A New OLAP Aggregation Based on the AHC Technique

Riadh Ben Messaoud
rbenmessaoud@eric.univ-lyon2.fr

Omar Boussaid
omar.boussaid@univ-lyon2.fr

Sabine Rabaséda
sabine.rabaseda@univ-lyon2.fr

Laboratoire ERIC, Université Lumière Lyon 2
5 avenue Pierre Mendès–France,
69676 Bron Cedex, France

ABSTRACT

Nowadays, decision support systems are evolving in order to handle complex data. Some recent works have shown the interest of combining on-line analysis processing (OLAP) and data mining. We think that coupling OLAP and data mining would provide excellent solutions to treat complex data. To do that, we propose an enhanced OLAP operator based on the agglomerative hierarchical clustering (AHC). The here proposed operator, called $OpAC$ (Operator for Aggregation by Clustering) is able to provide significant aggregates of facts refereed to complex objects. We complete this operator with a tool allowing the user to evaluate the best partition from the AHC results corresponding to the most interesting aggregates of facts.

Categories and Subject Descriptors: H.2.8 Data mining, Image databases: Database applications, I.5.2 Classifier design and evaluation, Pattern analysis: Design Methodology, I.5.3 Algorithms, Similarity measures: Clustering.

General Terms: Algorithms, Measurement.

Keywords: Complex data, Complex objects, On-line analysis processing, Data mining, Clustering, Semantic aggregation, AHC.

1. INTRODUCTION

The concept of data warehousing was introduced to provide a support for making better decisions on a huge amount of data. In fact, a data warehouse is an analysis oriented structure that stores a large collection of subject-oriented, integrated, time variant and non-volatile data [17][16]. On-line analytical processing (OLAP) is a key feature supported by most data warehousing systems. OLAP tools explore, summarize and navigate into multidimensional data views, commonly called data cubes [2]. A data cube is a multi-dimensional data representation where each dimension consists in a set of categorical descriptors organized within hierarchical structures.

Generally, classical aggregation in OLAP is assimilated to the process of consolidating data values into a single summarized value. Typically additive data are well suited to be aggregated by elementary operations (*Sum*, *Average*, *Max*, *Min* and *Count*) in a simple computation of measures. For example, traditionally a user wants to observe the sale levels in different cities in a particular period of the year. This decision query should use descriptors as criteria to identify the target facts and make computation only over measures.

In certain cases, a user needs to express aggregates richer than those created from elementary computation of additive measures. Suppose that he/she wants to resume information about data by gathering similar facts in the same group and separating dissimilar facts into different groups, it is necessary to consider a computation of both descriptors and measures. In this case, instead of only computing measures, we should rather take measure and descriptors into account to aggregate facts expressing similarities and dissimilarities.

With classic OLAP aggregation, we summarize measures by computing them. Since we wish to summarize facts, we should take into account both measures and descriptors in a data cube. Measures and descriptors define facts expressing objects and concepts taken from real world situations. It can represent the sales benefits of a market within variation in location and time. Since we try to aggregate the facts linked to sales benefits, we are dealing with composed sets of categorical and numerical information that can be considered as complex data. And so, facts to aggregate present relevant analogies with complex object like texts, images, sounds and videos. These facts need appropriate tools and new ways of aggregation since we wish to analyze them. The current OLAP tools are unable to aggregate or summarize information contained in a set of such complex objects. For this, we intend to create a new type of aggregation operators.

We think that data mining techniques in such a situation can provide techniques in order to be used as aggregation operators. On the one hand, supported by database systems, OLAP has a powerful ability in organizing views and structuring data adapted to analysis, but is restricted to a simple data navigation and exploration which weakens its analysis power. On the other hand, data mining is not very powerful for organizing data, but is known for its descriptive and predictive power which can discover knowledge from both simple and complex data. The general issue of coupling KDD (Knowledge and Data Discovery) with Database systems was already discussed and motivated by Imielinski and Mannila in [15]. The authors argue that data mining

DOLAP'04, November 12–13, 2004, Washington, DC, USA.
Copyright 2004 ACM 1-58113-977-2/04/0011 ...$5.00.

sets new challenges to database technology. Their combination will lead to a *second-generation* database system able to manage KDD applications just as classical ones manage business applications.

Therefore, we consider OLAP and data mining as two complementary fields. Their association would be a potential solution to reinforce the weakness of each one. Furthermore, data cube structure can provide a suitable context for applying data mining methods. More generally, the association of OLAP and data mining allows a more elaborated analysis task exceeding the simple exploration of a data cube. We believe that, in the next years, it will emerge a new generation of decision-support systems. Our idea is to take advantages as well from OLAP as from data mining techniques and to integrate them in the same analysis framework in order to analyze complex objects. In spite of the fact that both OLAP and data mining were considered for long as two separate fields, several recent works (see section 2) proved the ability of their association to provide an interesting quality of analysis.

We have used the AHC (Agglomerative Hierarchical Clustering) as an aggregation technique. The resulting operator, called *OpAC* (Operator for Aggregation by Clustering) which provides significant aggregates of facts expressing interesting knowledge about the analysis domain. Furthermore, we propose a tool for evaluating the quality of the aggregates generated by *OpAC*. This tool is able to help the analyst to decide about the best aggregates responding to his objectives.

