Integrating Semantics within Compressed OLAP Views in the Hand-OLAP System

Alfredo Cuzzocrea, Domenico Saccà, Paolo Serafino
ICAR-CNR and University of Calabria
I-87036 Cosenza, Italy
cuzzocrea@si.deis.unical.it, {sacca, pserafino}@deis.unical.it

Abstract. In this paper, we provide further extensions of Hand-OLAP, a Java-based distributed system for enabling OLAP in mobile environments via intelligent data cube compression approaches. These extensions aim at integrating innovative semantics representation and management models within compressed OLAP views, in order to improve the data cube compression process itself, and to support an improved summarized, OLAP-like knowledge fruition from multidimensional data cubes throughout mobile devices. We complete our analytical contribution by means of an experimental evaluation of the novel semantics-based data cube compression approach on well-known benchmark data cubes, which definitely confirms to us the efficiency and the reliability of our proposed research.

1 Introduction

Data Warehousing (DW) systems offer integrated and consolidated views over heterogeneous repositories of business data in the form of multidimensional data cubes [10] that store SQL-based aggregations (e.g., SUM, COUNT) made available to decision makers via meaningful OLAP [6] tools. OLAP tools effectively and efficiently support advanced decision making processes via intuitive abstractions that found on a multidimensional and multi-resolution vision of data. Indeed, the latter particular feature represents a critical add-in value in Business Intelligence (BI) scenarios thanks to nice and insightful conceptual constructs like measures and dimensions, i.e. the attributes of interest and the functional attributes for the target decision making process, respectively. Also, hierarchies and levels, which are associated to dimensions, allow decision makers to meaningfully explore and mine huge amounts of multidimensional data on the basis of different-in-granularity perspectives of analysis. To give an example, sales data could be mined according to the granularities Month or Quarter, in an alternative or combined manner as well, being Month and Quarter members of levels of the hierarchy Time. Finally, a wide family of powerful operators [6,10] (like drill-down, which allows us to increase the level of detail of multidimensional data while decreasing their level of abstraction, and roll-up, the opposite one) and query classes (e.g., range queries [12]) meaningfully complete the capabilities of OLAP tools in extracting summarized knowledge from huge amounts of multidimensional data, by unquestionably overcoming the performance of conventional OLTP tools based on traditional SQL interfaces.

Despite the nice and well-understood characteristics of OLAP tools outlined above, the so-called curse of dimensionality problem [14], i.e. the enormous size and degree of multidimensionality of typical real-life data cubes, causes a natural disorientation in decision makers that, contrary to this, are instead very often interested in exploring and mining specific multidimensional ranges of the target data cube, depending on particular business analysis goals, and more and more neglect to access and query a global view of the data cube. Inspired by this breaking evidence, several data cube compression techniques [3] have been proposed during the past two decades. These techniques aim at obtaining compressed representations of
multidimensional data cubes in order to retrieve fast approximate answers to time-consuming OLAP queries, and allow us to achieve faster and more efficient data cube computation and query evaluation methodologies at a cost of a degree of approximation that is reasonably regarded as tolerable for OLAP applications, where decimal precision is not required (e.g., [7]). Unfortunately, state-of-the-art data cube compression techniques are not enough to effectively deal with challenges posed by exploring, querying and mining very large and highly-dimensional data cubes, as they indeed deliberately neglect the semantics exposed by data cube models and schemas, and try to adopt an “algorithmic” vision of the intrinsic data cube compression problem according to which a data cube is just viewed as a “bulk of data cells”.

Described drawbacks get even worse when data cubes are processed, analyzed and mined in a mobile environment, where limitations of mobile devices [2], such as low computational capability, bounded memory capacity and limited visualization aptitudes, make traditional data cube compression approaches completely unsuitable to the mobile OLAP goal. On the other hand, mobile environments are more and more becoming leading application scenarios for BI, due to their clear aptitude and complete suitability for emerging computational paradigms and metaphors, like pervasive and ubiquitous information systems [17], information and knowledge dissemination approaches [20], location-aware and context-aware data warehouse management [19], and so forth.

