Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
The 24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2020)
September 16-18, 2020, Verona, Italy

SPPOLAP: Computing Privacy-Preserving OLAP Data Cubes Effectively and Efficiently

Algorithms, Complexity Analysis and Experimental Evaluation

Alfredo Cuzzocrea*

tDEA Lab University of Calabria, Rende, Italy
& LORIA, Nancy, France

Abstract

This paper provides significant contributions in the line of the so-called privacy-preserving OLAP research area, via extending the previous SPPOLAP’s results provided recently. SPPOLAP is a state-of-the-art algorithm whose main goal consists in computing privacy-preserving OLAP data cubes effectively and efficiently. The main innovations carried-out by SPPOLAP are represented by the novel privacy OLAP notion and the flexible adoption of sampling-based techniques in order to achieve the final privacy-preserving data cube. In line with the main SPPOLAP’s results, this paper significantly extends the previous research efforts by means of the following contributions: (i) complete algorithms of the whole SPPOLAP algorithmic framework; (ii) complexity analysis and results; (iii) comprehensive experimental analysis of SPPOLAP against real-life multidimensional data cubes, according to several experimental parameters. These contributions nice-fully complete the state-of-the-art SPPOLAP’s results.

© 2020 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the KES International.
Keywords: Big Data; OLAP; Privacy of Big Data; Privacy-Preserving OLAP.

1. Main text

Privacy-preserving big data management and analytics (e.g., [7,8,9,26,27,28]) is a novel and extremely-emergent research area with plenty of real-life applications spanning from sensor network analysis to smart city tools, from social networks analysis to bio-informatics and bio-medical systems, from epidemic analysis and spread prediction tools (like those related to the recent COVID-19 pandemic – e.g., [10,11,12]) to graph analytics systems, and so forth.

In all these application scenarios, the main goal consists in providing big data management and analytics services in a privacy-preserving manner, i.e. preserving the privacy of sensitive information (e.g., [13]), while still ensuring effectiveness and, moreover, efficiency. In this context, privacy-preserving OLAP (e.g., [14,15]) is a prominent research line, due to two main aspects: (i) OLAP tools are “premiere-class” big data analytics tools; (ii) structural nature of OLAP data cubes, i.e. array-based, lends itself “naturally” to elegant privacy-preserving solutions, as confirmed by recent, authoritative initiatives (e.g., [16,17]).

In this so-delineated context, SPPOLAP [1] is a state-of-the-art privacy-preserving OLAP algorithm that introduces the two following main innovations: (i) a novel privacy OLAP notion; (ii) flexible adoption of sampling-based techniques in order to achieve the final privacy-preserving data cube. As regards the first innovation, in fact, SPPOLAP overcomes classical privacy-preserving approaches (e.g., [16]) that adopt the vision of “seeing” OLAP data cubes as intrinsic multidimensional arrays, and try to compute the respective privacy-preserving multidimensional arrays. Contrary to this, SPPOLAP introduces the nice and elegant amenity of focusing on the privacy of OLAP aggregations, thus better exploiting the semantics of the powerful OLAP data cube model [18]. As regards the second innovation, too, SPPOLAP effectively and efficiently makes use of sampling-based techniques as follows. First, a suitable privacy grid is computed on top of the target OLAP data cube by considering a reference query workload on the cube. Then, an innovative privacy metrics that focuses on the privacy of OLAP aggregations is introduced, and the so-called privacy threshold is consequently defined, as capturing a lower bound for the deriving inference error. The latter measures how much OLAP queries issued to the privacy-preserving data cube are “wrong”, while still keeping an acceptable accuracy degree, so that they preserve the privacy of sensitive OLAP aggregations (i.e., limit the possibility of inferring further information from the knowledge already acquired). Finally, samples are extracted from the grid regions (determined by the reference privacy grid) as to satisfy the privacy threshold, according to a greedy strategy. By “possibly” satisfying the privacy threshold on all the grid regions, then the sampling-based privacy-preserving OLAP data cube is finally obtained.

It should be note that, as regards the emerging big data management and analytics research area, SPPOLAP can really provide relevant opportunities. In fact, while the need for powerful big data analytics tools is clear (e.g., [19,20]), at the same, the privacy-preserving requirements are, nowadays, indispensable aspects to be considered in every real-life system. Smart city application scenarios represent convincing case studies on this, where the privacy preservation axiom becomes a hard obstacle to smart citizen services for e-society, e-procurement, e-government, etc.

With respect to [1] where the main SPPOLAP’s results are presented, this paper significantly extends [1] by means of the following contributions:

- complete algorithms of the whole SPPOLAP algorithmic framework;
- complexity analysis results;
- comprehensive experimental analysis of SPPOLAP against real-life multidimensional data cubes, according to several experimental parameters.