The remaining of this paper is organized as follow. We expose in section 2 some works combining OLAP and data mining. In section 3, we present the objectives of the proposed operator. We develop, in section 4, a formalization for the *OpAC* operator. In section 5, we propose a new tool to evaluate the quality of the aggregates generated by *OpAC*. In section 6, we describe an enhanced implementation of our prototype and summarize its results through a case study of an image data cube. Finally, in section 7, we conclude our work and propose some future research directions.

2. RELATED WORK

The major difficulty of combining OLAP and data mining is that traditional data mining algorithms are mostly designed for two-dimension datasets organized in the *Attributes-Values* form [8]. Therefore multidimensional data are not suited for these algorithms. Nevertheless, a lot of previous works proved the possibility and the interest of coupling the two fields. We distinguish three major approaches for coupling OLAP and data mining.

The first one try to extend the query language of the decision support systems in order to achieve data mining tasks. The *DBMiner* system, proposed by Han [11], resume this approach where extended OLAP operators can perform some data mining methods including association, classification, prediction, clustering and sequencing over numerical data cubes. Han defines the *OLAP Mining* as the mechanism which integrates OLAP technology with data mining techniques in order to be performed in different portions and levels of abstraction of a data cube. He presents the *OLAM* (On-Line Analytical Mining) as the process of extracting knowledge from multidimensional databases. Han also expects that OLAM will be a natural addition to the OLAP technology in order to enhance the power of multidimensional data analysis. In [5], Chen et al. discover behavior patterns by mining association rules about customers from transactional e-commerce data. They extend OLAP functions and use a distributed OLAP server with a data mining infrastructure and the resulting association rules are represented in particular cubes (*Association Rule Cubes*). Goil and Choudhary think that a dimension hierarchies can be used to provide interesting information at multiple concept levels. Their approach summarizes information in a data cube, extend OLAP operators and mine association rules [9]. Some other works consist in integrating mining functions in the database system using SQL language. Chaudhuri, from *Microsoft Research*, argues that data mining promises a giant leap over OLAP [1]. He proposes a data mining system based on extending the SQL language to build data mining methods over relational databases. Chaudhuri et al. developed a client-server middleware that performs a decision tree classifier over *MS SQL Server 7.0* [3]. Meo et al. propose a model that enables a uniform description of the problem of discovering association rules. The model extends the SQL language to provide the *MINE RULE* operator [20].

The second approach consists in adapting multidimensional data inside or outside the database system and apply the classical data mining algorithms on the resulting dataset. We divide this approach into two possible strategies. The first one consists in taking advantages from multidimensional database management system (*MDBMS*) in order to help the construction of learning models. Laurent proposes a cooperation between *Oracle Express* and a fuzzy decision tree software (*Salammbô*) [18]. This cooperation allows transferring learning tasks, storage constraints and data handling to the *MDBMS*. The second one consists in transforming the multidimensional data and making them usable by data mining methods. For instance, Pinto et al. integrate multidimensional information in data sequences and apply on them the discovery of frequent patterns [25]. In order to implement a decision tree on multidimensional data, Goil and Choudhary flatten data cubes and extract contingency matrix for each dimension at each construction step of the tree [10]. Chen et al. propose to adopt OLAP as a pre-processing step in the knowledge discovery process [4]. In the same context, Maedche et al. combine databases with classical data mining systems by using OLAP engine as interface to treat telecommunication data [19]. In this interface, OLAP tools create a target data set to generate new hypotheses by applying a data mining algorithm.

The third approach is rather based on adapting data mining methods and applying them directly on multidimensional data. Palpanas thinks that adapting data mining algorithms is an interesting solution to provide elaborated analysis and precious knowledge [23]. Parsaye claims that decision-support applications must consider data mining within multiple dimensions [24]. He proposes a theoretical *OLAP Data Mining System* that integrates a multidimensional discovery engine in order to perform discovery along multiple dimensions. Sarawagi et al. propose to integrate a statistical module, based on multidimensional regression, (*Discovery-driven*), in OLAP servers. This module guides the user to detect relevant areas at various hierarchical levels from cubes [27]. In [26], Sarawagi proposes a new tool, *iDiff*, that detects both relevant areas in a data cube and the reasons of their presence. The same approach was adopted by Favero and Robin to generate quantitative analysis reports