Fig. 1. Hand-OLAP overview (a) and data cube compression approach (b)

With the aim of enabling OLAP in mobile environments, in [9] we assert that the data cube compression process must be driven by semantics kept in data cubes, thus making this process more effective, beyond that more efficient, and, above all, overcoming the natural disorientation of (mobile) decision makers during interactive OLAP exploration, querying and mining tasks over very large wireless-accessible data cubes. According to our vision, while this approach reasonably improves the performance of OLAP tasks over data cubes in conventional wired architectures, it plays an even more critical role in mobile environments, thus eliciting here the role of enabling database technology. Motivations and guidelines of OLAP research above have convinced us to author the system Hand-OLAP [8], a Java-based distributed system that relies on intelligent data cube compression techniques, and effectively and efficiently supports OLAP in mobile environments (see Figure 1 (a)). Following our previous research results [8], where we preliminarily introduce Hand-OLAP, and [9], where we firstly introduce the notion of semantics-based data cube compression as an alternative and more solid compression approach with respect to state-of-the-art ones, in this paper we present further research achievements due to Hand-OLAP, with particular emphasis on the innovative feature consisting in delivering semantics-aware compressed OLAP views in mobile environments, which is a novel and promising facet of next-generation BI mobile OLAP scenarios.
2 The Hand-OLAP Approach

Hand-OLAP is based on an innovative semantics-aware histogram-based data cube compression technique [9] (see Figure 1 (b)), which is finally codified in the so-called Hierarchy-driven Indexed Quad-Tree Summary (H-IQTS), a novel histogram for (mobile) MOLAP data cubes, i.e. data cubes represented in memory in the vest of multidimensional arrays. Histograms [3] are well-recognized and well-understood synopsis data structures for OLAP, which pursue the idea of obtaining compressed representations of data cubes by first generating a partitioned representation of the input data cube $D$, and then computing a bucket-based representation of the partition such that each bucket stores a SQL-based aggregation (e.g., SUM, COUNT) computed on top of data stored in the range it refers.

Traditional histograms are indeed not suitable to mobile environments [8]. Therefore, in order to deal with data cube compression challenges posed by these environments, Hand-OLAP introduces an innovative double-step approach according to which, in the first step, a two-dimensional OLAP view $V$ is extracted from the target multidimensional data cube $D$ by means of the so-called OLAP dimension flattening process. This process basically projects $D$ onto $V$ via (i) selecting two dimensions, called Visualization Dimensions (VD), from the whole collection of dimensions of $D$, and (ii) modifying the original hierarchies of the VD by meaningfully merging these hierarchies with hierarchies of the remaining dimensions of $D$. This allows us to represent a multidimensional domain $D$ equipped with hierarchies onto a two-dimensional domain $V$ whose hierarchies are composed by sub-trees of the $D$’s hierarchies, combined in a hierarchical fashion. Then, $V$ is re-aggregated according to the so-generated specialized hierarchies. Both the selection and the merge phase are driven by particular business analysis goals. It is easy to understand how, since this early stage, decision makers naturally select “isolated” and semantically-correlated two-dimensional ranges in $V$, targeted to their specific BI purposes.

In the second step of the Hand-OLAP data cube compression approach, the histogram H-IQTS is finally computed by taking as input the two-dimensional OLAP view $V$ and the bounded storage space $B$ available to house the H-IQTS in-memory representation. In more detail, H-IQTS is obtained via generating a quad-tree based partition of $V$ storing sub-domains that “follow” groups defined in the (specialized) OLAP hierarchies of $V$, i.e. according to the semantics these hierarchies expose. This allows us to finally originate in H-IQTS the so-called semantics-aware buckets, i.e. buckets computed over semantically-correlated two-dimensional ranges of $V$. Clearly, semantics-aware buckets are capable of supporting OLAP exploration, querying and mining tasks with higher effectiveness and efficiency than conventional ones one can find in traditional histograms, as traditional histogram-based data cube compression techniques are meant to generate the underlying partitioned representation of the input data cube $D$ in dependence on intensional properties of data stored in $D$, such as the variance of multidimensional ranges or the skeweness of final buckets [3]. Unfortunately, the latter general approach fails in capturing and supporting a typical BI subject-oriented analysis that is indeed archetypal in OLAP decision making [6], as it is well-understood and recognized that OLAP data are intrinsically clustered and correlated.