The remaining part of this paper is organized as follows. In Section 2, we provide a summary of the main SPPOLAP approach. Section 3 provides the complexity analysis and results of the SPPOLAP algorithm. In Section 4, a comprehensive experimental analysis of SPPOLAP against real-life multidimensional data cubes, according to several experimental parameters is presented. Finally, Section 5 contains conclusions and future work of our research. Further, Appendix A reports all the algorithms of the complete SPPOLAP algorithmic framework.
2. SPPOLAP in a Nutshell

In this Section, we provide a summary of the main SPPOLAP approach [1]. Given an OLAP data cube $C$, the goal of SPPOLAP algorithm is computing a synopsis privacy-preserving data cube $C'$. Here, we first provide basic definitions and metrics introduced by SPPOLAP, then we describe each step of SPPOLAP. Appendix A reports all the algorithms of the whole SPPOLAP algorithmic framework.

2.1. Basic Definitions

A data cube $C$ is defined as a tuple $C = \langle \Delta, D, O, A \rangle$, such that: (i) $\Delta$ is the data domain of $C$, which contains data cells, i.e. the elementary aggregations of $\Delta$ computed against the relational data source $R$; (ii) $D$ is the set of dimensions of $C$, which are the functional attributes with respect to which tuples in $R$ are aggregated; (iii) $O$ is the set of hierarchical representations of the functional attributes in $D$, shaped in form of trees and known as hierarchies; (iv) $A$ is the set of measures of $C$, which are the set of attributes with respect to which SQL aggregations stored in data cells of $C$ are computed. Let $u \in D$ be a generic dimension of $C$; we define $|u|$ as the cardinality of dimension $u$ and $O(u)$ as the hierarchy of $u$.

Let $C$ be a $|D|$-dimensional data cube. Let $m$ be an integer such that $m \leq |D|$. We define an $m$-dimensional range query $G$ against $C$ as a tuple $G = \langle l_0, l_1, \ldots, l_{m-1}, \ominus \rangle$, such that (i) $l_i$ is a contiguous range defined on the dimension $u_i$ of $C$, with $0 \leq i \leq m-1$, and (ii) $\ominus$ is a SQL aggregation operator (i.e., SUM, COUNT, AVG). When applied to $C$, $G$ returns the $\ominus$-based aggregation over the data cells in $C$, contained in the multidimensional sub-domain of $C$ and bounded by ranges $l_0, l_1, \ldots, l_{m-1}$. In this work, we focus on range-SUM queries, as SUM aggregations are very popular in OLAP, and it also supports many different SQL aggregations, such as COUNT and AVG.

We define also the region of a query $G$, $H(G)$, as the sub-domain of $C$ bounded by ranges $l_0, l_1, \ldots, l_{m-1}$. Given a data domain $\Delta$, we define the volume of $\Delta$ as follows:

$$||\Delta|| = |u_0| \times |u_1| \times \ldots \times |u_{|D|-1}|$$  \hspace{1cm} (1)

The definition of volume can be extended also to a multidimensional data cube $C$, thus introducing the volume of $C$, denoted as $||C||$, and to a multidimensional range query $G$, thus introducing the volume of $G$, $||G||$, which in literature is also known as the selectivity of $G$.

Given a $|D|$-dimensional data cube $C$, its privacy grid is defined as a tuple $Z(C) = \langle r_0, r_1, \ldots, r_{|D|-1} \rangle$, such that $r_k$ is a range partitioning the dimension $u_k$, with $0 \leq k \leq |D|-1$ in a $r\ell$-based partition. By combining the partitions along all dimensions $u \in D$, we obtain $Z(C)$ as a regular partition of $H(C)$ (i.e., the multidimensional region associated to $C$) composed by the grid regions $H_{Z(C),k} = \{ r_{\ell_{0,b}} r_{\ell_{1,b}} \ldots \ r_{\ell_{|D|-1,b}} \}$. Starting from this definition, we can define $Z(C)$ as a collection of grid regions, namely, $Z(C) = \{ H_{Z(C),0}, H_{Z(C),1}, \ldots, H_{Z(C),|C|-1} \}$.

2.2. Accuracy and Privacy Metrics

We now describe metrics that we use for evaluating the accuracy of answers to queries and privacy of multidimensional aggregates. We start from the definition of relative query error between exact and approximate answers, which is a well-recognized quality measure in literature (e.g., [2,3]).

