A Constraint-Based Framework for Computing Privacy Preserving OLAP Aggregations on Data Cubes

Alfredo Cuzzocrea[1], Domenico Saccà[2]

[1] ICAR-CNR and University of Calabria
87036 Cosenza, Italy
cuzzocrea@si.deis.unical.it
[2] DEIS Dept, University of Calabria
87036 Cosenza, Italy
sacca@deis.unical.it

Abstract. A *constraint-based framework for computing privacy preserving OLAP aggregations on data cubes* is proposed and experimentally assessed in this paper. Our framework introduces a *novel privacy OLAP notion*, which, following consolidated paradigms of OLAP research, *looks at the privacy of aggregate patterns defined on multidimensional ranges rather than the privacy of individual tuples/data-cells*, like similar efforts in privacy preserving database and data-cube research. To this end, we devise a *threshold-based* method that aims at *simultaneously accomplishing* the so-called *privacy constraint*, which inferiorly bounds the *inference error*, and the so-called *accuracy constraint*, which superiorly bounds the *query error*, on OLAP aggregations of the target data cube, following a *best-effort approach*. Finally, we complete our main theoretical contribution by means of an experimental evaluation and analysis of the effectiveness of our proposed framework on synthetic, benchmark and real-life data cubes.

1 Introduction

Following state-of-the-art initiatives in the context of *privacy preserving OLAP* (e.g., [2,9,12]), in this paper we propose an innovative framework based on *flexible sampling-based data cube compression techniques for computing privacy preserving OLAP aggregations on data cubes while allowing approximate answers to be efficiently evaluated over such aggregations*. This framework addresses an application scenario where, given a multidimensional data cube A stored in a *producer* Data Warehouse server, a collection of multidimensional portions of A defined by a given (range) *query-workload* QWL of interest must be published online for *consumer* OLAP client applications. Moreover, after published, the collection of multidimensional portions is no longer connected to the Data Warehouse server, and updates are handled from the scratch at each new online data delivery. The query-workload QWL is cooperatively determined by the Data Warehouse server and OLAP client applications, mostly depending on OLAP analysis goals of client applications, and other parameters such as business processes and requirements, frequency of accesses, and locality. OLAP client applications wish for retrieving summarized knowledge from A via adopting a *complex multi-resolution query model* whose components are (*i*) queries of QWL and, for each query Q of QWL, (*ii*) *sub-queries* of Q (i.e., in a multi-resolution fashion). To this end, for each query Q of QWL, an *accuracy grid* $G(Q)$, whose cells model sub-queries of interest, is defined. While aggregations of (authorized) queries and (authorized) sub-queries in QWL are disclosed to OLAP client applications, it must be avoided that, by meaningfully combining aggregate patterns extracted from multidimensional ranges associated to queries and sub-queries in QWL, malicious users could infer sensitive knowledge about other multidimensional portions of A that, due to privacy reasons, are hidden to unauthorized users. Furthermore, in our reference application scenario, target data cubes are also massive in size, so that data compression techniques are needed in order to efficiently evaluate queries, yet introducing *approximate answers* having a certain *degree of approximation* that,

however, is perfectly tolerable for OLAP analysis goals [3]. In our proposal, the described application scenario with accuracy and privacy features is accomplished by means of the so-called *accuracy/privacy contract*, which determines the *accuracy/privacy constraint* under which client applications must access and process multidimensional data. In this contract, the Data Warehouse server and client OLAP applications play the role of mutual subscribers, respectively.

Given a multidimensional range R of a data cube A, an *aggregate pattern* over R is defined as an aggregate value extracted from R that is able of providing a "description" of data stored in R. In order to capture the privacy of aggregate patterns, in this paper we introduce a *novel notion of privacy OLAP*. According to this novel notion, given a data cube A *the privacy preservation of* A *is modeled in terms of the privacy preservation of aggregate patterns defined on multidimensional data stored in* A. Therefore, we say that a data cube A is privacy preserving iff aggregate patterns extracted from A are privacy preserving. Contrary to our innovative privacy OLAP notion above, previous privacy preserving OLAP proposals totally neglect this even-relevant theoretical aspect, and, inspired by well-established techniques that focus on the privacy preservation of relational tuples [10,7], mostly focus on the privacy preservation of data cells (e.g., [9]) accordingly.

2 Basic Constructs and Definitions

A *data cube* A defined over a relational data source S is a tuple $A = \langle D,\mathcal{F},\mathcal{H},\mathcal{M}\rangle$, such that: (*i*) D is the data domain of A containing (OLAP) data cells, which are the basic aggregations of A computed over relational tuples stored in S; (*ii*) $\mathcal{F}$ is the set of *dimensions* of A, i.e. the *functional attributes* with respect to which the underlying OLAP analysis is defined (in other words, $\mathcal{F}$ is the set of attributes along which tuples in S are aggregated); (*iii*) $\mathcal{H}$ is the set of *hierarchies* related to the dimensions of A, i.e. hierarchical representations of the functional attributes shaped in the form of general trees; (*iv*) $\mathcal{M}$ is the set of *measures* of A, i.e. the *attributes of interest* for the underlying OLAP analysis (in other words, $\mathcal{M}$ is the set of attributes taken as argument of SQL aggregations whose results are stored in data cells of A). Given these definitions, (*i*) $|\mathcal{F}|$ denotes the number of dimensions of A, (*ii*) $d \in \mathcal{F}$ a generic dimension of A, (*iii*) $|d|$ the cardinality of d, and (*iv*) $H(d) \in \mathcal{H}$ the hierarchy related to d. Finally, for the sake of simplicity, we assume to deal with data cubes having a single measure (i.e., $|\mathcal{M}| = 1$). However, extending schemes, models and algorithms proposed in this paper as to deal with data cubes having *multiple measures* (i.e., $|\mathcal{M}| > 1$) is straightforward.