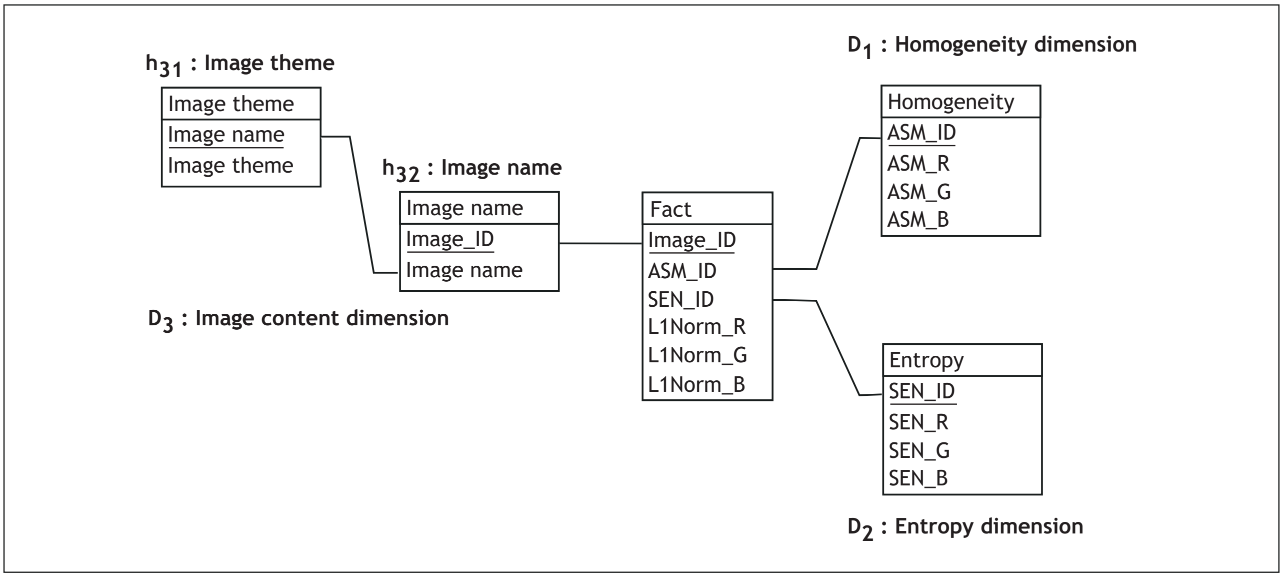

Figure 1: Snow-flake schema of an image data cube

from data cubes. They integrate in the *HYSSOP* system a content determination component based on data mining methods [7]. Imielinski et al. propose a generalized version of association rules called *Cubegrades* [14]. The authors claim that association rules can be viewed as the change of an aggregate due to a change in the cube's structure. They also introduce a language *CGQL* for expressing queries on *Cubegrades* and define a scheme to evaluate them. Dong et al. enhanced the *Cubegrades* and introduce the constrained gradient analysis [6]. It focuses on extracting pairs of cube cells that are quite different in aggregates and similar in dimensions. Instead of dealing with the whole cube, three constraints (*Significance constraint*, *Probe constraint* and *Gradient constraint*) are added to limit the search range.

Remind that our purpose is to use data mining to treat complex data in a multidimensional context. Some recent works have used data mining for modeling complex objects into database systems. The authors of the PANDA project [13] propose to extract knowledge from huge volumes and different sources of data. Through data mining paradigm this knowledge is modeled and represented as *patterns*. We can consider these *patterns* as complex objects. Our approach is rather dedicated to couple data mining and OLAP in order to create new on-line analysis techniques for complex data.

We finally note that the previous presented works proved that associating data mining to OLAP is a promising way to involve rich analysis tasks. They resume that data mining is able to extend the analysis power of OLAP tools. In addition to these works, we propose a new contribution by creating a new operator *OpAC* based on the association between OLAP and data mining. Our operator does not only use this association to enhance the analysis power of OLAP, but also creates significant aggregates of facts linked to complex data.

3. THE OBJECTIVES OF THE PROPOSED OPERATOR

The construction of a data cube targets precise analysis goals. The selection of its dimensions and measures depends on the analysis needs. Usually, a dimension is organized according to several hierarchies defining various levels of data granularity. Each hierarchy contains a set of modalities (also called *members*), and each modality of a hierarchy includes modalities from the hierarchy immediately below according to the logical membership order. Therefore, by moving from a hierarchical level to a higher one, modalities are gathered together into groups. In consequence, measures related to the modalities are computed and so information is resumed to a smaller sets number. It is how aggregation is made.

Let's consider images as an example of complex data modeled according to a data cube. The data cube describes RGB color channel (Red, Green and Blue) characteristic of images. We resume the cube's example by the snow-flake schema presented in Fig.1. It is made up of three dimensions:

- **The homogeneity dimension** describes the homogeneity levels (*very high, high, medium, low, very low*) of an image for the three color channel based on the Angular Second Moment presented in [12] (ASM_R, ASM_G, ASM_B);
- **The entropy dimension** which defines the disorder levels of colors in images (*very high, high, medium, low, very low*) for the three color channel based on the sum of entropy presented in [12] (SEN_R, SEN_G, SEN_B);
- **The image content dimension** defines the semantic subject of the image. This dimension is organized into two hierarchical levels: the *image name* and the *image theme*. The *image name* defines the precise content of an image. For example, an *image name* can have *"New York city"*, *"Tulip garden in spring"* or *"Everest*

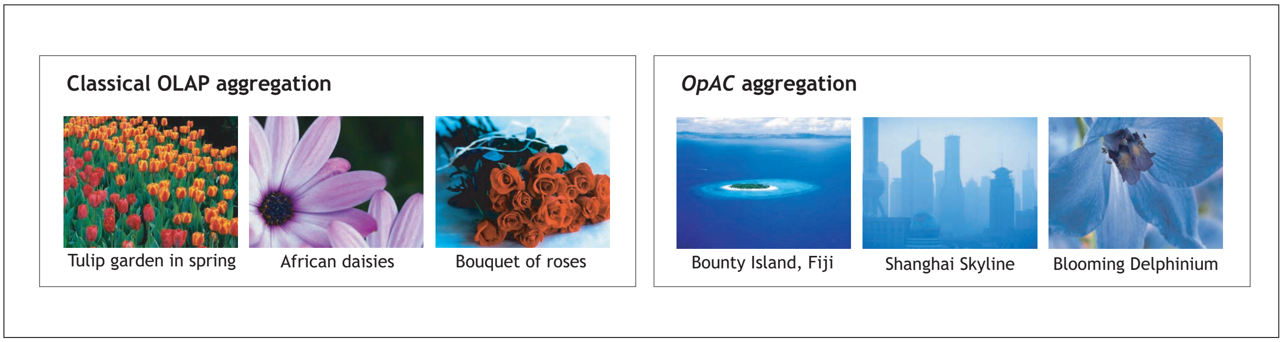

Figure 2: Example of $OpAC$ aggregation for complex data

mountain" as values. Whereas, an image theme can have *"Towns"*, *"Flowers"* or *"Nature"* as values;

- We define three measures for our cube based on the medium color characteristic of images (L1Normalised) for each color channel presented in [12]. We note these measure as follows:
  - M_1: L1Normalised$_R$;
  - M_2: L1Normalised$_G$;
  - M_3: L1Normalised$_B$.