A specific characteristic of Hand-OLAP is represented by the ability of handling data cubes having both regular OLAP hierarchies, which are usually represented as conventional $n$-ary trees such that each node models a member of the hierarchy at a certain level, as well as more probing irregular OLAP hierarchies [16], whose most popular instance is the one exposing a lattice-based organization. In Hand-OLAP, we make use of a unifying approach according to which, given a lattice-based OLAP hierarchy $H^L$, $H^L$ is transformed in the corresponding tree-based OLAP hierarchy $H^T$ by means of the so-called hierarchy transformation process.
Basically, this process aims at finding an “equivalent” regular OLAP hierarchy starting from the irregular one via inspecting the equivalence of classes the OLAP members of respective hierarchies describe [8].

Finally, Hand-OLAP also fully supports interactive exploration and querying of the compressed two-dimensional OLAP view via high-performance query algorithms over the histogram H-IQTS. These algorithms are able of dealing with compressed two-dimensional data materialized in H-IQTS, and retrieving approximate answers to OLAP queries. In more detail, Hand-OLAP supports range queries [12] over the compressed OLAP view. Range queries apply an SQL aggregation operator (such as SUM, COUNT etc) to a sub-domain of the target view, and are widely recognized as a very popular way of extracting summarized knowledge from data cubes. It should be noted that these queries can be used as baseline operations to support meaningfully and intuitive mining tasks in OLAP.

In order to achieve high-efficiency during (range) query evaluation, leaf buckets of the quad-tree based partition codified in H-IQTS are equipped with Indices [5], which are high-performance synopsis data structures providing a succinct description of the actual distributions of buckets they are associated to. Indices can be stored in few bytes, with limited dependency on the effective bucket size, thanks to a meaningful summarization approach [5]. Basically, Indices allow us to significantly improve the so-called intra-bucket query estimation, which, as we prove in [5], could seriously decrease the approximate query answering capabilities of general data cube compression approaches like histograms and wavelets [3]. It should be noted that, since Indices occupy in memory a limited amount of storage space, they turn to be particularly suitable to the mobile OLAP context, where size of synopsis data structures plays a critical role.

3 The Hand-OLAP Architecture

Figure 2 (a) shows the logical architecture of Hand-OLAP, where we conceptually isolate each logical layer of the system. According to well-known multi-layer software design patterns, each (logical) component of the system corresponds to a specific application logic of the overall knowledge management process underlying the system itself, and it can be separately designed and developed yet codifying its (software) interface for component interaction purposes. The latter paradigm allows us to provide meaningful separation, maintenance and re-engineering functionalities. The logical layers of Hand-OLAP are the following:

- **Data Sources Layer**: it models the collection of (i) OLAP servers from which the desired range of multidimensional data can be retrieved, and (ii) wrappers that extract metadata on the available multidimensional databases and data cubes as well as the actual data.

- **Application Server Layer**: it models the layer that elaborates user’s requests, via (i) interacting with OLAP servers, (ii) extracting the two-dimensional OLAP view \( V \) from the target multidimensional data cube \( D \) by means of the OLAP dimension flattening process, (iii) computing the compressed representation of \( V \), and (iv) sending the compressed view to the mobile device.

- **User’s Layer**: it comprises the client-side tool running on the mobile device that allows a user to access and process the desired range of multidimensional data, by also enabling useful functionalities such as connectivity services, OLAP metadata browsing, two-dimensional OLAP view editing, compressed view browsing and querying etc.
Figure 2 (b) shows the Hand-OLAP software architecture, where the logical components depicted in Figure 2 (a) are detailed and described in their reference software platforms. In particular, the software architecture of Figure 2 (b) shows the prototype of Hand-OLAP when Microsoft SQL Analysis Services 2000 is adopted as OLAP server platform, and Compaq iPAQ as mobile device, respectively. All the other software components of the system are completely independent by both the OLAP server and the mobile device software platform, respectively, meaning that they can be interfaced with any other OLAP server platform (provided that the specific OLAP wrapper is designed) and any other mobile device software platform (provided that its Application Programming Interfaces (API) can be interfaced with Java). Server-side and client-side components are based on Java 2 Enterprise Edition (J2EE) and Java 2 Micro Edition (J2ME), respectively.