Given an $|\mathcal{F}|$-dimensional data cube A, an *m-dimensional range-query* Q against A, with $m \leq |\mathcal{F}|$, is a tuple $Q = \langle R_{k_0}, R_{k_1}, ..., R_{k_{m-1}}, \mathcal{A}\rangle$, such that: (*i*) R_{k_i} denotes a *contiguous* range defined on the dimension d_{k_i} of A, with k_i belonging to the range [0, $|\mathcal{F}|$–1], and (*ii*) $\mathcal{A}$ is a SQL aggregation operator. The evaluation of Q over A returns the $\mathcal{A}$-based aggregation computed over the set of data cells in A contained within the multidimensional sub-domain of A bounded by the ranges $R_{k_0}, R_{k_1}, ..., R_{k_{m-1}}$ of Q. Range-SUM queries, which return the SUM of the involved data cells, are trendy examples of range-queries. In our framework, we take into consideration range-SUM queries, as SUM aggregations are very popular in OLAP, and efficiently support summarized knowledge extraction from massive amounts of multidimensional data as well as other SQL aggregations (e.g., COUNT, AVG etc). Therefore, our framework can be straightforwardly extended as to deal with other SQL aggregations different from SUM. However, the latter research aspect is outside the scope of this paper, thus left as future work.

Given a query Q against a data cube A, the *query region* of Q, denoted by $R(Q)$, is defined as the sub-domain of A bounded by the ranges $R_{k_0}, R_{k_1}, ..., R_{k_{m-1}}$ of Q.

Given an m-dimensional query Q, the accuracy grid $\mathcal{G}(Q)$ of Q is a tuple $\mathcal{G}(Q) = \langle \Delta\ell_{k_0}, \Delta\ell_{k_1}, ..., \Delta\ell_{k_{m-1}}\rangle$, such that $\Delta\ell_{k_i}$ denotes the range partitioning Q along the dimension

d_{k_i} of A, with k_i belonging to $[0, |\mathcal{F}|-1]$, in a $\Delta\ell_{k_i}$-based (one-dimensional) partition. By combining the one-dimensional partitions along *all* the dimensions of Q, we finally obtain $\mathcal{G}(Q)$ as a *regular multidimensional partition* of $R(Q)$. From Section 1, recall that the elementary cell of the accuracy grid $\mathcal{G}(Q)$ is implicitly defined by sub-queries of Q belonging to the query-workload QWL against the target data cube.

Based on the latter definitions, in our framework we consider the broader concept of *extended range-query* Q^+, defined as a tuple $Q^+ = \langle Q, \mathcal{G}(Q)\rangle$, such that (*i*) Q is a "classical" range-query, $Q = \langle R_{k_0}, R_{k_1}, ..., R_{k_{m-1}}, \mathcal{A}\rangle$, and (*ii*) $\mathcal{G}(Q)$ is the accuracy grid associated to Q, $\mathcal{G}(Q) = \langle \Delta\ell_{k_0}, \Delta\ell_{k_1}, ..., \Delta\ell_{k_{m-1}}\rangle$, with the condition that each interval $\Delta\ell_{k_i}$ is defined on the *corresponding* range R_{k_i} of the dimension d_{k_i} of Q. For the sake of simplicity, here and in the remaining part of the paper we assume $Q \equiv Q^+$.

Given an n-dimensional data domain D, we introduce the *volume* of D, denoted by $\|D\|$, as follows: $\|D\| = |d_0| \times |d_1| \times \ldots \times |d_{n-1}|$, such that $|d_i|$ is the cardinality of the dimension d_i of D. This definition can also be extended to a multidimensional data cube A, thus introducing the volume of A, $\|A\|$, and to a multidimensional range-query Q, thus introducing the volume of Q, $\|Q\|$.

Given a data cube A, a range query-workload QWL against A is defined as a *collection* of (range) queries against A, as follows: $QWL = \{Q_0, Q_1, ..., Q_{|QWL|-1}\}$, with $R(Q_k) \subseteq R(A)\ \forall\ Q_k \in QWL$.

Given a query-workload $QWL = \{Q_0, Q_1, \ldots, Q_{|QWL|-1}\}$, we say that QWL is *non-overlapping* if there not exist two queries Q_i and Q_j belonging to QWL such that $R(Q_i) \cap R(Q_j) \neq \emptyset$. Given a query-workload $QWL = \{Q_0, Q_1, \ldots, Q_{|QWL|-1}\}$, we say that QWL is *overlapping* if there exist two queries Q_i and Q_j belonging to QWL such that $R(Q_i) \cap R(Q_j) \neq \emptyset$. Given a query-workload $QWL = \{Q_0, Q_1, \ldots, Q_{|QWL|-1}\}$, the *region set* of QWL, denoted by $R(QWL)$, is defined as the *collection* of regions of queries belonging to QWL, as follows: $R(QWL) = \{R(Q_0), R(Q_1), \ldots, R(Q_{|QWL|-1})\}$.

3 Accuracy Metrics

As accuracy metrics for answers to queries of the target query-workload QWL, we make use of the *relative query error* between exact and approximate answers, which is a well-recognized-in-literature measure of quality for approximate query answering techniques in OLAP (e.g., see [3]).

Formally, given a query Q_k of QWL, we denote as $A(Q_k)$ the exact answer to Q_k (i.e., the answer to Q_k evaluated over the original data cube A), and as $\tilde{A}(Q_k)$ the approximate answer to Q_k (i.e., the answer to Q_k evaluated over the synopsis data cube A'). Therefore, the relative query error $E_Q(Q_k)$ between $A(Q_k)$ and $\tilde{A}(Q_k)$ is defined as follows: $E_Q(Q_k) = \dfrac{|A(Q_k) - \tilde{A}(Q_k)|}{\max\{A(Q_k), 1\}}$.

$E_Q(Q_k)$ can be extended to the whole query-workload QWL, thus introducing the *average relative query error* $\bar{E}_Q(QWL)$ that takes into account the contributions of relative query errors of all the queries Q_k in QWL, each of them weighted by the volume of the query, $\|Q_k\|$, with respect to the whole volume of queries in QWL, i.e. the *volume of QWL*, $\|QWL\|$. $\|QWL\|$ is defined as follows: $\|QWL\| = \sum_{k=0}^{|QWL|-1} \|Q_k\|, Q_k \in QWL$.