In the OLAP context, it is well known that such hierarchical organization of a dimension induces sets of modalities' aggregates according to the logical order of membership. In the case of the image content dimension, *"Tulip garden in spring"*, *"African daisies"* and *"Bouquet of roses"* constitute an aggregate in the image theme level since they all represent images of *"flowers"* (see Fig.2). This classical aggregation is fully established in the conceptual step of the data cube. Therefore, such aggregation does not always induce significant relations to a miner, in instance an image structure analyzer.

The main idea of our operator $OpAC$ is to exploit the cube's facts describing complex data (particularly images in our case) in order to provide a more significant aggregation over these complex objects. To do that, we use a clustering method and automatically highlight aggregates semantically richer than those provided by the current OLAP operators. The new operator enables us to note, for instance, that the images *"Bounty Island, Fiji"*, *"Shanghai Skyline"* and *"Blooming Delphinium"* (see Fig.2) constitute a significant aggregate since they represent similar objects. In fact, these images, for a low levels of the Blue channel entropy, have slightly similar values of medium color characteristic in the same channel (L1Normalised$_B$). This aggregation is different from the classic one where images are gathered by theme and so based on subjective elements provided by human. $OpAC$ aggregation is rather based on objective descriptors and measures directly extracted from the complex objects. The aggregation results are submitted to the user who will decide to admit them or not. Therefore, the clustering method provides a new OLAP aggregation concept. This aggregation can easily be compared to the classic one since it provides hierarchical groups of objects resuming information and offer at the same time the possibility of navigating into the different levels of these groups. Furthermore, like the classic OLAP, the aggregates are computed from precomputed ones created from the original data cube.

Existing OLAP tools, like the *Slicing* operator, can also create new restricted aggregates in a cube dimension. Therefore, these tools always need a handmade user assistance, whereas our operator is based on a clustering algorithm that provides automatically relevant aggregates. Furthermore, with classical OLAP tools, aggregates are created in an intuitive way in order to compare some measure values, whereas $OpAC$ creates significant aggregates expressing deep relations with the cube's measures. Thus, the construction of this kind of aggregates is very interesting to establish a richer on-line analysis context.

According to the above objectives, we chose the agglomerative hierarchical clustering (AHC) as an aggregation method. We motivate this choice by the fact that the hierarchical aspect constitutes a relevant analogy between the AHC results and a hierarchical structure of a dimension. The objectives and the results expected for $OpAC$ match perfectly with the AHC strategy. Furthermore, the AHC adopts an agglomerative strategy that begins by the finest partition where each individual is considered as a cluster. This allows to include the finest modalities of a dimension level in the results of $OpAC$. Moreover, the results of the AHC are compatible with the exploratory aspect of OLAP and can be reused by its classical operators. The AHC provides several hierarchical partitions. By moving from a partition level to the higher one, two aggregates are joined together. And conversely, by moving from a partition level to the lower one, an aggregate is divided into two new ones. These operations are strongly similar to the classical operators *Roll-up* and *Drill-down*. The AHC is a well suited clustering method to resume information and create OLAP aggregates from a complex facts contained in data cube.

4. THE FORMALIZATION OF OPAC

This formalization aims to define individuals and variables domains for the clustering problem of our operator. Note that these domains are extracted from multidimensional environment. Thus, we should respect some constraints to ensure the statistical and logical validity of extracted data. Let's consider Ω the set of individuals and Σ the set of variables. We suppose that:

- $\mathcal{C}$ is a data cube having d dimensions and m measures;
- $D_1, \ldots, D_i, \ldots, D_d$ the dimensions of $\mathcal{C}$;

$$\begin{aligned}&\mathcal{G} = \prod_{i=1}^{d}(\underbrace{\mathcal{G}(h_{ij})}_{j\in\{1,\ldots,n_i\}} \cup \{*\}) \\ &\mathcal{G} = (\underbrace{\mathcal{G}(h_{1j})}_{j\in\{1,\ldots,n_1\}} \cup \{*\}) \times \ldots \times (\underbrace{\mathcal{G}(h_{ij})}_{j\in\{1,\ldots,n_i\}} \cup \{*\}) \times \ldots \times (\underbrace{\mathcal{G}(h_{dj})}_{j\in\{1,\ldots,n_d\}} \cup \{*\})\end{aligned} \tag{1}$$

$$\Sigma \subset \left\{ \begin{array}{l} X \ / \ \forall t \in \{1,\ldots,l_{ij}\} \\ \underbrace{X(g_{ijt})}_{j\in\{1,\ldots,n_j\}} = M_q(\, * \, , \ldots , \, * \, , \underbrace{g_{ijt}}_{j\in\{1,\ldots,n_j\}} , \, * \, , \ldots , \, * \, , \underbrace{g_{srv}}_{r\in\{1,\ldots,n_s\}} , \, * \, , \ldots , * \,) \\ \text{with } s \neq i, \ r \text{ is unique for each } s, \ v \in \{1,\ldots,l_{sr}\} \text{ and } q \in \{1,\ldots,m\} \end{array} \right\} \tag{2}$$