For a given query $G$, we denote as $C(G)$ the exact answer to $G$ (more precisely, the answer to $G$ evaluated on the data cube $C$) and as $C'(G)$ the approximate answer to $G$ (more precisely, the answer to $G$ evaluated on the synopsis data cube $C'$). The relative query error is then defined as follows:

$$E_G = \frac{|C(G) - C'(G)|}{C(G)}$$  \hspace{1cm} (2)

Since our goal is to preserve privacy, we design metrics that consider how sensitive information can be discovered from aggregate data. Starting from knowledge about a given query $G$ (i.e., volume and exact answer), it is possible to
infer knowledge about sensitive ranges of data contained in \( H(G) \). For instance, it is possible to derive the average value of the contribution throughout which each elementary data cell contributes to \( C(G) \). We name this quantity as **singleton aggregation of** \( G \), denoted by \( V(G) \). \( V(G) \) is defined as follows:

\[
V(G) = \frac{C(G)}{||G||}
\]  

(3)

Starting from \( V(G) \), it is possible to progressively discover aggregations of larger ranges of data within \( H(G) \). Indeed, by combining the knowledge about the synopsis data cube \( C' \) and the knowledge about the query \( G \), we can derive an estimation on \( V(G), V'(G) \), defined as follows:

\[
V'(G) = \frac{C'(G)}{\Sigma N(H)}
\]

(4)

such that \( \Sigma N(G) \) is the number of samples extracted from \( H(G) \) to compute \( C' \). We then introduce the **relative inference error** \( E_i(G) \), which gives us a metric of the privacy of \( C'(G) \), defined as follows:

\[
E_i(G) = \frac{|V(G) - V'(G)|}{V(G)}
\]

(5)

We also introduce the definition of **user-perceived singleton aggregation**, denoted by \( V'(G) \), which is the singleton aggregation perceived by external application based on the knowledge made available to them. \( V'(G) \) is defined as follows:

\[
V'(B) = \frac{C'(G)}{||G||}
\]

(6)

Then, we modify Equation (5) as to define the **relative user-perceived inference error**, as follows:

\[
E_i(B) = \frac{|V(G) - V'(B)|}{V(G)}
\]

(7)

2.3. **SPPOLAP Algorithm**

The input of **SPPOLAP** consists of: (i) the data cube \( C \); (ii) the space constraint \( B \); (iii) an integer parameter \( b \) (described in the following); (iv) the privacy threshold \( T \); (v) the typical query-workload on \( C \), defined as \( G_w \).

The first step of **SPPOLAP** consists in computing the privacy grid of the data cube \( C, Z(C) \). To compute \( Z(C) \), we need to calculate the range \( r_{ik} \) for each dimension \( u \in D \). We determine \( r_{ik} \) as an adequately-small fraction of the selectivity of queries in \( G_w \). Let \( S \) be the typical query selectivity of \( G_w \), computed by composing all the most frequent query ranges in \( G_w \), and \( ||H_{Z(C),k}|| \) the volume of \( H_{Z(C),k} \). If \( ||H_{Z(C),k}|| < S \), \( C' \) can be computed by using the grid region as the elementary reasoning unit, and adopting a resolution level lower than the resolution level of queries against \( C \). By doing so, it is possible to achieve an effective information gain during the computation of \( C' \). If sampling of the grid region \( H_{Z(C),k} \) is performed in a way that satisfies the privacy threshold \( T \) while ensuring the accuracy of approximate answers that involve region \( H_{Z(C),k} \), the same properties can also be inherited by the input queries on \( C \), since the latter queries are defined on top of grid regions.

The second step of **SPPOLAP** consists in obtaining the synopsis data cube \( C' \). This is done by using a greedy strategy guided by the space constraint \( B \). Due to the space constraint \( B \), we must compute a “best-effort” synopsis data cube \( C' \) such that: (i) it satisfies the privacy threshold \( T \); (ii) it ensures accuracy of approximate answers; (iii) it fits within \( B \). The greedy strategy works as follows: at each iteration \( j \), it selects a grid region, denoted by \( H_{Z(C),k} \), and extracts a set of samples, denoted by \( \Sigma(H_{Z(C),k}) \). The greedy selection is performed until space constraint \( B \) is consumed.

The greedy selection criterion considers the properties of **data distributions** associated to the grid regions in the
privacy grid \(Z(C)\), more precisely, the skewness of data distributions associated to different grid regions. Let \(F(H_{Z,j,k})\) be the data distribution associated to grid region \(H_{Z,j,k}\). The main idea behind the selection criterion is based on the assumption that, in order to describe a grid region whose distribution is skewed (i.e., values of \(F(H_{Z,j,k})\) are distributed according to a Zipf distribution, with asymmetric peaks), a number of samples higher than the number of samples required to describe a uniform grid region (i.e. a region where values of \(F(H_{Z,j,k})\) are distributed around their average value) is required. We can safely assume that \(F(H_{Z,j,k})\) are multidimensional, being the grid regions defined on top of multidimensional data cubes.

