires computing the eigenvalues of a covariance matrix. This computation cost for a large amount of data, such as a data cube, can be intractable.

Precomputing: The paper presented by Ho et al. (1997) introduces the idea of precomputing multidimensional prefix sums of a data cube for speeding up the range-sum query. Prefix sum means, for any cell at a position $\langle d_1, d_2, \ldots, d_n \rangle$ in a data cube, summing up all the values in the range starting from the position $\langle 0, \ldots, 0 \rangle$ to $\langle d_1, d_2, \ldots, d_n \rangle$. The resulting value of the summation is then stored in the cell of $\langle d_1, d_2, \ldots, d_n \rangle$ of the auxiliary data structure. Because this summation looks like a prefix of the summation of the whole data cube, it is given the name *prefix sums*. By precomputing as many prefix sums as the number of elements in the original data cube, any range-sum query can be answered by accessing and combining 2^d appropriate prefix sums, where d is the number of dimensions for which ranges have been specified in the query. Precomputing has no effect on the data security but only affects the tradeoff between response speed and storage space.

Size of block: Each dimension should have at least two values. The boundary of a block, if matched with a dimensional hierarchy, will guarantee that the drill-up operation is completely correct.

Sparsity: Sparse cubes are cubes in which many cells are empty. An empty cell means that there is no valid value in the cell. These cells can be treated as either zeroes or as missing values that are not counted. If they are treated as zeroes, after scrambling, a zero cell may become nonzero. This increases the density of a block. Also, counting empty cells as zeroes will affect the calculation on average. However, if empty cells are treated as uncounted, formula (1) cannot be guaranteed to obey zero-sum forms. This is because the number of counted cells in each row or column can be (very) different. So Theorem 1 is no longer valid. However, the experimental results show that such a nonstrict zero-sum form does not affect the accuracy much.

A special case is that a row (or column) contains no more than one valid cell. Our zero-sum method simply leaves this row unadjusted.

Absolute data distortion vs. relative data distortion: Absolute data distortion is distorting original data by an absolute value, whereas relative data distortion is distorting data by a relative value, say 20%. The absolute distortion suffers in terms of scale. For example, perturbing a salary of $15,000 by 3,000 would preserve the confidentiality of the data while at the same time perturbing a salary of $150,000 would be considered a compromise. Perturbing the data in a relative-distortion range is an alternative way to overcome the scale problem.

4. Experimental evaluation

In this section, we present an experimental evaluation of the zero-sum method. After giving a brief description of our experimental setup and performance-evaluation measures, we show the performance of the zero-sum method with respect to parameters, including initial distortion, sparsity and block size. Also, we illustrate the interactive effect between privacy and range-query accuracy. This section is then concluded with a summary of experimental results and discussions.

Experimental data sets. The experimental data sets are generated using the APB Benchmark program from OLAP Council (1998). There are four dimensions: customer, product, channel and time. The size of each dimension is 900 (customer), 9,000 (product), 9 (channel) and 17 (time). The measure attribute is dollar, with range $[0, 699]$. Cells with -1 dollar value are empty cells and are treated as missing or uncounted. The overall density of the data cube is 20%. Hence, the file size of the data cube is $900 \times 9{,}000 \times 9 \times 17 \times 0.2 = 247{,}860{,}000$ bytes $= 0.24$ GB. The original size of data file generated by APB Benchmark is approximately 15 G (ASCII text). Each record in this file is read to fill into the corresponding cells in the data cube.

Distortion range. Relative distortion range is applied to generate initial distortions. For example, given a relative distortion range $[0, 50\%]$ and a cell value α, the initial distortion of this cell is randomly generated within range $[-50\%\alpha, 50\%\alpha]$.

Experimental platform. Our experiments were performed on a PC with a Pentium III 733 MHz, 40 G hard disk and 256 MBytes of memory.

4.1. Evaluation methods

To evaluate the performance of the zero-sum methods, two measures, *privacy* and *accuracy*, are used.