At the Data Sources Layer (see Figure 2 (a)), in order to handle the proper format of multidimensional data exposed by the OLAP server platform Analysis Services, we designed and developed a set of Win32 libraries based on the Microsoft ADOMD API that are able of (i) extracting the target two-dimensional OLAP view from Analysis Services via the multidimensional query language MultiDimensional eXpressions (MDX), (ii) writing the output OLAP view in a set of XML files (according to a specific partitioned in-memory-representation), and (iii) finally sending the XML-formatted OLAP view to the Java-based server-side components by means of serialization. Here, thanks to conventional Java/XML libraries (e.g., JDOM), the view is parsed and processed in order to extract the embedded multidimensional data. All considering, these libraries realize a specific OLAP wrapper for Analysis Services, and, above all, a “neutral” XML-based protocol that allows us to make the COM-compliant Analysis Services and the J2EE-compliant Application Server communicant. The same approach can be devised for any other OLAP server platform that cannot be interfaced with Java directly.

The Application Server Layer (see Figure 2 (a)), which is the most interesting component of Hand-OLAP with respect to the Data Engineering point of view, consists of five components that cooperate to fulfill user’s requests:

- **Request Manager**: it is the component that receives the request by the user, and translates it either into a request to the Metadata Manager for retrieving meta-information about the content of the target data cube, or into a request to the View Manager for retrieving a compressed representation of the two-dimensional OLAP view defined by the user.
- **Metadata Manager**: it is the component that extracts meta-information about the OLAP server it is connected to, and returns them in an XML format.
- **View Manager**: it is the component that (i) extracts from the selected data cube the two-dimensional OLAP view, (ii) invokes the Compression Agent to compress the view, and (iii) returns the compressed representation of the view to the mobile device;
- **Compression Agent**: it is the component that receives the two-dimensional OLAP view from the View Manager, and returns the compressed representation of the view.
• **Query Manager**: it is the component that is in charge of supporting range query evaluation on the compressed two-dimensional view, and visualizing the results on the mobile device according to a partitioned hierarchical representation.

The **User’s Layer** (see Figure 2 (a)) consists of the Hand-OLAP client-side tool, which supports several functionalities such as accessing OLAP metadata stored in the target OLAP server, editing the two-dimensional OLAP view of interest, browsing and querying the compressed view. In the following, we will consider a case study running on the OLAP server platform **Analysis Services**, and particularly focusing on the multidimensional data cube **Sales** of the multidimensional database **FoodMart 2000**, which is released in bundle with the OLAP server platform. **FoodMart 2000** stores information about an hypothetical company selling foods in a geographically-distributed area, and captures several typical sale entities like customers, employers, stores, warehouses, inventories, regions etc. **Sales** is a SUM-based 12-dimensional data cube storing 5 measures that fully supports OLAP analysis over sale data stored in **FoodMart 2000** via simultaneously aggregating them according to the following conceptual OLAP dimensions: **Time**, **Customer**, **Promotion**, **Product**, and **Store**, which well describe a typical sale scenario.

![Fig. 3. Accessing OLAP metadata (a), defining the two-dimensional OLAP view of interest (b), browsing (c) and querying (d) the compressed two-dimensional OLAP view](image)

Throughout Hand-OLAP, the user is allowed to connect the OLAP server and access OLAP metadata on the server (e.g., exploring the collection of multidimensional databases available in the server), as well as OLAP metadata on data cubes that are regarded as relevant for her/his decision making process (see Figure 3 (a)). Having selected a target multidimensional data cube, the user is allowed to define the two-dimensional OLAP view of interest by means of intuitive tools that hierarchically explore the available dimensions in order to progressively select specific semantically-correlated multidimensional ranges (see Figure 3 (b)), on the basis of particular business analysis goals. The definition of the view is thus sent to the Application Server that is in charge of compressing the view and re-sending it to the mobile device. Finally, the user is allowed to browse (see Figure 3 (c)) and query (see Figure 3 (d)) the compressed two-dimensional OLAP view in an off-line manner, yet being capable of refreshing the view when updates occur in the underlying data sources alimenting the data cube of interest.