In order to determine if a given data distribution is skewed, we adopt the result described in [4]. According to [4], a data distribution \(F\) is considered skewed if the skewness value of \(F\), \(\text{skew}(F)\), is greater than its standard deviation, \(\sigma(\text{skew}(F))\), by a factor equal to 2.6, namely: \(\text{skew}(F) > 2.6 \times \sigma(\text{skew}(F))\). According to [5], \(\text{skew}(F)\) can be computed as follows:

\[
\text{skew}(F) = \left( \frac{\mu_3(F)}{\mu_2(F)} \right)^2
\]

\[\tag{8}\]

where \(\mu_i(F)\) denotes the \(i\)-th central moment. Based on the definitions in [4], we introduce the characteristic function \(\varphi(F)\), which maps to 1 if \(F\) is skewed and to 0 if \(F\) is uniform. \(\varphi(F)\) is defined as follows:

\[
\varphi(F) = \begin{cases} 
1 & \text{if } \frac{\sqrt{q} \left( \sum_{k=0}^{n-1} (k-\mu)^3 F(k) \right)^2}{\sqrt{\sum_{k=0}^{n-1} (k-\mu)^2 F(k)}} > 2.6 \\
0 & \text{otherwise}
\end{cases}
\]

\[\tag{9}\]

such that \(q\) and \(\mu\) are, respectively, the number of data items and the mean value of the distribution \(F\).

Once the region \(H_{Z,j,k}\) is selected according to the aforementioned greedy criterion, samples must be extracted from it in order to obtain the set of samples \(\Sigma(H_{Z,j,k})\). This operation is the baseline for computing the synopsis data cube \(C\). We employ a strategy called uniform sampling, i.e. based on the conventional Uniform distribution. Uniform sampling works as follows: let \([\Delta_{\text{min}}, \Delta_{\text{max}}]\), with \(\Delta_{\text{max}} > \Delta_{\text{min}}\), be the interval where the data domain \(\Delta\) is defined (we assume that \(\Delta\) is one-dimensional). The \(i\)-th sample is extracted by first sampling a random indexer \(n\) using a Uniform distribution defined in \([\Delta_{\text{min}}, \Delta_{\text{max}}]\) and then returning the value \(\Delta[n]\), if it has not been already selected in a previous iteration. This procedure can be easily generalized to a multidimensional data domain by iterating the same procedure over all dimensions \(u \in D\).

Given the grid region \(H_{Z,j,k}\) at the iteration \(j\) of SPPOLAP, we first consider the corresponding range-SUM query \(G_{Z,j,k}\) whose range is equal to the range of \(H_{Z,j,k}\). Then, we use the input parameter \(b\), with \(b > 0\), for iteratively sampling region \(H_{Z,j,k}\). More precisely, we sample \(b\)-sized sub-sets of samples from \(H_{Z,j,k}\) until (i) the privacy constraint on \(H_{Z,j,k}\) is satisfied (i.e., \(E\varphi(G_{Z,j,k}) \geq T\)) or (ii) \(B\) is consumed.

According to our design of SPPOLAP, \(b\) can be seen as a buffer size, used during the sampling phase. This solution allows avoiding computational overheads that would appear by performing sampling on massive in-size data cubes. Our sampling method considers the nature of OLAP queries and the privacy requirements. More precisely, we apply Uniform sampling on a subset \(U_{Z,j,k}\) of data cells in \(H_{Z,j,k}\) whose value is higher than the average value of \(H_{Z,j,k}\). This particular strategy allows us to obtain an approximate answer to \(G_{Z,j,k}\) having a good degree of approximation and, at the same time, a high degree of privacy, as it allows to satisfy the constraint \(E\varphi(G_{Z,j,k}) \geq T\). These properties are in turn inherited by input queries on \(C\), as above described.

3. SPPOLAP: Complexity Analysis and Results

In this Section, we provide the complexity analysis and results of the SPPOLAP algorithm.