4.1.1. A measure of privacy

The measure of privacy indicates how closely the original value can be estimated. Original distortions are usually generated by using a density function, $f(z)$, in an interval of $[0, \alpha]$ and then assigning positive or negative values with equal probability to it. Therefore, the expectation of distortion variable, z, is $E(z) = 0$. The expectation of variable $|z|$ is

$$E(|z|) = \int_{-\alpha}^{\alpha} |z| \frac{f(z)}{2} dz = \int_{0}^{\alpha} z f(z) dz .$$

The work in Agrawal and Srikant (2000) used a simple measure in terms of the amount of privacy, which is defined as follows: assume the original distortion is uniformly distributed in an interval of $[0, \alpha]$, then the expectation is intuitively considered as the amount of privacy.

We generalise the privacy measure in Agrawal and Srikant (2000) to all types of distributions based on the mean value (expectation): if the (original) distortion

distribution has density function $f(z)$ in an interval $[0, \alpha]$, then the amount of privacy is defined as $2 \times \int_0^\alpha z f(z) dz = 2E(|z|)$.

A more sophisticated privacy measure based on differential entropy is given by Agrawal et al. (2001): if the (original) distortion distribution has density function $f(z)$ in an interval $[0, \alpha]$, then the amount of privacy is $2^{-\int_0^\alpha f(z) log_2 f(z) dz}$.

However, in this paper, we are more interested in what the expected amount of privacy of the adjusted distortions is. Because we do not know the density function of the adjusted distortions, we can only estimate the amount of privacy at cell i by using the difference between x_i (the true original value) and y_i (the adjusted distorted value). Based on the spirit of expectation and large-number theory, we will give the definition of a privacy factor below and use it to define the privacy amount of the adjusted distortions.

Privacy factor, F_p,

$$F_p = \frac{1}{N} \sum_{i=1}^{N} |y_i - x_i| \,. \tag{3}$$

In Eq. (3), the value F_p is the *average* amount of (adjusted) distortions over a block. The value $2 \times F_p$ is considered as the amount of privacy of the adjusted distortions. It is interesting to compare the value $2 \times F_p$ with the amount of privacy of the original distortions. The privacy amount calculated in Eq. (3) does not take into account the value of the original data. To quantify privacy accurately, we need a method that takes such information into account. A modified version of formula (3) is given as

$$F_c = \frac{1}{N} \sum_{i=1}^{N} \frac{|y_i - x_i|}{|x_i|} \,. \tag{4}$$

The value F_c is the average amount of distortions (relative to x-value) over a block. The value $2 \times F_c$ is considered as the amount of *conditional privacy* of the adjusted distortions. Accordingly, the *conditional privacy* of the original distortions is also modified by a factor x_i at cell i. That is, a cell with original value x will generate distortions that are (uniformly) distributed over an interval of $[0, \alpha x]$.

Generally speaking, the measurement of privacy is dependent on the expectation, $E(|X|)$, of the distortion distribution.

4.1.2. A measure of accuracy

The difference between the sum of the distorted values and the original values over a query Q is referred to as the *accuracy* loss of Q. Let *true_sum* be the sum of all original values of cells in query Q and *answer* be the sum of all distorted values. We have the *relative accuracy loss* of Q equal to $|\frac{answer \ - \ true\_sum}{true\_sum}|$. Then a measure of accuracy is defined as follows:

Accuracy factor, $F_{a,Q}$,

$$F_{a,Q} = 2^{-|\frac{answer \ - \ true\_sum}{true\_sum}|} \,. \tag{5}$$

Note that the accuracy factor lies between 0 and 1 (inclusive). Also, because $|answer - true\_sum| = |z_1 + \ldots + z_n|$ and $\frac{|answer\ -\ true\_sum|}{true\_sum} = |\frac{z_1+\ldots+z_n}{x_1+\ldots+x_n}|$, if all $|z_i| < \alpha$ and $E[z_i] = 0$, we can derive the following expression by applying Corollary 1 of Theorem 2:

$$Pr\left[\left|\frac{answer - true\_sum}{true\_sum}\right| > a\right] = Pr[|z_1 + \ldots + z_n| > a|x_1 + \ldots + x_n|] < 2e^{-a^2|x_1+\ldots+x_n|^2/2n\alpha^2}.$$