- $M_1, \ldots, M_q, \ldots, M_m$ the measures of $\mathcal{C}$;
- $\forall i \in \{1,\ldots,d\}$ the dimension D_i contains n_i hierarchical levels. For instance, the image content dimension (D_3) of the cube presented in Fig.1 is composed of two hierarchical levels. So $n_3 = 2$;
- h_{ij} the j^{th} hierarchical level of D_i, where $j \in \{1,\ldots,n_i\}$;
- $\forall j \in \{1,\ldots,n_i\}$ the hierarchical level h_{ij} contains l_{ij} modalities (or members). In our cube example, the level h_{21} of the entropy dimension contains five modalities. So $l_{21} = 5$;
- g_{ijt} the t^{th} modality of h_{ij}, where $t \in \{1,\ldots,l_{ij}\}$;
- $\mathcal{G}(h_{ij})$ the set of modalities of h_{ij}.

Let's suppose that we seek to classify modalities from the level h_{ij}. In the formalization presented in [21], the choice of the user was limited to the dimension D_i and its hierarchical level h_{ij}, and so we defined all its modalities $\mathcal{G}(h_{ij})$ as the set of individuals. We enhanced this formalization by providing the user an additional possibility of choosing modalities into h_{ij}. Therefore, we define the new set of individuals as follow:

$$\Omega \subset \mathcal{G}(h_{ij}) = \{g_{ij1}, \ldots, g_{ijt}, \ldots, g_{ijl_{ij}}\} \tag{3}$$

Let's now adopt the following notations:

- $*$ a meta-symbol indicating the total aggregate of a dimension;
- $\forall q \in \{1,\ldots,m\}$, we define the measure M_q as the function: $M_q : \mathcal{G} \longrightarrow \Re$;

Where $\mathcal{G}$ is the set of d-tuples of all the hierarchical level's modalities of the cube $\mathcal{C}$ including the total aggregates of dimensions (see Formula (1)).

Reconsider again the cube of Fig.1. In this case:

- $M_3(\mathit{high}_B,\ \mathit{medium}_G,\ *\ )$ indicates the value of blue medium color characteristic (L1Normalised$_B$) for all images having a high homogeneity in the blue channel and a medium entropy in the green one;
- $M_1(\mathit{low}_R,\ *\ ,\ \mathit{flowers}\ )$ indicates the value of red medium color characteristic (L1Normalised$_R$) for images of flowers having a low homogeneity in the red channel.

Remind that the objective of the *OpAC* operator is to establish a semantic aggregation by using a clustering technique on real facts contained in a data cube. We adopt so the cube measures as quantitative variables describing the population Ω. Nevertheless, according to [21], it is necessary to satisfy two fundamental constraints in the variables choice:

- **First constraint**. Hierarchical levels belonging to the dimension D_i retained for the individuals can't generate variables. In fact, describing an individual by a property which contains it has no logical sense. Conversely, a variable which specifies a property of an individual would only describe this one;
- **Second constraint**. In a dimension, only one hierarchical level can be chosen to generate variables. This constraint insures the independence of variables. In fact, the value taken by a modality from a hierarchical level can be obtained by linear combination of modalities' values belonging to the lower level.

Since Ω is chosen, we formulate the possible extracted set of variables Σ (see Formula (2)).

In [21], we gave the possibility to choose the dimensions D_s, the hierarchical levels h_{sr} and the measures M_q, then we consider all the modalities of h_{sr} to extract variables. We enhanced this process by providing a possibility of selecting precise modalities g_{srv} in h_{sr}. This extention allows to realize a targeted analysis tasks. Of course, the selection of g_{srv} depends on the objectives carried out by the user's analysis.

5. THE AGGREGATES EVALUATION TOOL

Recall that we propose to use the AHC as an aggregation operator over the modalities of a cube dimension. For n individuals to classify, the AHC generates n hierarchical partitions. The main weakness of the method is that it doesn't give any help about the best partition to choose. It is often the data miner who decides about the number of clusters that corresponds both to the context and to the goal of his analysis. However, the greater n is, the more difficult the choice of the best partition is. To help choosing the best partition generated by *OpAC*, we propose to evaluate the quality of each partition. The quality is measured according to the following criteria:

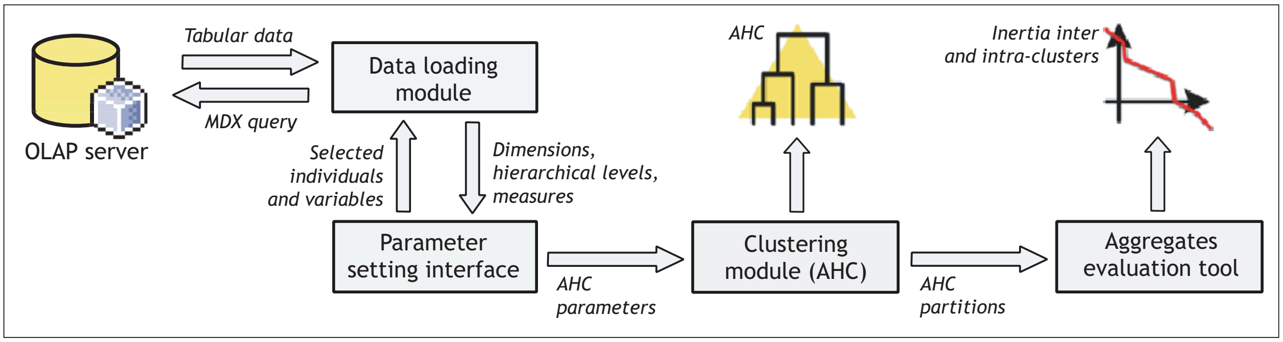
Tabular data
Data loading module
MDX query
OLAP server
Selected individuals and variables
Dimensions, hierarchical levels, measures
Parameter setting interface
AHC parameters
Clustering module (AHC)
AHC partitions
Aggregates evaluation tool
AHC
Inertia inter and intra-clusters

