Computing Appropriate Representations for Multidimensional Data

Yeow Wei Choong
LI - Université F. Rabelais
HELP Institute - Malaysia
choong_yw@help.edu.my

Dominique Laurent
LI - Université F. Rabelais
Tours - France
laurent@univ-tours.fr

Patrick Marcel
LI - Université F. Rabelais
Tours - France
marcel@univ-tours.fr

ABSTRACT

On-Line Analytical Processing (OLAP) provides an interactive query-driven analysis of multidimensional data based on a set of navigational operators like roll-up or slice and dice. In most cases, the analyst is expected to use these operations intuitively to find interesting patterns in a huge amount of data of high dimensionality.

In this paper, we propose an approach to enhance this analysis by preparing the data set so that the analyst can explore it in a more systematic and effective manner. More precisely we define a measurement of the quality of the representation of multidimensional data and we present a framework for investigating the computation of appropriate representations. We identify the problems of computing such representations and study them w.r.t. an OLAP restructuring operator.

1. INTRODUCTION

On-Line Analytical Processing (OLAP) [1, 3] technology provides a platform for analyzing data according to multiple dimensions (e.g., product, location, time) and multiple granularities (e.g., city, district, country). Data is presented under the form of a cube. A cube can be seen as a set of cells, and a cell represents the association of a *measure* with one *member* in each dimension. For example, if dimensions are products, stores and days, the measures of a particular cell can be the sales of one product in a particular store on a given day.

The user is provided with a set of operators for navigating through the data set to identify interesting and relevant patterns. This navigation is a query-driven process, and a number of proposals have investigated formal models and languages to this end (see [6, 9] for surveys). Obviously, as the size and the dimensionality of the data set increase, the whole process becomes very tedious and complex. To deal with this complexity, it has been recently pointed out [8, 7] that the manual effort spent in analysis could be reduced by anticipating the user strategy.

year 2000 sales					
Africa	3	5	6	3	5
America	4	6	7	5	7
Asia	2	4	6	2	5
Europe	4	5	7	4	6
	beer	milk	soda	water	wine

(a)

year 2000 sales					
America	4	5	6	7	7
Europe	4	4	5	6	7
Africa	3	3	5	5	6
Asia	2	2	4	5	6
	beer	water	milk	wine	soda

(b)

Figure 1: A 2-dimensional cube before and after restructuring

In typical OLAP analysis, the strategy is mostly based on observing the measures, whereas most of the OLAP restructuring operators are parameterized by members.

For example, consider the cube of Figure 1 (a). This cube displays sales of beer, milk, soda, water and wine in different continents during year 2000. Assume that the analyst wants to visualize the sales having the highest values on the one hand, and the lowest values on the other hand. The way the cube is represented does *not* provide such a visualization easily, because the cells are displayed according to the lexical ordering of the members in each dimension, and *not* according to the measures. On the other hand, it can be seen that the cube of Figure 1 (b) contains the same information as that of Figure 1 (a), but displays the sales in an appropriate way for the analyst. Indeed, the lowest values of sales are located down-left in the cube, whereas the highest values are located top-right. It should be noticed from the example that a clear distinction between a cube and its representation is needed here. This is precisely what we propose in this paper.

In our approach, the representation of a n-dimensional cube consists of n functions, each of them being a numbering of the members of a dimension. Given a cube C and one of its representations R, we assume that C is displayed according to the ordering defined by R. For example, the numbering defining the dimension product for the representation (a) of Figure 1 associates beer with 1, milk with 2, soda with 3, water with 4 and wine with 5. The numbering defining this dimension for the representation (b) associates beer with 1, water with 2, milk with 3, wine with 4 and soda with 5.

Representation (b) of Figure 1 can be interactively constructed by the user from representation (a) via some restructuring operators proposed in the OLAP context. These operators allow users to change the representation of the cube but *not* its logical structure: the association between one member in each dimension and the measure is preserved. For example, the *switch* operator [5, 6] allows users to exchange the position of 2 members on the axis corresponding to a given dimension while preserving the cells. The order over the columns in representation (b) of Figure 1 can be obtained from representation (a) by 1/ switching *milk* and *soda*, 2/ switching *soda* and *wine* and 3/ switching *wine* and *water*.

As a contribution to automating OLAP analysis, we propose to study how to arrange the representation of the cube according to its measures. We believe that computing appropriate representations can help to identify patterns which would otherwise remain unknown to the user. This contributes also to obtain the result of typical OLAP ranking queries like top-n.

We notice that even dimensions that are inherently ordered like e.g., time, can be rearranged so as to make some patterns apparent. For example, consider the cube of Figure 2 that displays monthly sales of chocolate in various regions. In representation (a) the months are depicted in the standard ordering, whereas in representation (b) the ordering is imposed by the measures. Representation (b) can be exploited by the analyst to discover that e.g., chocolate sales are the highest around new year and easter.