Formula (6) shows the relationship between the relative accuracy *loss* (denoted as a) and privacy (denoted as α). Note that it is also related to the size of the query (denoted as n) and the sum of the query (expressed in terms of $(|x_1 + \ldots + x_n|)$). If the *relative* distortion is taken into consideration, in which all $|z_i| < \alpha|x_i|$ and $E[z_i] = 0$, then formula (6) becomes the following:

$$Pr\left[\left|\frac{answer - true\_sum}{true\_sum}\right| > a\right] < 2e^{-a^2(|x_1+\ldots+x_n|)^2/2\alpha^2(|x_1|^2+\ldots+|x_n|^2)}. \quad (6)$$

4.2. The experimental results

4.2.1. The effect of query sizes

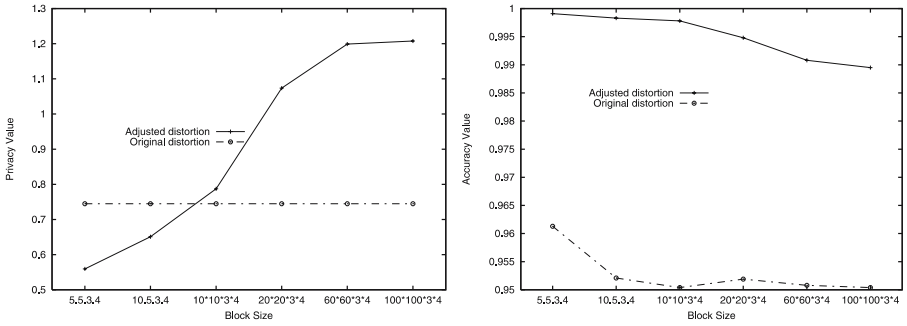
Here, we tested the effect of query sizes using a fixed block size as $5 \times 5 \times 3 \times 2$. More specifically, we examined the performance of adjusted distortion as well as original distortion. We applied uniform distributions with the range [50%, 100%] for the initial distortion and generated 200 range-sum queries with the change of size from (50, 100) to (1,000, 2,000). The experimental results are shown in Table 1. In the table, the obtained average values for the privacy factor and the accuracy factor of 200 queries show that the zero-sum method yields better accuracy on data sets with adjusted distortion, as well as satisfactory privacy.

Table 1. Privacy and accuracy factors

	Original distortion	Adjusted distortion
F_c	0.74522	0.434961
F_a	0.85515	0.984401

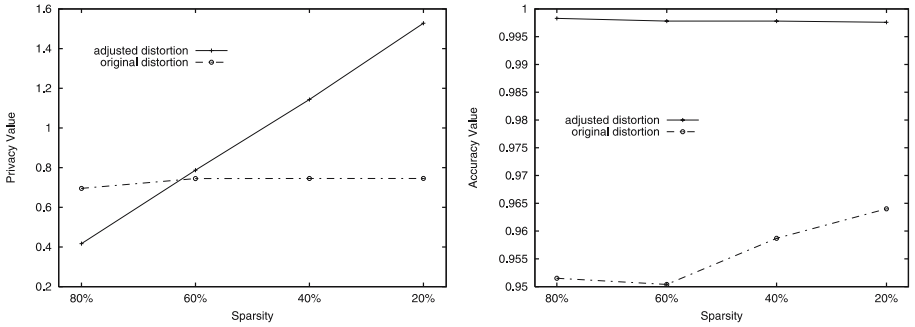
4.2.2. The effect of block sizes

To test the effect of block sizes, we arranged the sparsity of a data cube to be 60% and the distortion range to be [50%, 100%]. In this experiment, 200 range-sum queries with query size in [200, 1,000] were generated. Figure 5(a) shows the obtained privacy values for adjusted distortion and original distortion. Although the privacy values of adjusted distortion are smaller than that of original distortion when the block size is small, the overall privacy of adjusted distortion is better than that of original distortion. Figure 5(b) shows that the accuracy values of adjusted distortion are significantly and systematically better than the accuracy values of original distortion regardless of the block size.