Figure 3: The architecture of the implementation

- Minimizing the intra-cluster distances, i.e. the distance between individuals within a cluster;
- Maximizing the inter-cluster distances, i.e. the distance between the gravity's centers of the clusters.

This lead us to define measures based on the inertia inter and intra-cluster. Let's formulate these two criteria:

- Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ be the set of individuals to classify;
- Each individual takes the weight $P(\omega)$ and is described by p numerical variables $X_1, X_2, \ldots, X_p$. Currently, in the case of our operator, we adopt a uniform distribution of weights. We attribute to each modality the weight $P(\omega) = 1 \; \forall \omega \in \Omega$;
- Let's suppose that, at the $(n-k)^{th}$ iteration of the AHC, $A_1, A_2, \ldots, A_k$ represents the current partition of Ω;
- $\forall i \in \{1, \ldots, k\}$ the subset A_i takes the weight $P(A_i) = \sum_{\omega \in A_i} P(\omega)$;
- $G(A_i) = \frac{1}{P(A_i)} \sum_{\omega \in A_i} P(\omega) X(\omega)$ represents the gravity center of A_i;
- $G = \sum_{\omega \in \Omega} P(\omega) X(\omega)$ represents the gravity center of Ω.

Let's consider d as a distance measure. For a given subset of individuals A_i, the inertia intra-cluster is defined as follow:

$$I(A_i) = \sum_{\omega \in A_i} P(\omega) d(X(\omega), G(A_i)) \qquad (4)$$

The total inertia intra-clusters of the partition is defined by the sum of its k subsets's inertia.

$$I_{intra}(k) = \sum_{i=1}^{k} I(A_i) \qquad (5)$$

The inertia inter-clusters is defined by the weighted sum of distances between the gravity's center of Ω and the gravity's centers of all the subsets A_i of the current partition.

$$I_{inter}(k) = \sum_{i=1}^{k} P(A_i) d(G(A_i), G) \qquad (6)$$

According to the theorem of *Huygens*, we prove that for each partition, the sum of the two inertia is constant and equal to the inertia of Ω.

$$\forall k \in \{1, \ldots, n\}, \; I_{intra}(k) + I_{inter}(k) = I(\Omega) \qquad (7)$$

The inertia intra-clusters is a decreasing function and the inertia inter-clusters is an increasing one. While moving from a partition to another, a remarkable change of the inertia intra or inter-clusters will be a relevant indicator in the choice of the number k of aggregates. Through this tool, we help the user to realize a better compromise between the minimization of the inertia intra-clusters, the maximization of the inertia inter-clusters, the number of aggregates, the significance of the aggregates and the analysis' objectives.

6. THE IMPLEMENTATION

6.1 The architecture of the implementation

To validate our new operator *OpAC*, we extended our first implementation[1] of *OpAC* presented in [21]. It is composed of three components: a *Parameter setting interface*, a *Data loading module* from *MS SQL Server 2000/Analysis Services* and a *Clustering module*. We propose now a new version of the prototype. We have enhanced some functions in the existing components and added an *Aggregates evaluation tool* to measure the quality of the partitions generated by the operator (see Fig.3). We propose to redefine the functions of each component in the following points:

- **The data loading module** ensures three tasks: it connects the prototype to a data cube via the OLAP server; uses *MDX queries* (*Multidimensional Expressions*) to import information about the cube's structure (labels of dimensions, hierarchical levels and measures) and to extract the data selected by the user;
- **The parameter setting interface** assists the user to extract both individuals and variables from a data cube. It allows the user to navigate into the hierarchical levels of dimensions and to select the modalities