Based on the previous definition of $\|QWL\|$, the average relative query error $\bar{E}_Q(QWL)$ for a given query-workload QWL can be expressed as a *weighted linear combination* of relative query

errors $E_Q(Q_k)$ of all the queries Q_k in QWL, as follows: $\overline{E}_Q(QWL) = \sum_{k=0}^{|QWL|-1} \frac{\|Q_k\|}{\|QWL\|} \cdot E_Q(Q_k)$, i.e.: $\overline{E}_Q(QWL) = \sum_{k=0}^{|QWL|-1} \frac{\|Q_k\|}{\sum_{j=0}^{|QWL|-1} \|Q_j\|} \cdot \frac{|A(Q_k) - \tilde{A}(Q_k)|}{\max\{A(Q_k),1\}}$, under the constraint: $\sum_{k=0}^{|QWL|-1} \frac{\|Q_k\|}{\|QWL\|} = 1$.

4 Privacy Metrics

Since we deal with the problem of ensuring the privacy preservation of OLAP aggregations, our privacy metrics takes into consideration how sensitive knowledge can be discovered from aggregate data, and tries to limit this possibility. On a theoretical plane, this is modeled by the privacy OLAP notion introduced in Section 1.

To this end, we first study how sensitive aggregations can be discovered from the target data cube A. Starting from the knowledge about A (e.g., range sizes, OLAP hierarchies etc), and the knowledge about a given query Q_k belonging to the query-workload QWL (i.e., the volume of Q_k, $\|Q_k\|$, and the exact answer to Q_k, $A(Q_k)$), it is possible to infer knowledge about sensitive ranges of data contained within $R(Q_k)$. For instance, it is possible to derive the average value of the contribution throughout which each basic data cell of A within $R(Q_k)$ contributes to $A(Q_k)$, which we name as *singleton aggregation* $I(Q_k)$. $I(Q_k)$ is defined as follows: $I(Q_k) = \frac{A(Q_k)}{\|Q_k\|}$.

It is easy to understand that, starting from the knowledge about $I(Q_k)$, it is possible to *progressively* discover aggregations of larger range of data within $R(Q_k)$, rather than the one stored within the basic data cell, thus inferring even-more-useful sensitive knowledge. Also, by exploiting OLAP hierarchies and the well-known roll-up operator, it is possible to discover aggregations of ranges of data at higher degrees of such hierarchies. It should be noted that the singleton aggregation model $I(Q_k)$ above represents indeed an *instance* of our privacy OLAP notion target to the problem of preserving the privacy of range-SUM queries (the focus of our paper). As a consequence, $I(Q_k)$ is essentially based on the conventional SQL aggregation operator AVG. Despite this, the underlying theoretical model we propose is general enough to be straightforwardly extended as to deal with more sophisticated privacy OLAP notion instances, depending on the particular class of OLAP queries considered. Without loss of generality, given a query Q_k belonging to an OLAP query class C, in order to handle the privacy preservation of Q_k we only need to define the formal expression of the related singleton aggregation $I(Q_k)$ (like the previous one for the specific case of range-SUM queries). Then, the theoretical framework we propose works at the same way.

Secondly, we study how OLAP client applications can discover sensitive aggregations from the knowledge about approximate answers, and, similarly to the previous case, from the knowledge about data cube and query metadata. Starting from the knowledge about the synopsis data cube A', and the knowledge about the answer to a given query Q_k belonging to the query-workload QWL, it is possible to derive an *estimation* on $I(Q_k)$, denoted by $\tilde{I}(Q_k)$, as follows: $\tilde{I}(Q_k) = \frac{\tilde{A}(Q_k)}{S(Q_k)}$, such that $S(Q_k)$ is the *number of samples* effectively extracted from $R(Q_k)$ to compute A' (note that $S(Q_k) < \|Q_k\|$). The relative difference between $I(Q_k)$ and $\tilde{I}(Q_k)$, named as *relative inference error* and denoted by $E_I(Q_k)$, gives us a metrics for the privacy of $\tilde{A}(Q_k)$, which is defined as follows: $E_I(Q_k) = \frac{|I(Q_k) - \tilde{I}(Q_k)|}{\max\{I(Q_k),1\}}$.

Indeed, while OLAP client applications are aware about the definition and metadata of both the target data cube and queries of the query-workload *QWL*, the number of samples $S(Q_k)$ (for each query Q_k in *QWL*) is not disclosed to them. As a consequence, in order to model this aspect of our framework, we introduce the *user-perceived singleton aggregation*, denoted by $\tilde{I}_U(Q_k)$, which is the *effective* singleton aggregation *perceived* by external applications based on the knowledge made available to them. $\tilde{I}_U(Q_k)$ is defined as follows: $\tilde{I}_U(Q_k) = \frac{\tilde{A}(Q_k)}{\|Q_k\|}$.

Based on $\tilde{I}_U(Q_k)$, we derive the definition of the *relative user-perceived inference error* $E_I^U(Q_k)$, as follows: $E_I^U(Q_k) = \frac{|I(Q_k) - \tilde{I}_U(Q_k)|}{\max\{I(Q_k),1\}}$.

Since $S(Q_k) < \|Q_k\|$, it is trivial to demonstrate that $\tilde{I}_U(Q_k)$ provides a better estimation of the singleton aggregation of Q_k rather than that provided by $\tilde{I}(Q_k)$, as $\tilde{I}_U(Q_k)$ is evaluated with respect to *all* the items contained within $R(Q_k)$ (i.e., $\|Q_k\|$), whereas $\tilde{I}(Q_k)$ is evaluated with respect to the effective number of samples extracted from $R(Q_k)$ (i.e., $S(Q_k)$). In other words, $\tilde{I}_U(Q_k)$ is an *upper bound* for $\tilde{I}(Q_k)$. Therefore, in our framework we consider $\tilde{I}(Q_k)$ to compute the synopsis data cube, whereas we consider $\tilde{I}_U(Q_k)$ to model inference issues on the OLAP client application side.

$E_I^U(Q_k)$ can be extended to the whole query-workload *QWL*, by considering the *average relative inference error* $\overline{E}_I(QWL)$ that takes into account the contributions of relative inference errors $E_I(Q_k)$ of all the queries Q_k in *QWL*. Similarly to what done for the average relative query error $\overline{E}_Q(QWL)$, we model $\overline{E}_I(QWL)$ as follows: $\overline{E}_I(QWL) = \sum_{k=0}^{|QWL|-1} \frac{\|Q_k\|}{\|QWL\|} \cdot E_I(Q_k)$, i.e.: $\overline{E}_I(QWL) = \sum_{k=0}^{|QWL|-1} \frac{\|Q_k\|}{\sum_{j=0}^{|QWL|-1} \|Q_j\|} \cdot \frac{|I(Q_k) - \tilde{I}_U(Q_k)|}{\max\{I(Q_k),1\}}$, under the constraint: $\sum_{k=0}^{|QWL|-1} \frac{\|Q_k\|}{\|QWL\|} = 1$.