First, consider that the SPPOLAP approach is inspired by the well-known research area of OLAP data cube compression algorithms (e.g., [25]). In this context, given an OLAP data cube \(C\) with \(|D|\) dimensions, the complexity of any arbitrary data cube compression algorithm \(P_c\) due to computing the compressed version of \(C\), said \(C_P\), denoted by \(O_{P_c}(n)\), is already known to being exponential in the number of dimensions \(|D|\), i.e.
O_{Pc}(n) = e^{\|D\|} \tag{10}

Obliviously, the exponential nature of the algorithm cannot be avoided, but suitable lower-costly computations of SPPOLAP can be achieved in most executions. The latter is another hidden goal of the SPPOLAP algorithm. Let \(|G_w|\) be the cardinality of the input query workload \(G_w\), and \(|Z(C)|\) the cardinality of the privacy grid \(Z(C)\) built on top of \(C\) (see Section 2). The complexity of SPPOLAP algorithm due to computing the privacy-preserving version of \(C\), \(C'\), denoted by \(O_{SPPOLAP}(n)\), is given by:

\[ O_{SPPOLAP}(n) = |G_w| \times e^{|Z(C)|} \tag{11} \]

Since, due to the SPPOLAP construction method (see Section 2), the following relation holds:

\[ |Z(C)| \ll |D| \tag{12} \]

then, the following relation holds:

\[ O_{SPPOLAP}(n) \ll O_{Pc}(n) \tag{13} \]

Hence, we can conclude that, given an OLAP data cube \(C\), the complexity of SPPOLAP algorithm due to computing the privacy-preserving version of \(C\), \(C'\), is, for most executions, significantly bounded by the exponential computational complexity with respect to the number of dimensions of \(C\), \(|D|\).

The latter is a significant theoretical result that is further corroborated by the experimental analysis provided in Section 4 (specifically, Section 4.3, where we experimentally assess the performance of SPPOLAP).

4. SPPOLAP: Experimental Assessment and Evaluation

In this Section, we provide an experimental assessment of SPPOLAP using real-life data cubes. We consider several perspectives of analysis, aiming at testing quality, effectiveness and sensitivity of our proposed technique. SPPOLAP performance is compared with Zero-Sum algorithm described in [6].

We focus on two-dimensional data cubes, which well covers the goals of a reliable experimental evaluation of privacy preservation capabilities. Indeed, we also tested our method on more probing multidimensional data cubes, obtaining results very similar to those experienced on two-dimensional data cubes. For this reason, here we present only the results on two-dimensional data cubes.

The parameters used in our evaluations are the following: (i) the cardinality of each dimension of the data cube, denoted by \(L_0\) and \(L_1\); (ii) the range sizes of grid regions of the privacy grid, \(K_0\) and \(K_1\); (iii) the sparseness coefficient \(s\), which measures the ratio of non-null data cells with respect to the total number of data cells; (iv) the space constraint \(B\); (v) the privacy threshold \(T\); (vi) \(C\), which models the type of data cube we are using for our evaluation; (vii) the query selectivity \(S\). In addition, we introduce ad-hoc metrics for each perspective of analysis that is object of our experimental assessment.

In the quality analysis, we employ the privacy and accuracy factors defined by Sung et al. [6], respectively \(F_P\) and \(F_A\). We define both of them in the following.

Let \(C\) and \(C'\) be, respectively, the input data cube and the synopsis data cube. Let \(\omega(k)\) be a data cube cell having \(k\) as multidimensional indexer, with \(\omega = \{C, C'\}\), the privacy factor \(F_P\) measures the average amount of distorted cells contained in blocks of \(C'\). In the Zero-Sum method, the block is a sub-cube with respect to which marginal sums of perturbed data cells along rows and columns are maintained equal to zero. \(F_P\) is defined as follows:

\[ F_P(C, C') = \frac{1}{|C|} \cdot \sum_{k=0}^{\left|C\right|-1} \frac{|C'(k) - C(k)|}{|C(k)|} \tag{14} \]

Basically, \(F_P\) is a way to measure how much good the privacy preservation of \(C'\) is. Since Zero-Sum is a method
oriented to data cells and our technique is based on the privacy OLAP notion, we will adapt the definition of \( F_P \) to better fit our problem. First, we replace the concept of block, underlying the definition of Equation (14), with the concept of grid region (indeed, they are very similar). Second, in case \( C'[k] = \text{NULL} \) (i.e., when \( C'[k] \) has not yet been sampled), we replace \( C'[k] \) with the corresponding singleton aggregation computed with respect to the grid region containing \( C'[k] \).

Concerning accuracy factor \( F_A \), it is instead defined in dependence of a query \( G \) on the synopsis data cube \( C' \), as follows:

\[
F_A(G) = 2^{-\frac{|C(G) - C'(G)|}{|C(G)|}}
\]  

(15)

such that \( C(G) \) is the exact answer to \( G \) and \( C'(G) \) is the approximate answer to \( G \). Basically, \( F_A \) is a measure of how much good the degree of approximation ensured by \( C' \) for a given query \( G \) is. We extend the definition of Equation (15) to an input query workload \( G_w \), as follows:

\[
F_A(G_w) = \frac{1}{|G_w|} \sum_{k=0}^{|G_w|} F_A(G_k)
\]  

(16)

For the purpose of the quality analysis, \( G_w \) is set as composed by the collection of range-SUM queries corresponding to blocks for the case of the Zero-Sum method, and to grid regions for SPPOLAP.