(a) Privacy of different block size (b) Accuracy of different block size

Fig. 5. The effect of block size



(a) Privacy of different sparsity (b) Accuracy of different sparsity

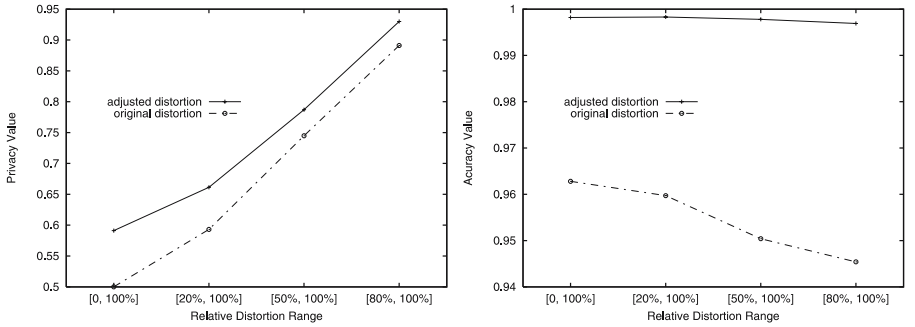
Fig. 6. The effect of sparsity

4.2.3. The effect of cube sparsity

To test the effect of cube sparsity, we arranged the block size to be $10 \times 10 \times 3 \times 4$ and the distortion range to be [50%, 100%]. In this experiment, we applied 200 range-sum queries with a query size of [200, 1,000]. Figure 6(a) shows the obtained privacy values of adjusted distortion and original distortion. As shown in the figure, when sparsity is as high as 80%, the privacy of original distortion is better than that of adjusted distortion. However, when the sparsity is less than 60%, the privacy of adjusted distortion is much better than that of original distortion. In addition, we observed that the privacy of adjusted distortion is increased with the decrease of sparsity.

Figure 6(b) shows the accuracy of adjusted distortion and original distortion with the change of the cube sparsity. As shown, the accuracy of adjusted distortion is significantly better than that of original distortion in all ranges of the cube sparsity. In addition, the accuracy of original distortion is slightly increased with the decrease of sparsity.

In summary, the more sparse the cube is, the smaller the privacy will be. But the cube sparsity does not affect greatly the accuracy of adjusted distortion.



(a) Privacy (b) Accuracy

Fig. 7. The effect of varying the lower bound of distortion range

4.2.4. The effect of distortion range

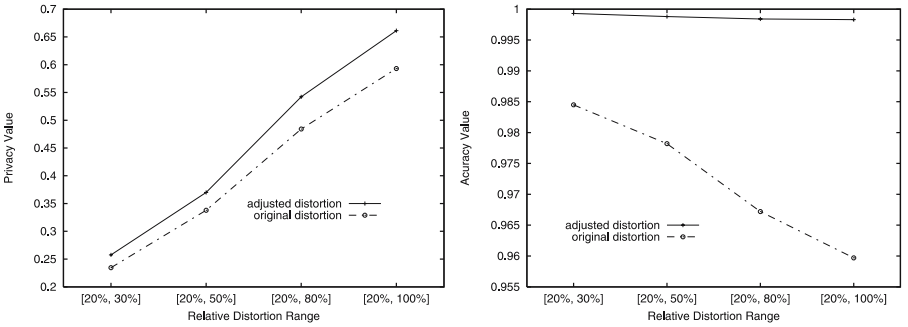
In the experiment of examining the effect of distortion range, we arranged the block size to be $10 \times 10 \times 3 \times 4$ and the cube sparsity to be 60%. We applied 200 range-sum queries with query sizes between 200 and 1,000. We then examined the average privacy and accuracy of these 200 queries with the change of distortion ranges. Figure 7(a) shows the privacy performance of the zero-sum method on both adjusted and original distortions when changing the lower bound of distortion range. In the figure, it is not surprising to see a trend that the privacy values of both adjusted distortions and original distortions are increased with the increase of the lower bound of relative distortion ranges, because higher relative distortion can bring better privacy preservation. Also, we observed that the obtained privacy of adjusted distortions is systematically better than that of original distortion.

Figure 7(b) shows the accuracy performance of the zero-sum method on both adjusted and original distortions when changing the lower bound of distortion range. The accuracy of adjusted distortions is significantly and systematically higher than that of original distortions. Furthermore, the accuracy of original distortion is decreased with the increase of the lower bound of distortion range. In contrast, changing distortion ranges does not affect significantly the accuracy of adjusted distortion.