This paper presents a framework for investigating the quality of cube representations. Obviously there may be several ways of considering what an appropriate representation is and how to reach it.

Concerning appropriate representations, we define a cell as *misplaced* if there exists at least one other cell with lower measure and with greater or equal numberings in all dimensions. For example, the cell containing the sales of soda in Europe is misplaced in representation (a) of Figure 1. Indeed the cell containing the sales of water in Europe 1/ contains a lower measure and 2/ has greater numbering in dimension product, and the same numbering in dimension continent. We call *appropriate* the representations having the least number of misplaced cells, and we study the problem of finding these representations. To this end, we show that the *switch* operation proposed in the context of OLAP [5, 6] is the basic operator that allows us to compute these representations.

The main results of the paper are:

- First, we define a measurement for the quality of the representation of a cube by computing the number of its misplaced cells, and
- Second, we identify several problems related to the representation of cubes w.r.t. this measurement:
  - Test for the existence of a representation with no misplaced cells (called a *perfect* representation. Representation (b) of Figure 1 is an example of a perfect representation). In this case, we give the sequence of restructuring operations for reaching such a representation, if it exists. We show that this problem is polynomial with respect to the size of the cube.
  - If no representation having no misplaced cells exists, we outline the problems of finding representations having the least number of misplaced cells.
- Third, we propose an algorithm to test if a perfect representation exists and if so, compute this representation.

Related work

A variant of the switch operator has been defined in [5] in the context of 2-dimensional tabular databases. This operator allows users to exchange two rows of a matrix regardless of the status of the rows (members and measures are treated uniformly). However, in [5], the authors did not consider the problem of using this operation to restructure matrices in a more appropriate way for the user.

In [7, 8], Sarawagi & al. propose a new set of operators for reducing the number of roll-ups and drill-downs (changing the granularity of the representation) needed to discover abnormalities or to explain drops or increases in the values of the measures. Their work concentrates on the “vertical” aspect of OLAP data where the link between aggregated data is exploited.

While our motivations are essentially the same as the authors of [7, 8], our work is orthogonal to their approach in the sense that we concentrate on the “horizontal” aspect of OLAP data. Our goal is to reduce the number of restructuring operations used during the analysis. We are interested in the representation of the data at a given level and we do not take granularity into account.

The paper is organized as follows. The next section introduces basic definitions on the multidimensional data model, on the notion of representation, and on the quality measurement. In Section 3, we define and study the problems of finding appropriate representations. We conclude and discuss future work in Section 4. Proofs are omitted due to lack of space, and can be found in [2].

2. PRELIMINARIES

In this section, we give the formal definitions of the concepts used in this paper. The terminology concerning OLAP (members, measures, ...) is that of [6].

2.1 The multidimensional model

chocolate sales												
east	8	5	4	6	6	3	1	0	2	4	5	7
north	9	5	5	7	7	4	1	1	3	4	6	8
south	7	3	2	5	4	1	0	0	1	2	3	5
west	6	3	3	6	5	2	0	0	1	2	4	7
	jan	feb	mar	apr	may	jun	jul	aug	sep	oct	nov	dec

(a)

chocolate sales												
north	1	1	3	4	4	5	5	6	7	7	8	9
east	0	1	2	3	4	4	5	5	6	6	7	8
west	0	0	1	2	2	3	3	4	5	6	7	7
south	0	0	1	1	2	2	3	3	4	5	5	6
	aug	jul	sep	jun	oct	mar	feb	nov	may	apr	dec	jan

(b)

Figure 2: Restructuring a 2-dimensional cube with an inherently ordered dimension

In our model, we distinguish a cube from its representation. Intuitively, a cube is a logical multidimensional structure, and a representation can be seen as a way of displaying the cube to the analyst.

Definition 2.1 An n-dimensional cube, or simply a cube, is a tuple $\langle C, dom_1, \dots, dom_n, dom_m, m_C\rangle$ where

- C is the name of the cube,
- $dom_1, \dots, dom_n$ are n finite sets of symbols for the members associated with dimension $1, \dots, n$, respectively,
- let dom_{mes} be a finite totally ordered set of measures. Let $\perp$ be a constant not in dom_{mes} used to represent null values. Then $dom_m = dom_{mes} \cup \{\perp\}$, and $\perp$ cannot be compared to the elements of dom_{mes},
- m_C is a mapping from $dom_1 \times \dots \times dom_n$ to dom_m

Definition 2.2 A *representation* $R_C = \{rep_1, \dots, rep_n\}$ of a cube $\langle C, dom_1, \dots, dom_n, dom_m, m_C\rangle$ is a set of n bijective mappings $rep_1, \dots, rep_n$ such that for every $i = 1, \dots, n$, rep_i is a mapping from dom_i to the initial segment of $\mathbb{N}$ $\{1, \dots, |dom_i|\}$. The set of all different representations of a cube $\langle C, dom_1, \dots, dom_n, dom_m, m_C\rangle$ is denoted by S_{R_C}.