In the effectiveness analysis, we use the average relative user-perceived inference error, generalized from Equation (7) on the typical query workload \( G_w \), as follows:

\[
E_{P'}(G_w) = \frac{1}{|G_w|} \sum_{k=0}^{|G_w|} E_{P'}(G_k)
\]  

(17)

In our experimental framework, queries in \( G_w \) are synthetically generated as those queries that completely “span” the target data cube and having selectivity \( S \) fixed to a percentage of the volume of the data cube.

For our experimental evaluation, we select two-dimensional data cubes coming from three well-known datasets: APB, TPC-H and UsCensus. Fig. 1 shows the DFM [21] of the two-dimensional data cube built from APB. Fig. 2 and Fig 3 show the same for the case of TPC-H and UsCensus, respectively.

![Product класс](http://example.com/product_class)

**Fig. 1.** Two-Dimensional APB Data Cube used in our Experimental Evaluation.

![PART.P Name](http://example.com/part_p_name)

**Fig. 2.** Two-Dimensional TPC-H Data Cube used in our Experimental Evaluation.

![dAncestry.id](http://example.com/dancestry_id)

**Fig. 3.** Two-Dimensional UsCensus Data Cube used in our Experimental Evaluation.
4.1. Quality Analysis

In this Section, we describe the results of quality analysis of SPPOLAP on APB, TPC-H and UsCensus data cubes. Quality analysis is first performed with respect to the sparseness coefficient \( s \). The metrics that we use are the above-described accuracy and privacy factors \( F_a \) and \( F_b \), defined in Equation (14) and Equation (16), respectively.

Fig. 4 shows results of quality analysis with respect to the sparseness coefficient \( s \) against the target data cubes, with the following settings of experimental parameters: \( K_b = 10\% \), \( L_b = 10\% \), \( T = 70\% \), \( B = 20\% \), \( b = 20\% \), \( S = 10\% \). More precisely, Fig. 4 (a), Fig. 4 (b) and Fig. 4 (c) show the accuracy factor trend of SPPOLAP against the APB, TPC-H and UsCensus data cubes, respectively, in comparison with the results given by Zero-Sum, while Fig. 4 (d), Fig. 4 (e) and Fig. 4 (f) show the privacy factor trend of SPPOLAP against the APB, TPC-H and UsCensus data cubes, respectively, still in comparison with Zero-Sum.

Fig. 4. Quality Analysis on Two-Dimensional Data Cubes w.r.t. Sparseness Coefficient \( s \).

Fig. 5 shows results of quality analysis with respect to the space bound \( B \) against the target data cubes, with the following settings of experimental parameters: \( K_b = 10\% \), \( L_b = 10\% \), \( T = 70\% \), \( b = 20\% \), \( s = 20\% \), \( S = 10\% \). In more details, Fig. 5 (a), Fig. 5 (b) and Fig. 5 (c) show the accuracy factor trend of SPPOLAP against the APB, TPC-H and UsCensus data cubes, respectively, in comparison with the results given by Zero-Sum, while Fig. 5 (d), Fig. 5 (e) and Fig. 5 (f) show the privacy factor trend of SPPOLAP against the APB, TPC-H and UsCensus data cubes, respectively, still in comparison with Zero-Sum.
Fig. 5. Quality Analysis on Two-Dimensional Data Cubes w.r.t. Space Bound $B$.

Fig. 6 shows results of quality analysis with respect to the sampling percentage $sp$, which is defined as the number of sampled cells with respect to the total number of cells in the input data cube. In this analysis, the following setting of experimental parameters is used: $K_k = 10\%$, $L_k = 10\%$, $T = 70\%$, $B = 20\%$, $s = 20\%$, $S = 10\%$. In more details, Fig. 6 (a), Fig. 6 (b) and Fig. 6 (c) show the accuracy factor trend of SPPOLAP against the APB, TPC-H and UsCensus data cubes, respectively, while Fig. 6 (d), Fig. 6 (e) and Fig. 6 (f) show the privacy factor trend of SPPOLAP against the APB, TPC-H and UsCensus data cubes, respectively.

4.2. Effectiveness Analysis

In this Section, we describe the results of effectiveness analysis of SPPOLAP on APB, TPC-H and UsCensus data cubes. The metrics that we use is the average relative privacy error reported in Equation (17).
Fig. 6. Quality Analysis on Two-Dimensional Data Cubes w.r.t. Sampling Percentage $sp$.