Note that, as stated above, $\overline{E}_I(QWL)$ is defined in dependence on $\tilde{I}_U(Q_k)$ rather than $\tilde{I}(Q_k)$. For the sake of simplicity, here and in the remaining part of the paper we assume $E_I(Q_k) \equiv E_I^U(Q_k)$.

Concepts and definitions above allow us to introduce the *singleton aggregation privacy preserving model* $\mathcal{X} = \langle I(\bullet), \tilde{I}(\bullet), \tilde{I}_U(\bullet) \rangle$, which is a fundamental component of the privacy preserving OLAP framework we propose. $\mathcal{X}$ properly realizes our privacy OLAP notion.

Given a query $Q_k \in QWL$ against the target data cube A, in order to preserve the privacy of Q_k under our privacy OLAP notion, we must *maximize the inference error* $E_I(Q_k)$ *while minimizing the query error* $E_Q(Q_k)$. While the definition of $E_Q(Q_k)$ can be reasonably considered as an *invariant* of our theoretical model, the definition of $E_I(Q_k)$ strictly depends on $\mathcal{X}$. Therefore, given a particular class of OLAP queries C, in order to preserve the privacy of queries of kind C, we only need to *appropriately* define $\mathcal{X}$. This nice amenity states that the privacy preserving OLAP framework we propose is orthogonal to the particular class of queries considered, and can be straightforwardly adapted to a large family of OLAP query classes.

5 Thresholds

Similarly to related proposals appeared in literature recently [9], in our framework we introduce the accuracy threshold Φ_Q and the privacy threshold Φ_I. Φ_Q and Φ_I give us an *upper bound* for the average relative query error $\overline{E}_Q(QWL)$ and a *lower bound* for the average relative inference error $\overline{E}_I(QWL)$ of a given query-workload QWL against the synopsis data cube A', respectively. As stated in Section 1, Φ_Q and Φ_I allow us to meaningfully model and treat the accuracy/privacy constraint by means of rigorous mathematical/statistical models.

In our application scenario, Φ_Q and Φ_I are cooperatively negotiated by the Data Warehouse server and OLAP client applications. The issue of determining how to set these parameters is a non-trivial engagement. Intuitively enough, for what regards the accuracy of answers, it is possible to (*i*) refer to the widely-accepted *query error threshold* belonging to the interval [15, 20] % that, according to results of a plethora of research experiences in the context of approximate query answering techniques in OLAP (e.g., see [4]), represents the current state-of-the-art, and (*ii*) use it as baseline to trade-off the parameter Φ_Q. For what regards the privacy of answers, there are not immediate guidelines to be considered since privacy preserving techniques for advanced data management (like OLAP) are relatively new hence we cannot refer to any widely-accepted threshold like happens with approximate query answering techniques. As a result, the parameter Φ_I can be set according to a *two-step approach* where *first* the accuracy constraint is accomplished in dependence of Φ_Q, and *then* Φ_I is *consequently* set by trying to maximize it (i.e., augmenting the privacy of answers) *as much as possible, thus following a best-effort approach*.

6 Computing the Synopsis Data Cube via a Constraint-based Approach

From the Sections above, it follows that our privacy preserving OLAP technique, which is finally implemented by greedy algorithm `computeSynDataCube`, encompasses three main phases: (*i*) allocation of the input storage space B, (*ii*) sampling of the input data cube A, (*iii*) refinement of the synopsis data cube A'. In this Section, we present in detail these phases.

6.1 The Allocation Phase. Given the input data cube A, the target query-workload QWL, and the storage space B, in order to compute the synopsis data cube A' *the first issue to be considered is how to allocate B across query regions of QWL*. Given a query region $R(Q_k)$, allocating an amount of storage space to $R(Q_k)$, denoted by $B(Q_k)$, corresponds to assign to $R(Q_k)$ a certain number of samples that can be extracted from $R(Q_k)$, denoted by $N(Q_k)$. To this end, during the allocation phase of algorithm `computeSynDataCube`, we *assign more samples to those query regions of QWL having skewed* (*i.e., irregular and asymmetric*) *data distributions* (*e.g., Zipf*)*, and less samples to those query regions having Uniform data distributions*. The idea underlying such an approach is that few samples are enough to "describe" Uniform query regions as data distributions of such regions are "regular", whereas we need more samples to "describe" skewed query regions as data distributions of such regions are, contrary to the previous case, not "regular". Specifically, we face-off the deriving allocation problem by means of a *proportional storage space allocation scheme*, which allows us to efficiently allocate B across query regions of QWL via assigning a *fraction* of B to each region. This allocation scheme has been preliminarily proposed in [5] for the different context of approximate query answering techniques for two-dimensional OLAP data cubes, and, in this work, it is extended as to deal with multidimensional data cubes and (query) regions.

First, if QWL is overlapping (see Section 3), we compute its *equivalent non-overlapping query-workload*, denoted by QWL', as follows. For each pair of overlapping queries Q_i and Q_j in QWL having R_{ij} as intersection region (i.e., $Q_i \cap Q_j = R_{ij}$), we add to QWL' a new set of queries composed by the query $Q_k \equiv R_{ij}$ plus all the queries corresponding to intersection regions originated via prolonging the ranges of R_{ij} along the dimensions of Q_i and Q_j, respectively. All the remaining queries in QWL are kept unchanged in QWL'. Hence, in both cases (i.e., QWL is overlapping or not) a set of regions $R(QWL) = \{R(Q_0), R(Q_1), \ldots, R(Q_{|QWL|-1})\}$ is obtained. Let $R(Q_k)$ be a region belonging to