[1] http://bdd.univ-lyon2.fr/download/opac.zip

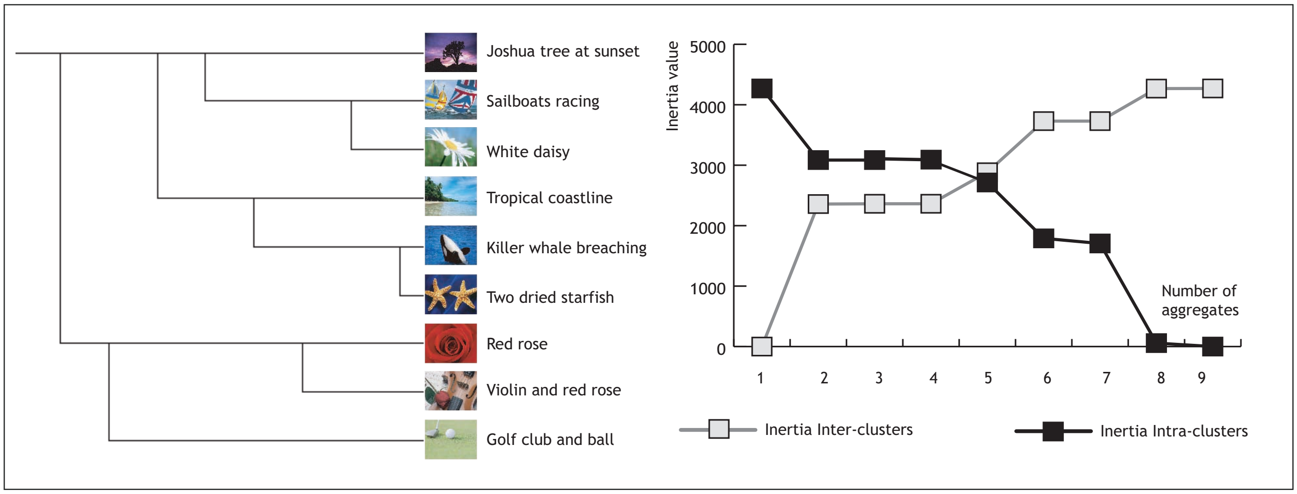

Figure 4: Results generated by *OpAC*

g_{ijt} for individuals, the modalities g_{srv} and the measures M_q for the variables of the clustering problem. It also provides a user assistance respecting constraints we defined in the theoretical formalization;

- **The clustering module** allows the definition of the clustering task by choosing the dissimilarity metric and the aggregation criterion, constructs the AHC model and plots the result within a dendrogram. The dendrogram includes a summary of the AHC's parameters and the analyzed data;
- **The aggregates evaluation tool** computes for each partition the inertia inter and intra-clusters and plots the results in a graph. As we exposed in the previous section, this tool permits to give an idea about the quality of the partitions of the AHC. It helps the user to decide about the best number of aggregates he wants to take.

6.2 A case study

To illustrate the results of our prototype, we propose a case study based on the cube presented in Fig.1. Let's suppose that a user seeks to create aggregates from the modalities of the level *Image name* (h_{32}) of the *Image content* dimension (D_3). He selects from $\mathcal{G}(h_{32})$ the following set of cities as individuals:

$$\Omega = \left\{ \begin{array}{c} \textit{White daisy, Red rose, Two dried starfish,} \\ \textit{Joshua tree at sunset, Tropical coastline,} \\ \textit{Violin and red rose, Killer whale breaching,} \\ \textit{Sailboats racing, Golf club and ball} \end{array} \right\} \quad (8)$$

Suppose that the user chooses the modalities *very high*$_B$ and *low*$_R$ from the *homogeneity* dimension (D_1) and the measure *L1Norm*$_B$ (M_3) to generate variables. According to (2), the set of variables is:

$$\Sigma = \left\{ \begin{array}{l} X_1 = M_3(\ \textit{very high}_B\ ,\ *\ ,\ \omega) \\ X_2 = M_3(\ \textit{low}_R\ ,\ *\ ,\ \omega) \\ \text{where } \omega \in \Omega \end{array} \right\} \quad (9)$$

Now, suppose that the user wants to construct the aggregates. If he selects *Euclidean distance* as a dissimilarity metric and *Ward's criterion* as an aggregation strategy, he will obtain the dendrogram and the graph of inertia presented in Fig.4. We notice that the inertia inter and intra-clusters release leaps for both small and great number of aggregates. Logically, it is not well suited to choose the optimal partition of modalities from these areas. Of course, a single cluster (including the whole set of individuals) and n clusters (where each one contains a single individual) are two insignificant partitions for the clustering problem. According to the results of Fig.4, one user can notice some other relevant leaps in the middle area of the graph and so he can decide to choose, for example, a partition with four clusters. He will obtain the following aggregates:

- $A_1 = \{$*Joshua tree at sunset, Sailboats racing, White daisy*$\}$
- $A_2 = \{$*Tropical coastline, Killer whale breaching, Two dried starfish*$\}$
- $A_3 = \{$*Red rose, Violin and red rose*$\}$
- $A_4 = \{$*Golf club and ball*$\}$

Each aggregate of the previous ones expresses a similarity of the medium blue channel characteristic (*L1Norm*$_B$), for a *very high* levels of Blue homogeneity and a *low* level of Red homogeneity.

Through this example, we prove that our operator can easily handle complex data and generate from measures and facts significant aggregates. These aggregates induce semantic relations different from the classical OLAP relations based on the membership order. Such aggregates are fully established in an automatic way with respect to the user choices and his analysis needs.

7. CONCLUSION

The objective of our work is to carry out a coupling between data mining and OLAP technology in order to satisfy the need of more elaborated on-line analysis on complex data like texts, images, sounds and videos. For this, we have created *OpAC*, a new aggregation operator based on the integration of the AHC in multidimensional data. The association of the two fields is an excellent solution to reinforce the weakness of each one. In this paper, we propose some extensions to our operator. In order to target precise analysis, we improved the formalization by giving the user flexibility in the selection of individuals and variables from a data cube. We proposed also a new tool, based on the inertia intra and inter-clusters, to evaluate the quality of partition generated by *OpAC*. This tool will help the user to decide about the best number of aggregates well suited for his objectives. The operator we proposed presents a possible way to realize on-line analysis on complex data. Unlike the classical OLAP context where aggregation is based on measures, *OpAC* take measures and descriptors of a cube to establish automatically a *semantic* aggregation.