Fig. 7 shows results of effectiveness analysis with respect to the selectivity $S$ against the target data cubes, with the following settings of experimental parameters: $K_k = 10\%$, $L_k = 10\%$, $T = 70\%$, $B = 20\%$, $b = 20\%$, $s = 20\%$. More precisely, Fig. 7 (a), Fig. 7 (b) and Fig. 7 (c) show the effectiveness trend of SPPOLAP against the APB, TPC-H and UsCensus data cubes, respectively, in comparison with the results given by Zero-Sum.

Fig. 8 shows results of effectiveness analysis with respect to the space bound $B$ against the target data cubes, with the following settings of experimental parameters: $K_k = 10\%$, $L_k = 10\%$, $T = 70\%$, $b = 20\%$, $s = 20\%$, $S = 10\%$. More precisely, Fig. 8 (a), Fig. 8 (b) and Fig. 8 (c) show the effectiveness trend of SPPOLAP against the APB, TPC-H and UsCensus data cubes, respectively, in comparison with the results given by Zero-Sum.

4.3. Performance Analysis

In this Section, we evaluate the performance of SPPOLAP in comparison with Zero-Sum approach. Performance is modeled in terms of sampling time in seconds. We evaluate performance with respect to buffer size $b$, due to the fact that, as described in Section 2.3, the use of a buffer size allows us to avoid excessive computation overhead that would be caused by performing sampling on massive in-size data cubes without buffering.

Fig. 7. Effectiveness Analysis on Two-Dimensional Data Cubes w.r.t. Query Selectivity $S$.
Fig. 8. Effectiveness Analysis on Two-Dimensional Data Cubes w.r.t. Space Bound B.

Results of the performance analysis are shown in Fig. 9. In this analysis, the following setting of experimental parameters is superimposed: $K_k = 10\%, \ L_k = 10\%, \ T = 70\%, \ B = 20\%, \ s = 20\%; \ S = 10\%$. Fig. 9 (a), Fig. 9 (b) and Fig. 9 (c) show the sampling time for APB, TPC-H and UsCensus data cubes, respectively, in comparison with the Zero-Sum approach.

5. Conclusions and Future Work

Starting from the extremely-emerging context of privacy-preserving big data management and analytics, this paper has further extended the research contributions initially provided by the state-of-the-art SPPOLAP algorithm [1] for supporting privacy-preserving OLAP. The novel contributions are focused on complete algorithms of the whole SPPOLAP algorithmic framework, complexity analysis and results, and comprehensive experimental analysis of SPPOLAP against real-life multidimensional data cubes, according to several experimental parameters. Particularly, the experimental results have confirmed the suitability of SPPOLAP in a wide range of emerging big data applications, such as smart city application scenarios, where the privacy preservation axiom becomes a hard obstacle to smart citizen services for e-society, e-procurement, e-government, etc. Future work is mainly oriented towards the extension of the actual framework to other big data management and analytics paradigms (e.g., [22,23,24]).

Fig. 9. Effectiveness Analysis on Two-Dimensional Data Cubes w.r.t. Buffer Size b

Acknowledgements

This research has been partially supported by the French PIA project “Lorraine Université d’Excellence”, reference ANR-15-IDEX-04-LUE. The author is grateful to Dr. Vincenzo De Maio for his contribution in the experimental part of early versions of this work.