$R(QWL)$, the amount of storage space allocated to $R(Q_k)$, $B(Q_k)$, is determined according to a proportional approach that considers (*i*) the nature of the data distribution of $R(Q_k)$ and geometrical issues of $R(Q_k)$, and (*ii*) the latter parameters of $R(Q_k)$ in proportional comparison with the same parameters of all the regions in $R(QWL)$, as follows: $B(Q_k) = \left\lfloor \frac{\varphi(R(Q_k)) + \Psi(R(Q_k)) \cdot \xi(R(Q_k))}{\sum_{h=0}^{|QWL|-1} \varphi(R(Q_k)) + \sum_{h=0}^{|QWL|-1} \Psi(R(Q_k)) \cdot \xi(R(Q_k))} \cdot B \right\rfloor$, such that [5]: (*i*) $\Psi(R)$ is a Boolean *characteristic function* that, given a region R, allows us to decide if data in R are Uniform or skewed; (*ii*) $\varphi(R)$ is a factor that captures the *skewness* and the *variance* of R in a combined manner; (*iii*) $\xi(R)$ is a factor that provides the ratio between the skewness of R and its standard deviation, which, according to [8], allows us to estimate the *skewness degree* of the data distribution of R. Previous formula can be extended as to handle the overall allocation of B across regions of QWL, thus achieving the formal definition of our proportional storage space allocation scheme, denoted by $\mathcal{W}(A,R(Q_0),R(Q_1),\ldots,R(Q_{|QWL|-1}),B)$, via the following system:

$$\begin{cases} B(Q_0) = \left\lfloor \frac{\varphi(R(Q_0)) + \Psi(R(Q_0)) \cdot \xi(R(Q_0))}{\sum_{k=0}^{|QWL|-1} \varphi(R(Q_k)) + \sum_{k=0}^{|QWL|-1} \Psi(R(Q_k)) \cdot \xi(R(Q_k))} \cdot B \right\rfloor \\ \ldots \\ B(Q_{|QWL|-1}) = \left\lfloor \frac{\varphi(R(Q_{|QWL|-1})) + \Psi(R(Q_{|QWL|-1})) \cdot \xi(R(Q_{|QWL|-1}))}{\sum_{k=0}^{|QWL|-1} \varphi(R(Q_k)) + \sum_{k=0}^{|QWL|-1} \Psi(R(Q_k)) \cdot \xi(R(Q_k))} \cdot B \right\rfloor \\ \sum_{k=0}^{|QWL|-1} B(Q_k) \leq B \end{cases} \quad (1)$$

In turn, for each query region $R(Q_k)$ of $R(QWL)$, we further allocate the amount of storage space $B(Q_k)$ across the sub-queries of Q_k, $q_{k,0}$, $q_{k,1}$, …, $q_{k,m-1}$, via using the *same* allocation scheme (1). Overall, this approach allows us to obtain a storage space allocation for each *sub-query* $q_{k,i}$ of QWL in terms of the maximum sample number $N(q_{k,i}) = \left\lfloor \frac{B(q_{k,i})}{32} \right\rfloor$ that can be extracted from $q_{k,i}$[1], being $B(q_{k,i})$ the amount of storage space allocated to $q_{k,i}$.

It should be noted that the approach above allows us to achieve an extremely-accurate level of detail in handling accuracy/privacy issues of the final synopsis data cube *A'*. To become convinced of this, recall that the granularity of OLAP client applications is *the one of queries* (see Section 1), which is *much greater* than the one of sub-queries (specifically, the latter depends on the degree of accuracy grids) we use as atomic unit of our reasoning. Thanks to this difference between granularity of input queries and accuracy grid cells, which, in our framework, is made "conveniently" high, we finally obtain a crucial *information gain* that allows us to efficiently accomplish the accuracy/privacy constraint.

6.2 The Sampling Phase Given an instance of our proportional allocation scheme (1), $\mathcal{W}$, during the second phase of algorithm `computeSynDataCube`, we sample the input data cube A in order to obtain the synopsis data cube *A'*, in such a way as to satisfy the accuracy/privacy constraint with respect to the target query-workload QWL. To this end, we apply a different strategy in dependence on the fact that query regions characterized by Uniform or skewed distributions are handled, according to similar insights that have inspired our allocation technique (see Section 6.1). Specifically, for a skewed region $R(q_{k,i})$, given the maximum number of samples that can be extracted

[1] Here, we are assuming that an integer is represented in memory by using 32 bits.

from $R(q_{k,i})$, $N(q_{k,i})$, we *sample the* $N(q_{k,i})$ *outliers of* $q_{k,i}$. It is worthy to notice that, for skewed regions, *sum of outliers represents an accurate estimation of the sum of all the data cells contained within such regions*. Also, it should be noted that this approach allows us to gain advantages with respect to approximate query answering as well as the privacy preservation of sensitive ranges of multidimensional data of skewed regions. Contrary to this, for a Uniform region $R(q_{k,i})$, given the maximum number of samples that can be extracted from $R(q_{k,i})$, $N(q_{k,i})$, let (*i*) $\overline{C}_{R(q_{k,i})}$ be the average of values of data cells contained within $R(q_{k,i})$, (*ii*) $\mathcal{U}(R(q_{k,i}),\overline{C}_{R(q_{k,i})})$ be the set of data cells C in $R(q_{k,i})$ such that *value*$(C) > \overline{C}_{R(q_{k,i})}$, where *value*$(C)$ denotes the value of C, and (*iii*) $\overline{C}^{\uparrow}_{R(q_{k,i})}$ be the average of values of data cells in $\mathcal{U}(R(q_{i,k}),\overline{C}_{R(q_{k,i})})$, we adopt the strategy of extracting $N(q_{k,i})$ samples from $R(q_{k,i})$ by selecting them as the $N(q_{k,i})$ *closer-to-*$\overline{C}^{\uparrow}_{R(q_{k,i})}$ *data cells* C *in* $R(q_{k,i})$ *such that value*$(C) > \overline{C}_{R(q_{k,i})}$. Just like previous considerations given for skewed regions, it should be noted that the above-described sampling strategy for Uniform regions allows us to meaningfully trade-off the need for efficiently answering range-SUM queries against the synopsis data cube, and the need for limiting the number of samples to be stored within the synopsis data cube.

In order to satisfy the accuracy/privacy constraint, the sampling phase aims at accomplishing (decomposed) accuracy and privacy constraints *separately*, based on a two-step approach Given a query region $R(Q_k)$, we *first* sample $R(Q_k)$ in such a way as to satisfy the accuracy constraint, and, *then*, we check if samples extracted from $R(Q_k)$ *also* satisfy, beyond the accuracy one, the privacy constraint. As mentioned in Section 4, this strategy follows a best-effort approach aiming at minimizing computational overheads due to computing the synopsis data cube, and it is also the conceptual basis of guidelines for setting the thresholds Φ_Q and Φ_I.