### 4 Experimental Assessment and Analysis

In order to test the capabilities of H-IQTS, the core data structure of Hand-OLAP, in supporting effective and efficient compressed OLAP data fruition, we conducted some experiments on benchmark data cubes, which are more probing and reliable that synthetic ones with respect to
the assessment of any data-intensive OLAP processing algorithm/technique. In particular, we considered the well-known benchmark data cubes TPC-H [21] and APB-1 [15].

As regards the input of our experimental assessment, we considered Hierarchical Range Queries (HRQ) introduced by us in [9] as a significant extension of the work of Koudas et al. in [13]. In our implementation, a HRQ $Q_{HR}(W_H, P_H)$ is a full tree such that: (i) the depth of such tree is equal to $P_H$; (ii) each internal node $N_i$ has a fan-out degree equal to $W_H$; (iii) each node $N_i$ stores the definition of a (“traditional”) range-SUM query $Q_i$; (iv) for each node $N_i$ in $Q_{HR}(W_H, P_H)$, there not exists any sibling node $N_j$ of $N_i$ such that $Q_i \cap Q_j \neq \emptyset$.

In our experimental framework, given a HRQ $Q_{HR}(W_H, P_H)$ we measure the Average Accessed Bucket Number (AABN), which models the average number of buckets accessed during the evaluation of $Q_{HR}(W_H, P_H)$, and it is defined as follows:

$$AABN(Q_{HR}(W_H, P_H)) = \frac{\sum_{k=0}^{2^H} 1}{(W_H)^{P_H}} \cdot \sum_{l=0}^{Q_{HR}} AABN(N_l)$$

where, in turn, $AABN(N_l)$ is the average number of buckets accessed during the evaluation of the population of queries $Q_{S,l}$ of the node $N_l$ in $Q_{HR}(W_H, P_H)$, defined as follows:

$$AABN(N_l) = \frac{1}{|Q_{S,l}|} \cdot \sum_{k=0}^{Q_{S,l}-1} ABN(Q_k)$$

such that, for each query $Q_k$ in $Q_{S,l}$, $ABN(Q_k)$ is the number of buckets accessed during the evaluation of $Q_k$. It should be noted that the metrics AABN allows us to measure the capabilities of H-IQTS in effectively and efficiently supporting compressed OLAP data access and fruition based on semantics of data. In fact, intuitively enough, during an OLAP data fruition session, the less is the number of buckets accessed the less is the entropy of the overall knowledge extraction process while the information gain results to be maximized.

Figure 4 (a) and Figure 4 (b) show our experimental results on the target benchmark data cubes when considering as input random populations of HRQ having different values of depth $P_H$ ($W_H = 5$ and compression ratio $r = 10\%$). As comparison, we considered the following state-of-the-art histogram-based data cube compression techniques: MinSkew [1], GenHist [11], and STHoles [4]. From the analysis of the experimental results, it clearly follows that H-IQTS outperforms state-of-the-art histogram-based data cube compression techniques in effectively and efficiently supporting compressed OLAP data fruition, thanks to nice and meaningful semantics-based metaphors.

5 Conclusions and Future Research Efforts
Following some previous research results provided by us in [8,9], in this paper we have presented further research achievements due to Hand-OLAP, a Java-based distributed system for effectively and efficiently supporting OLAP in mobile environments, with particular emphasis on the innovative feature consisting in delivering semantics-aware compressed OLAP views in mobile environments, which is a novel and promising facet of next-generation BI mobile OLAP scenarios. These innovative Hand-OLAP features have been described in detail, also making use of some intuitive case studies. Another contribution of this paper is represented by the experimental assessment of the capabilities of H-IQTS, the core data structure of Hand-OLAP, in supporting effective and efficient compressed OLAP data fruition via nice and meaningful semantics-based metaphors. Future work is mainly oriented towards devising novel methodologies for efficiently handling data updates that can occur in the underlying data sources alimenting the data cube of interest.
Fig. 4. Experimental results on the benchmark data cubes TPC-H (a) and APB-1 (b)

References