We proved that establishing a *semantic* aggregating over complex data provides a promising result and significant aggregates that target precise analysis needs. Moreover, we believe that some new extensions of *OpAC* are still possible. In addition to its analysis vocation and its *semantic* aggregation power, we think that we can exploit the modalities' aggregates generated by *OpAC* and reuse them to reorganize the cube's dimensions. This leads us to get a new cube more suited for elaborated analysis. In a such cube we can identify remarkable regions where each one corresponds to a specific group of values. The aggregates evaluation tool we have constructed may present some limits since the inertia's functions are monotonous and doesn't clearly show a remarkable gap for a particular number of aggregates. Currently we are studying some better tools to evaluate the quality of partitions such as the *variance ratio criterion* [22] and the principal of *separability of classes* [28].

8. REFERENCES

[1] S. Chaudhuri. Data Mining and Database Systems: Where is the Intersection? *Data Engineering Bulletin*, 21(1):4–8, 1998.

[2] S. Chaudhuri and U. Dayal. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record*, 26(1):65–74, 1997.

[3] S. Chaudhuri, U. Fayyad and J. Bernhardt. Scalable Classification over SQL Databases. In IEEE Computer Society, editor, *ICDE-99*, pages 470–479, Sydney, Australia, 1999.

[4] M. Chen, Q. Zhu and Z. Chen. An integrated interactive environment for knowledge discovery from heterogeneous data resources. *Information and Software Technology*, 43:487–496, 2001.

[5] Q. Chen, U. Dayal and M. Hsu. An OLAP-based Scalable Web Access Analysis Engine. In *2^{nd} International Conference on Data Warehousing and Knowledge Discovery*, London, UK, September 2000.

[6] G. Dong, J. Han, J. M. W. Lam et al. Mining Multi-Dimensional Constrained Gradients in Data Cubes. In *The VLDB Conference*, pages 321–330, 2001.

[7] E. Favero and J. Robin. Using OLAP and Data Mining for Content Planning in Natural Language Generation. *Lecture Notes in Computer Science*, 1959:164–175, 2001.

[8] U. M. Fayyad, G. P. Shapiro and P. Smyth and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[9] S. Goil and A. Choudhary. High Performance Multidimensional Analysis and Data Mining. In *High Performance Networking and Computing Conference*, Orlando, USA, November 1998.

[10] S. Goil and A. Choudhary. PARSIMONY: An Infrastructure for parallel Multidimensional Analysis and Data Mining. *Journal of parallel and distributed computing*, 61:285–321, 2001.

[11] J. Han. Toward On-line Analytical Mining in Large Databases. *SIGMOD Record*, 27:97–107, 1998.

[12] R.M. Haralick, K.Shanmugan and I. Dinstein. Texture features for image classification. *Man and Cybernetics*, 3:610–622, 1973.

[13] I. Bartolini, E. Bertino, B. Catania et al. PAtterns for Next-generation DAtabase systems: preliminary results of the PANDA project. In *Undicesimo Convegno Nazionale su Sistemi Evoluti Per Basi Di Dati*, pages 293–300, Cetraro, Italy, 2003.

[14] T. Imielinski, L. Khachiyan and A. Abdulghani. Cubegrades: Generalizing association rules. *Data Mining and Knowledge Descovery*, 6:219–258, 2000.

[15] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. Communication of the ACM, 39(11):58–64, 1996.

[16] W. H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, 1996.

[17] R. Kimball. *The Data Warehouse toolkit*. John Wiley & Sons, 1996.

[18] A. Laurent, B. Bouchon-Meunier, A. Doucet et al. Fuzzy Data Mining from Multidimensional Databases. In Springer-Verlag, editor, *International Symposium on Computational Intelligence (ISCI'2000)*, pages 278–283, Kosice, Slovakia, 2000.

[19] A. Maedche, A. Hotho and M. Wiese. Enhancing Preprocessing in Data-Intensive Domains using Online-Analytical Processing. In Springer-Verlag, editor, *2^{nd} International Conference on Data Warehousing and Knowledge Discovery*, pages 258–264, London, UK, 2000.

[20] R. Meo, G. Psaila and S. Ceri. A New SQL-like Operator for Mining Association Rules. In *22nd VLDB conf.*, pages 122–133, Bombay, India, 1996.

[21] R. B. Messaoud, S. Rabaséda, O. Boussaid et al. OpAC: Opérateur d'analyse en ligne basé sur une technique de fouille de données. In *Revue des Nouvelles Technologies de l'Information (RNTI) (EGC 04)*, volume 2, pages 35–46, Clermont-Ferrand, France, January 2004.

[22] G.W. Milligan and M.C. Cooper. An examination of procedures for detecting the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

[23] T. Palpanas. Knowledge Discovery in Data Warehouses. *SIGMOD Record*, 29:88–100, 2000.

[24] K. Parsaye. OLAP and Data Mining: Bridging the Gap. *Database Programming and Design*, 10:30–37, 1997.

[25] H. Pinto, J. Han, J. Pei et al. Multi-dimensional Sequential Pattern Mining. In *Information and Knowledge Management (CIKM'01)*, Atlanta, USA, November 2001.

[26] S. Sarawgi. *iDiff*: Informative summarization of differences in multidimensional aggregates. *Data Mining And Knowledge Discovery*, 5:213–246, 2001.

[27] S. Sarawgi, R. Agrawal and N. Megiddo. Discovery-driven Exploration of OLAP Data Cubes. In *The 6^{th} Int. Conference on Extending Database Technology (EDBT)*, Valencia, Spain, March 2001.

[28] D.A. Zighed, S. Lallich and S.Muhlenbach. A statistical approach for separability of classes. In *Statistical Learning, Theory and Applications*, Paris, France, 2002.