Moreover, our sampling strategy aims at obtaining a *tunable* representation of the synopsis data cube *A'*, which can be *progressively refined* until the accuracy/privacy constraint is satisfied as much as possible. This means that, given the input data cube A, we first sample A in order to obtain the *current* representation of *A'*. If such a representation satisfies the accuracy/privacy constraint, then the *final* representation of *A'* is achieved, and used at query time to answer queries instead of A. Otherwise, if the current representation of *A'* does not satisfy the accuracy/privacy constraint, then we perform "corrections" on the current representation of *A'*, thus refining such representation in order to obtain a final representation that satisfies the constraint, on the basis of a best-effort approach. What we call the *refinement process* (described in next Section 6.3) is based on a greedy approach that *"moves"*[2] *samples from regions of QWL whose queries satisfy the accuracy/privacy constraint to regions of QWL whose queries do not satisfy the constraint, yet ensuring that the former do not violate the constraint*.

Given a query Q_k of the target query-workload QWL, we say that Q_k satisfies the accuracy/privacy constraint iff the following inequalities simultaneously hold: $\begin{cases} E_Q(Q_k) \le \Phi_Q \\ E_I(Q_k) \ge \Phi_I \end{cases}$.

In turn, given a query-workload QWL, we decide about its *satisfiability* with respect to the accuracy/privacy constraint by inspecting the satisfiability of queries that compose QWL. Therefore, we say that QWL satisfies the accuracy/privacy constraint iif the following inequalities simultaneously hold: $\begin{cases} \overline{E}_Q(QWL) \le \Phi_Q \\ \overline{E}_I(QWL) \ge \Phi_I \end{cases}$.

Given the target query-workload QWL, the criterion of our greedy approach used during the refinement process is the *minimization* of the average relative query error, $\overline{E}_Q(QWL)$, and the *maximization* of the average relative inference error, $\overline{E}_I(QWL)$, within the *minimum* number of

[2] In Section 6.3, we describe in detail the meaning of "moving" samples between query regions.

movements that allows us to accomplish both the goals simultaneously (i.e., minimizing $\overline{E}_Q(QWL)$, and maximizing $\overline{E}_I(QWL)$). Furthermore, the refinement process is bounded by a *maximum occupancy of samples moved across queries of QWL*, which we name as *total buffer size* and denote as $\mathcal{L}_{A';QWL}$. $\mathcal{L}_{A';QWL}$ depends on several parameters such as the size of the buffer, the number of sample pages moved at each iteration, the overall available swap-memory etc.

6.3 The Refinement Phase In the refinement process, the third phase of algorithm `computeSynDataCube`, given the current representation of *A'* that does *not* satisfy the accuracy/privacy constraint with respect to the target query-workload *QWL*, we try to obtain an alternative representation of *A'* that satisfies the constraint, according to a best-effort approach. To this end, the refinement process encompasses the following steps: (*i*) sort queries in *QWL* according to their "distance" from the satisfiability condition, thus obtaining the ordered query set $QWL^{\mathcal{P}}$; (*ii*) select from $QWL^{\mathcal{P}}$ a pair of queries $Q^{\mathcal{T}}$ and $Q^{\mathcal{F}}$ such that (*ii.j*) $Q^{\mathcal{T}}$ is the query of $QWL^{\mathcal{P}}$ having the *greater positive distance* from the satisfiability condition, i.e. $Q^{\mathcal{T}}$ is the query of $QWL^{\mathcal{P}}$ that has the greater *surplus* of samples that can be moved towards queries in $QWL^{\mathcal{P}}$ that do not satisfy the satisfiability condition, and (*ii.jj*) $Q^{\mathcal{F}}$ is the query of $QWL^{\mathcal{P}}$ having the *greater negative distance* from the satisfiability condition, i.e. $Q^{\mathcal{F}}$ is the query of $QWL^{\mathcal{P}}$ that is in most need for new samples; (*iii*) move enough samples from $Q^{\mathcal{T}}$ to $Q^{\mathcal{F}}$ in such a way as to satisfy the accuracy/privacy constraint on $Q^{\mathcal{F}}$ while, at the same time, ensuring that $Q^{\mathcal{T}}$ does not violate the constraint; (*iv*) repeat steps (*i*), (*ii*), and (*iii*) until the current representation of *A'* satisfies, as much as possible, the accuracy/privacy constraint with respect to *QWL*, within the maximum number of iterations bounded by $\mathcal{L}_{A';QWL}$. For what regards step (*iii*), moving ρ samples from $Q^{\mathcal{T}}$ to $Q^{\mathcal{F}}$ means: (*i*) removing ρ samples from $R(Q^{\mathcal{T}})$, thus obtaining an *additional* space, said $B(\rho)$; (*ii*) allocating $B(\rho)$ to $R(Q^{\mathcal{F}})$, (*iii*) re-sampling $R(Q^{\mathcal{F}})$ by considering the additional number of samples that have became available – in practice, this means extracting from $R(Q^{\mathcal{F}})$ further ρ samples.

Let $S^*(Q_k)$ be the number of samples of a query $Q_k \in QWL$ satisfying the accuracy/privacy constraint. From the formal definitions of $E_Q(Q_k)$ (see Section 3), $I(Q_k)$, $\widetilde{I}(Q_k)$ and $E_I(Q_k)$ (see Section 4), and the satisfiability condition, it could be easily demonstrated that $S^*(Q_k)$ is given by the following formula: $S^*(Q_k) = \frac{(1-\Phi_Q)}{(1-\Phi_I)} \cdot \|Q_k\|$.