References

[1] A. Cuzzocrea, V. Russo, D. Saccà, “A Robust Sampling-Based Framework for Privacy Preserving OLAP”, DaWaK, pp. 97-114, 2008
[2] A. Bonifati, and A. Cuzzocrea, “X(PPX: a Lightweight Framework for Privacy Preserving P2P XML Databases in Very Large Publish-Subscribe Systems”, ECWeb, pp. 21-34, 2007
[3] J. Domingo-Ferrer (ed.), “Inference Control in Statistical Databases”, From Theory to Practice, Springer, 2002
[4] A. Stuart, J.K. Ord, Kendall’s Advanced Theory of Statistics, Vol. 1: Distribution Theory, sixth ed., Oxford University Press, 1998
[5] A. Papoulis, Probability, Random Variables, and Stochastic Processes, second ed., McGraw-Hill, 1984
[6] S.Y. Sung, Y. Liu, H. Xiong, P.A. Ng, “Privacy Preservation for Data Cubes”, Knowledge and Information Systems, vol. 9, no. 1, pp. 38-61, 2006
[7] A. Cuzzocrea, “Privacy and Security of Big Data: Current Challenges and Future Research Perspectives”, PSBD, pp. 45-47, 2014
[8] R. Lu, H. Zhu, X. Liu, J. K. Liu, J. Shao, “Toward Efficient and Privacy-Preserving Computing in Big Data Era”, IEEE Network, vol. 28, no. 4, pp. 46-50, 2014
[9] A. Chakravorty, T. W. Wlodarczyk, C. Rong, “Privacy Preserving Data Analytics for Smart Homes”, IEEE Symposium on Security and Privacy Workshops, pp.23-27, 2013
[10] Y. Wang, M. Hu, Q. Li, X.-P. Zhang, G. Zhai, N. Yao, “Abnormal Respiratory Patterns Classifier May Contribute to Large-Scale Screening of People Infected with COVID-19 in an Accurate and Unobtrusive Manner”, CoRR abs/2002.05534, 2020
[11] O. Gozes, M. Frid-Aidar, H. Greenspan, P. D. Browning, H. Zhang, W. Ji, A. Bernheim, E. Siegel, “Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection & Patient Monitoring using Deep Learning CT Image Analysis”, CoRR abs/2003.05037, 2020
[12] L. Li, Z. Yang, Z. Dang, C. Meng, J. Huang, H. T. Meng, D. Wang, G. Chen, J. Zhang, H. Peng, “Propagation Analysis and Prediction of the COVID-19”, CoRR abs/2003.06846, 2020
[13] H. Zakerzadeh, C. C. Aggarwal, K. Barker, “Privacy-Preserving Big Data Publishing”, SSDBM, pp. 26:1-26:11, 2015
[14] A. Cuzzocrea, E. Bertino, “Privacy Preserving OLAP over Distributed XML Data: A Theoretically-Sound Secure-Multiparty-Computation Approach”, Journal of Computer and System Sciences, vol. 77, no. 6, pp. 965-987, 2011
[15] A. Cuzzocrea, V. Russo, “Privacy Preserving OLAP and OLAP Security”, Encyclopedia of Data Warehousing and Mining, Vol. 3, second ed., pp. 1575-1581, 2009
[16] R. Agrawal, R. Srikant, D. Thomas, “Privacy Preserving OLAP”, SIGMOD, pp. 251-262, 2005
[17] Y. Zhu, L. Huang, W. Yang, F. Dong, “Privacy Preserving Aggregate Query of OLAP for Accurate Answers”, Journal of Computers, vol. 5, no. 11, pp. 1678-1685, 2010
[18] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, H. Pirahesh, “Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub Totals”, Data Mining and Knowledge Discovery, vol. 1, no 1, pp. 29-53, 1997
[19] P. Russom, “Big Data Analytics”, TDWI Best Practices Report, fourth quarter, 2011
[20] A. Cuzzocrea, R. Moussa, G. Xu, “OLAP*: Effectively and Efficiently Supporting Parallel OLAP over Big Data”, MEDI, pp. 38-49, 2013
[21] M. Golfarelli, D. Maioro, S. Rizzi, “The Dimensional Fact Model: A Conceptual Model for Data Warehouses”, International Journal of Cooperative Information Systems, vol. 7, no (2-3), pp. 215-247, 1998
[22] A. Cuzzocrea, C. De Maio, G. Fenza, V. Loia, M. Parente, “OLAP Analysis of Multidimensional Tweet Streams for Supporting Advanced Analytics”, SAC, pp. 992-999, 2016
[23] A. Cuzzocrea, I.-Y. Song, “Big Graph Analytics: The State of the Art and Future Research Agenda”, DOLAP, pp. 99-101, 2014
[24] Y. Pei, J. Zhang, G. H. L. Fletcher, M. Pechenizkiy, “DyNMF: Role Analytics in Dynamic Social Networks”, IJCAI, pp. 3818-3824, 2018
[25] A. Cuzzocrea, “OLAP Data Cube Compression Techniques: A Ten-Year-Long History”, FGIT, pp. 751-754, 2010
[26] A. Cuzzocrea, C. Mastroianni, G.M. Grasso, “Private Databases on the Cloud: Models, Issues and Research Perspectives”, BigData, pp. 3656-3661, 2016
[27] R.C. Camara, A. Cuzzocrea, G.M. Grasso, C.K. Leung, S.B. Powell, J. Souza, B. Tang, “Fuzzy Logic-Based Data Analytics on Predicting the Effect of Hurricanes on the Stock Market”, FUZZ-IEEE, pp. 1-8, 2018
[28] P. Braun, A. Cuzzocrea, C.K. Leung, A.G.M. Pazdor, S.K. Tanbeer, G.M. Grasso, “An Innovative Framework for Supporting Frequent Pattern Mining Problems in IoT Environments”, JCCS, pp. 642-657, 2018