The same effect on privacy and accuracy is observed when changing only the upper bound of distortion range, as shown in Figs. 8(a) and 8(b), respectively. In summary, no matter how we change the distortion range, the privacy and accuracy achieved by adjusted distortion are better than the privacy and accuracy obtained by original distortion.

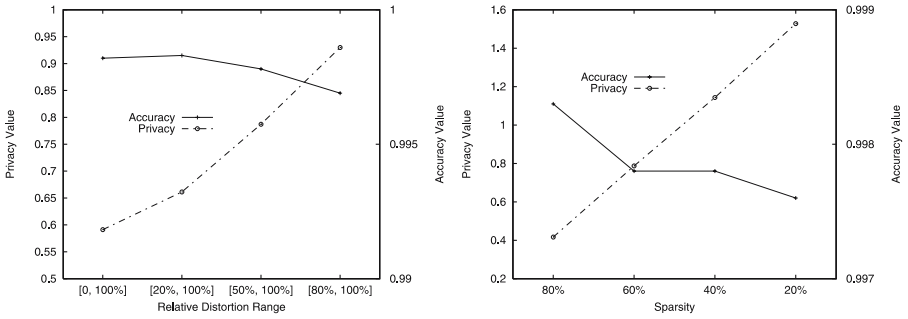
4.2.5. The interactive effect between privacy and accuracy

This experiment evaluated the interactive effect between privacy and accuracy with adjusted distortions. More specifically, we investigated the relative trend of privacy and accuracy with the change of distortion ranges, cube sparsity, and block sizes. Figure 9(a) shows the privacy and accuracy values with the change of distortion ranges. The privacy of adjusted distortion is increased as the lower bound of distortion ranges increases. However, the accuracy is decreased with the increase of the

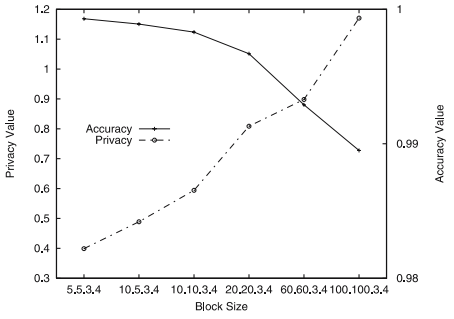


(a) Privacy (b) Accuracy

Fig. 8. The effect of varying the upper bound of distortion range



(a) Different relative distortion range (b) Different sparsity



(c) Different block size

Fig. 9. Relationship between privacy and accuracy

lower bound of distortion ranges. In other words, there is a trade-off between privacy and accuracy. The same trade-off effect can also be observed in Figs. 9(b) and 9(c). These two figures show the privacy and accuracy with the changes of cube sparsity and block sizes, respectively.

4.3. Summary of experimental results and discussions

Summary: Several major experimental results are summarised as follows.

- The accuracy of adjusted distortions is significantly and systematically better than the accuracy of original distortions. This is due to the fact that the zero-sum method constrains the marginal sums of distortions to be zero. In Sect. 3.2.1, we have shown in Fig. 2 that the partially covered blocks may yield the inaccuracy of range-sum query. Data sets with adjusted distortions may guarantee yielding higher accuracy of the range-sum query.
- Based on our experimental results, using relative data distortion, our zero-sum method obtains better privacy in many cases than the privacy of original distortions. This is because each marginal sum of the original distortion is redistributed back to each cell in the block. If the same amount of adjustment is applied to two different-valued cells, the cell with smaller value has more significant change in comparison with the change in the cell with the larger value. Therefore, when computing relative privacy, the gains by smaller-valued cells will exceed the losses by larger-valued cells. Thus, in general, our method can achieve better results.
- When the zero-sum method is applied, there is a trade-off between privacy and accuracy: higher privacy preserves better the confidentiality of the data by scarcifying the accuracy of range-sum query.

Discussions: We now address some limitations of the zero-sum method and several issues related to its experimental evaluation.