Let $S_{eff}(Q^{\mathcal{F}})$ and $S_{eff}(Q^{\mathcal{T}})$ be the numbers of samples *effectively* extracted from $R(Q^{\mathcal{F}})$ and $R(Q^{\mathcal{T}})$ during the previous sampling phase, respectively. Note that $S_{eff}(Q^{\mathcal{F}}) < S^*(Q^{\mathcal{F}})$ and $S_{eff}(Q^{\mathcal{T}}) \geq S^*(Q^{\mathcal{T}})$. It is easy to prove that the number of samples to be moved from $Q^{\mathcal{T}}$ to $Q^{\mathcal{F}}$ such that $Q^{\mathcal{F}}$ satisfies the accuracy/privacy constraint and $Q^{\mathcal{T}}$ does not violate the constraint, denoted by $S_{mov}(Q^{\mathcal{T}},Q^{\mathcal{F}})$, is finally given by the following formula: $S_{mov}(Q^{\mathcal{T}},Q^{\mathcal{F}}) = S^*(Q^{\mathcal{F}}) - S_{eff}(Q^{\mathcal{F}})$, under the constraint: $S_{mov}(Q^{\mathcal{T}},Q^{\mathcal{F}}) < S_{eff}(Q^{\mathcal{T}}) - S^*(Q^{\mathcal{T}})$.

Without going in details, it is possible to demonstrate that, given (*i*) an *arbitrary* data cube *A*, (*ii*) an *arbitrary* query-workload *QWL*, (*iii*) an arbitrary pair of thresholds Φ_Q and Φ_I, and (*iv*) an *arbitrary* storage space *B*, it is not always possible to make *QWL* satisfiable via the refinement process. From this evidence, our idea of using a best-effort approach makes sense perfectly.
cube.

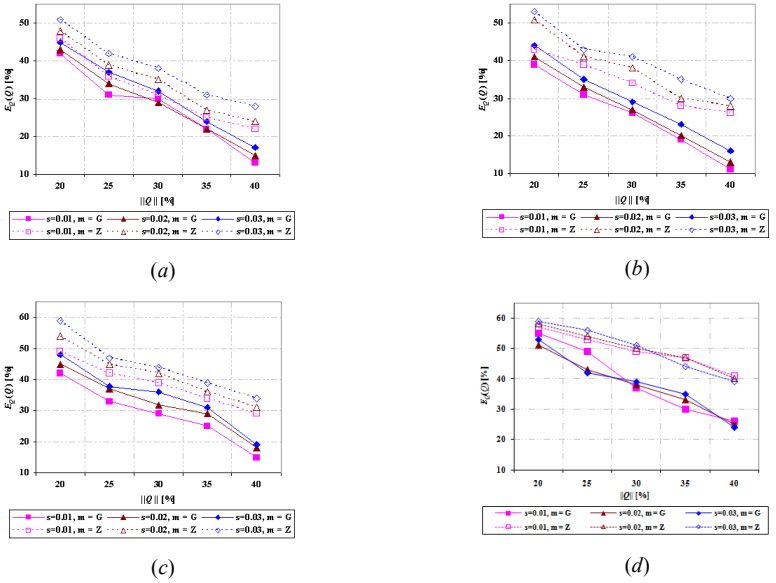

Fig. 1. Relative query errors of synopsis data cubes built from Uniform (*a*), skewed (*b*) TPC-H (*c*) and FCT (*d*) data cubes.

7 Experimental Evaluation

In order to test the effectiveness of our framework throughout studying the performance of algorithm `computeSynDataCube`, we conducted an experimental evaluation where we tested how the relative query error (similarly, the accuracy of answers) and the relative inference error (similarly, the privacy of answers) due to the evaluation of populations of randomly-generated queries, which model query-workloads of our framework, over the synopsis data cube range with respect to the volume of queries. The latter is a relevant parameter costing computational requirements of any query processing algorithm (also referred as *selectivity* – e.g., see [4]). We considered the zero-sum method [9] as the comparison technique, which is a state-of-the-art perturbation-based privacy preserving OLAP approach.

In our experimental assessment, we engineered three classes of two-dimensional data cubes: synthetic, benchmark and real-life data cubes. For all these data cubes, we limited the cardinalities of both dimensions to a threshold equal to 1,000, which represents a reliable value modeling significant OLAP applications (e.g., [4]). In addition to this, data cubes of our experimental framework expose different *sparseness coefficient s*, which measures the percentage number of non-null data cells with respect to the total number of data cells of a data cube. As widely-known since early experiences in OLAP research [1], the sparseness coefficient holds a critical impact on every data cube processing technique, thus including privacy preserving data cube computation as well.

In particular, synthetic data cubes store two kinds of data: Uniform data, and skewed data, being the latter obtained by means of a Zipf distribution. The benchmark data cube we considered has been built from the *TPC-H* data set [11], whereas the real-life one from the *Forest CoverType* (FCT) data set [6]. Both data sets are well-known in the Data Warehousing and OLAP research community. The final sparseness of the TPC-H and FCT data cube, respectively, has been easily *artificially* determined within the same OLAP data cube aggregation routine. The benefits deriving from using different kinds of data cubes are manifold, among which we recall: (*i*) the algorithm can be tested against *different* data distributions, thus stressing the reliability of the collection of techniques we propose (i.e., allocation, sampling, refinement), which, as described in Section 6, inspect the nature of input

data to compute the final synopsis data cube; (*ii*) parameters of data distributions characterizing the data cubes can be controlled easily, thus obtaining a reliable experimental evaluation. Selectivity of queries has been modeled in terms of a percentage value of the overall volume of synthetic data cubes, and, for each experiment, we considered queries with increasing-in-size selectivity, in order to stress our proposed techniques under the ranging of an increasing input.

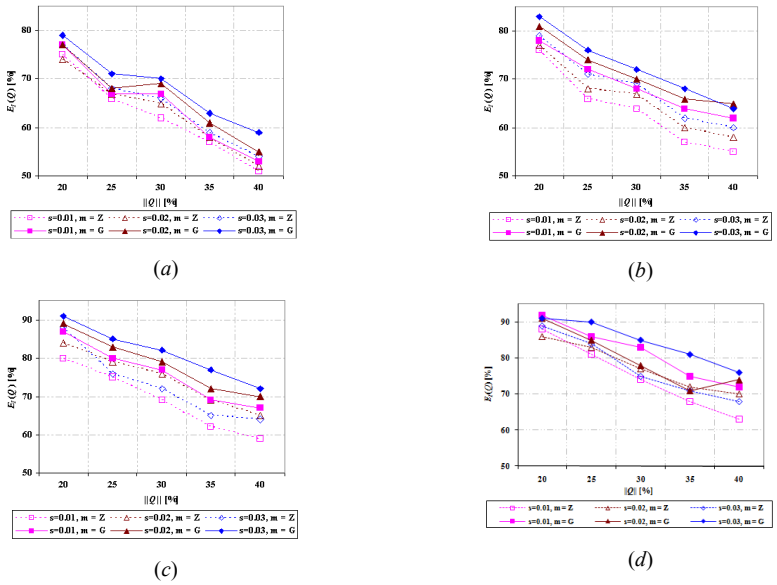

Fig. 2. Relative inference errors of synopsis data cubes built from Uniform (*a*), skewed (*b*) TPC-H (*c*) and FCT (*d*) data cubes.

For what regards compression issues, we imposed a *compression ratio r*, which measures the percentage occupancy of the synopsis data cube *A'*, *size*(*A'*), with respect to the occupancy of the input data cube *A*, *size*(*A*), equal to 20%, which is a widely-accepted threshold for data cube compression techniques (e.g., [4]). To simplify, we set the accuracy and privacy thresholds in such a way as not to trigger the refinement process. This also because [9] does not support any "dynamic" computational feature (e.g., tuning of the quality of the random data distortion technique), so that it would have been particularly difficult to compare the two techniques under completely-different experimental settings. On the other hand, this aspect puts in evidence the innovative characteristics of our privacy preserving OLAP technique with respect to [9], which is indeed a state-of-the-art proposal in perturbation-based privacy preserving OLAP techniques.

Figure 1 shows experimental results concerning relative query errors of synopsis data cubes built from Uniform, skewed, TPC-H, and FCT data, respectively, and for several values of *s*. Figure 2 shows instead the results concerning relative inference errors on the same data cubes. In both Figures, our approach is labeled as *G*, whereas [9] is labeled as *Z*. Obtained experimental results confirm the effectiveness of our algorithm, also in comparison with [9], according to the following considerations. First, relative query and inference errors decrease as selectivity of queries increases, i.e. the accuracy of answers increases and the privacy of answers decreases as selectivity of queries increases. This because the more are the data cells involved by a given query Q_k, the more are the samples extracted from $R(Q_k)$ able to "describe" the original data distribution of $R(Q_k)$ (this also depends on the proportional storage space allocation scheme (1)), so that accuracy increases. At the same time, more samples cause a decrease of privacy, since they provide *accurate* singleton aggregations and, as a consequence, the inference error decreases. Secondly, when *s* increases, we observe a higher query error (i.e., accuracy of answers decreases) and a higher inference error (i.e., privacy of answers

increases). In other words, data sparseness influences both accuracy and privacy of answers, with a negative effect in the first case (i.e., accuracy of answers) and a positive effect in the second case (i.e., privacy of answers). This because, similarly to results of [9], we observe that privacy preserving techniques, being essentially based on mathematical/statistical models and tools, *strongly* depend on the sparseness of data, since the latter, in turn, influences the *nature* and, above all, the *shape* of data distributions kept in databases and data cubes. Both these experimental evidences further corroborate our idea of trading-off accuracy and privacy of OLAP aggregations to compute the final synopsis data cube. Also, by comparing experimental results on Uniform, skewed, TPC-H, and FCT (input) data, we observe that our technique works better on Uniform data, as expected, while it decreases the performance on benchmark and real-life data gracefully. This is due to the fact that Uniform data distributions can be approximated better than skewed, benchmark, and real-life ones. On the other hand, experimental results reported in Figure 1 and Figure 2 confirm to us the effectiveness and, above all, the reliability of our technique even on benchmark and real-life data one can find in real-world application scenarios. Finally, Figure 1 and Figure 2 clearly state that our proposed privacy preserving OLAP technique outperforms the zero-sum method [9]. This achievement is another relevant contribution of our research.

8 Conclusions and Future Work

A complete framework for efficiently supporting privacy preserving OLAP aggregations on data cubes has been presented and experimentally assessed in this paper. We rigorously presented theoretical foundations, as well as intelligent techniques for processing data cubes and queries, and algorithms for computing the final synopsis data cube whose aggregations balance, according to a best-effort approach, accuracy and privacy of retrieved answers. An experimental evaluation conducted on several classes of data cubes has clearly demonstrated the benefits deriving from the privacy preserving OLAP technique we propose, also in comparison with a state-of-the-art proposal. Future work is mainly oriented towards extending the actual capabilities of our framework in order to encompass intelligent update management techniques (e.g., what happens when query-workload's characteristics change dynamically over time?), perhaps inspired by well-known principles of *self-tuning databases*.

References

[1] S. Agarwal et al., "On the Computation of Multidimensional Aggregates", *VLDB*, 506—521, 1996.

[2] R. Agrawal et al., "Privacy-Preserving OLAP", *ACM SIGMOD*, 251—262, 2005.

[3] A. Cuzzocrea, "Overcoming Limitations of Approximate Query Answering in OLAP", *IEEE IDEAS*, 200—209, 2005.

[4] A. Cuzzocrea, "Accuracy Control in Compressed Multidimensional Data Cubes for Quality of Answer-based OLAP Tools", *IEEE SSDBM*, 301—310, 2006.

[5] A. Cuzzocrea, "Improving Range-Sum Query Evaluation on Data Cubes via Polynomial Approximation", *Data & Knowledge Engineering*, 56(2), 85—121, 2006.

[6] UCI KDD Archive, *The Forest CoverType Data Set*, available at http://kdd.ics.uci.edu/databases/covertype/covertype.html

[7] A. Machanavajjhala et al., "*L*-diversity: Privacy beyond *k*-Anonymity", *ACM Trans. on Knowledge Discovery from Data*, 1(1), art. no. 3, 2007.

[8] A. Stuart et al., *Kendall's Advanced Theory of Statistics, Vol. 1: Distribution Theory*, 6th ed., Oxford University Press, New York City, NY, USA, 1998.

[9] S.Y. Sung et al., "Privacy Preservation for Data Cubes", *Knowledge and Information Systems*, 9(1), 38—61, 2006.

[10] L. Sweeney, "*k*-Anonymity: A Model for Protecting Privacy", *International Journal on Uncertainty Fuzziness and Knowledge-based Systems*, 10(5), 557—570, 2002.

[11] Transaction Processing Council, *TPC Benchmark H*, available at http://www.tpc.org/tpch/

[12] L. Wang et al., "Cardinality-based Inference Control in Data Cubes", *Journal of Computer Security*, 12(5), 655—692, 2004.