- The cost of data preparation for the zero-sum method is relatively high. For instance, it took about 20 hours to create the original distortion and adjusted distortion from the raw data cube generated by APB Benchmark. However, this is only one time expense for a data cube, and the subsequent query response time is not affected. For OLAP applications, the query response time is rather more critical. In contrast, many other privacy-preserving methods, such as query restriction or access control, require extra processing time each time when answering a query. This additional processing time will significantly slow down the response time.
- A common concern among people is to figure out what level of privacy would be reasonable for practical purposes. Generally speaking, a 100% privacy with 5% to 10% loss of accuracy is considered to be a satisfactory level. Another indicator as a good level of privacy is 25%. This is because many people consider 50% of distortion as the maximum tolerance for bias; beyond 50% is considered as too biased (Adam et al. 1989). Therefore, if we take half of the maximum tolerance for bias, the level of privacy at 25% is considerably acceptable.

5. Conclusions and future work

In this paper, we proposed an effective approach, called the zero-sum method, for preserving data privacy in data cubes. This method provides an accurate estimation of the summation for range queries while preserving the confidential information in individual data cells. The privacy preservation of this method is based on relative random data distortion techniques rather than simple unadjusted original random data distortion techniques. We provided a theoretical analysis of the performance of the

proposed method. Our experimental results showed that our method can achieve both better privacy preservation and range-query accuracy than the unadjusted original random data distortion method.

There are several directions for future work on this topic. First, the time complexity of the zero-sum method is $k \times n_1 \times \ldots \times n_k$, where k is the number of dimensions and n_i is the size of the ith dimension of a data cube. The cost is relatively high for higher dimensional data cubes. There may be a way to improve the computation performance of the proposed method. Second, we analysed privacy preservation for range-sum queries in a data cube. The privacy preservation for other query types should also be investigated for the sake of practical applications.

References

Adam NR, Wortman JC (1989) Security-control methods for statistical databases. ACM Comput Surv 21(4):515–556

Agrawal D, Aggarwal CC (2001) On the design and quantification of privacy preserving data mining algorithms. In: Proc of the ACM symposium on principles of database systems, Santa Barbara, CA, USA, pp 247–255

Agrawal R, Gupta A, Sarawagi S (1997) Modeling multidimensional databases. In: Proc of the 13th international conference on data engineering, Birmingham, UK, pp 232–243

Agrawal R, Srikant R (2000) Privacy-preserving data mining. In: Proc of the ACM SIGMOD conference on management of data, Dallas, TX, USA, pp 439–450

Barbara D, DuMouched W, Faloutsos C, Haas PJ, Hellerstein JM, Ioannidis Y (1997) The New Jersey data reduction report. Data Eng Bull 20:3–45

Barbara D, Sullivan M (1997) Quasi-cubes: exploiting approximations in multidimensional databases. ACM SIGMOD Rec 26(3):12–17

Beck LL (1980) A security mechanism for statistical databases. ACM TODS 5(3):316–338

Chaudhuri S, Dayal U (1997) An overview of data warehousing and OLAP technology. SIGMOD Rec 26(1):65–74

Conway R, Strip D (1976) Selective partial access to a database. In: Proc ACM annual conf, pp 85–89

Denning DE (1980) Secure statistical database with random sample queries. ACM TODS 5(3):291–315

Denning DE (1982) Cryptography and data security. Addison-Wesley

Denning DE, Denning PJ, Schwartz MD (1979) The tracker: a threat to statistical database security. ACM TODS 4(1):76–96

Dobkin D, Jones AK, Lipton RJ (1979) Secure database: protection against user influence. ACM TODS 4(1):97–106

Estivill-Castro V, Brankovic L (1999) Data swapping: balancing privacy against precision in mining for logic rules. In: Proc of international conference of data warehousing and knowledge discovery, Florence, Italy, pp 389–398

Evfimievski A, Srikant R, Agrawal R, Gehrke J (2002) Privacy preserving mining of association rules. In: Proc of the 8th ACM SIGKDD int'l conference on knowledge discovery in databases and data mining, Edmonton, Canada, pp 217–228

Faloutsos C, Jagadish H, Sidiropoulos N (1997) Recovering information from summary data. In: Proc of the 1997 VLDB. Athens, Greece, pp 36–45

Fellegi IP (1972) On the question of statistical confidentiality. Am Stat Assoc 67(337):7–18

Gray J, Bosworth A, Layman A, Pirahesh H (1996) Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub-total. In: Proc of 12th international conference on data engineering, pp 152–159

Ho CT, Agrawal R, Megiddo N, Srikant R (1997) Range queries in OLAP data cubes. SIGMOD, Tucson, AZ, USA, pp 73–88

Kantarcioglu M, Clifton C (2002) Privacy-preserving distributed mining of association rules on horizontally partitioned data. The ACM SIGMOD workshop on research issues in data mining and knowledge discovery, Madison, WI, pp 24–31

Kargupta H, Datta S, Wang Q, Sivakumar K (2003) On the privacy preserving properties of random data perturbation techniques. In: Proc of 2003 IEEE international conference on data mining. Melbourne, FL, pp 99–106

Kimball R (1997) Ensuring that your data warehouse is secure. DBMS Mag 10(4):14

Lee SY, Ling TW, Li HG (2000) Hierarchical compact cube for range-max queries. In: Proc of the 26th international conference on VLDB, Cairo, Egypt, pp 232–241

Liew CK, Choi U, Liew CJ (1985) A data distortion by probability distribution. ACM TODS 10(3):395–411

Motwani R, Raghavan P (1995) Randomized algorithms. Cambridge University Press

OLAP council's release II of the analytical processing benchmark (APB-1) for OLAP server performance, http://www.olapcouncil.org/news/APB1r2b_PR.htm, November 16, 1998

Priebe T, Pernul G (2000) Towards OLAP security design—survey and research issues. In: Proc of the 3rd ACM international workshop on data warehousing and OLAP, McLean, VA, USA, pp 33–40

Shoshani A (1997) OLAP and statistical databases: similarities and differences. In: Proc of the 16th ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems, Tucson, AZ, USA, pp 185–196

Traub JF, Yemini Y, Waznaikowski H (1984) The statistical security of a statistical database. ACM TODS 9(4):672–679

Author biographies

Sam Y. Sung is an Associate Professor in the Department of Computer Science, School of Computing, National University of Singapore. He received a B.Sc. from the National Taiwan University in 1973, the M.Sc. and Ph.D. in computer science from the University of Minnesota in 1977 and 1983, respectively. He was with the University of Oklahoma and University of Memphis in the United States before joining the National University of Singapore. His research interests include information retrieval, data mining, pictorial databases and mobile computing. He has published more than 80 papers in various conferences and journals, including IEEE Transaction on Software Engineering, IEEE Transaction on Knowledge & Data Engineering, etc.

Yao Liu received the B.E. degree in computer science and technology from Peking University in 1996 and the MS. degree from the Software Institute of the Chinese Science Academy in 1999. Currently, she is a Ph.D. candidate in the Department of Computer Science at the National University of Singapore. Her research interests include data warehousing, database security, data mining and high-speed networking.

Hui Xiong received the B.E. degree in Automation from the University of Science and Technology of China, Hefei, China, in 1995, the M.S. degree in Computer Science from the National University of Singapore, Singapore, in 2000, and the Ph.D. degree in Computer Science from the University of Minnesota, Minneapolis, MN, USA, in 2005. He is currently an Assistant Professor of Computer Information Systems in the Management Science & Information Systems Department at Rutgers University, NJ, USA. His research interests include data mining, databases, and statistical computing with applications in bioinformatics, database security, and self-managing systems. He is a member of the IEEE Computer Society and the ACM.

Peter A. Ng is currently the Chairperson and Professor of Computer Science at the University of Texas–Pan American. He received his Ph.D. from the University of Texas–Austin in 1974. Previously, he had served as the Vice President at the Fudan International Institute for Information Science and Technology, Shanghai, China, from 1999 to 2002, and the Executive Director for the Global e-Learning Project at the University of Nebraska at Omaha, 2000–2003. He was appointed as an Advisory Professor of Computer Science at Fudan University, Shanghai, China in 1999. His recent research focuses on document and information-based processing, retrieval and management. He has published many journal and conference articles in this area. He had served as the Editor-in-Chief for the Journal on Systems Integration (1991–2001) and as Advisory Editor for the Data and Knowledge Engineering Journal since 1989.

Correspondence and offprint requests to: Sam Y. Sung, Department of Computer Science, National University of Singapore, 3 Science Drive 2, Singapore 117543. Email: ssung@comp.nus.edu.sg