Given a representation R of a cube C, for every $i = 1, \dots, n$ and for every $m \in dom_i$, $rep_i(m)$ is called the *position* of m on dimension i in R.

Note that the notion of representation we propose does not associate a dimension with a particular axis (e.g., for 2-dimensions the vertical axis or the horizontal axis) for displaying the members. Only the relative position of the members in one dimension is relevant. On each dimension i, the values of dom_i are ordered according to their representation rep_i. In other words, placing value m of dom_i at the j^{th} position means that $rep_i(m) = j$.

The cardinality of S_{R_C} (i.e., the number of different representations of C) is the product of the number of different *rep* mappings for each dimension. Therefore, we have $|S_{R_C}| = \Pi_{i\in[1,\dots,n]}(|dom_i|!)$.

Example 2.1 Consider the 2-dimensional cube $\langle C, \{a,b\}, \{x,y\}, \{1,2,3,4\}, m_C\rangle$, where $m_C(a,x) = 1, m_C(a,y) = 2, m_C(b,x) = 3, m_C(b,y) = 4$. The number of different representations of this cube is $2! \times 2! = 4$. These representations, called R_1, R_2, R_3 and R_4 respectively, are displayed below. The representation R_1 is the set $\{rep_1, rep_2\}$ where rep_1 and rep_2 are defined by $rep_1(a) = 2, rep_1(b) = 1, rep_2(x) = 1, rep_2(y) = 2$.

As a convention throughout the paper, in this 2-dimensional example and the other examples, the horizontal axis is oriented from left to right and the vertical axis is oriented from bottom to top.

R_1

a	1	2
b	3	4
	x	y

R_2

b	3	4
a	1	2
	x	y

R_3

b	4	3
a	2	1
	y	x

R_4

a	2	1
b	4	3
	y	x

We note that all of these representations are different representations of the same cube C. Indeed, in C, we have for instance $m_C(a,y) = 2$, which holds in R_1, R_2, R_3 and R_4. The representations differ only in the ordering according to which the rows and the columns are displayed. On the other hand, the table below is *not* a representation of C since for instance, the measure associated with $\langle a, y\rangle$ is not 2.

a	1	3
b	2	4
	y	x

□

A cell is the association of a member in each dimension with a measure.

Definition 2.3 A *cell* c of a cube $\langle C, dom_1, \ldots, dom_n, dom_m, m_C\rangle$, is a tuple $\langle m_1, \ldots, m_n, m\rangle$ where $\forall i \in [1, \ldots, n], m_i \in dom_i, m \in dom_m$ and $m_C(m_1, \ldots, m_n) = m$.

A cell c of a cube C is an element of the graph of the function m_C. Therefore we feel allowed to consider a cube C as the set of its cells, and we write $c \in C$ to mean that c is a cell of C. A cell containing $\perp$ is called an *empty cell.*

Let $\langle C, dom_1, \ldots, dom_n, dom_m, m_C\rangle$ be a cube, $R_C = \{rep_1, \ \ldots, rep_n\}$ a representation of C and $c = \langle m_1, \ldots, m_n, m\rangle$ a cell of C. The position of c in C according to R_C is the tuple $\langle x_1, \ldots, x_n\rangle$ where $rep_i(m_i) = x_i$, for every $i \in [1, \ldots, n]$.

Note that the position of a cell in a representation is only based on the functions rep_i. This means that the position is invariant w.r.t. a rotation of the cube.

Example 2.2 Consider representation R_1 of Example 2.1. For this representation, the position of the cell $c_1 = \langle a, x, 1\rangle$ is the tuple $\langle 2, 1\rangle$, and the position of the cell $c_4 = \langle b, y, 4\rangle$ is the tuple $\langle 1, 2\rangle$. □

2.2 Cell arrangement

We can now define the ordering over cell positions.

Definition 2.4 Let $\langle C, dom_1, \ldots, dom_n, dom_m, m_C\rangle$ be a cube and $R_C = \{rep_1, \ldots, rep_n\}$ a representation of C. Let $c = \langle m_1, \ldots, m_n, m\rangle$ and $c' = \langle m'_1, \ldots, m'_n, m'\rangle$ be two cells of C. We define the relation $\prec_{R_C}$ as a partial ordering over cells by $c \prec_{R_C} c' \iff \forall i \in [1, \ldots, n], rep_i(m_i) \leq rep_i(m'_i)$.

Example 2.3 Consider the cube of Example 2.1. This cube has cells $c_1 = \langle a, x, 1\rangle$, $c_2 = \langle a, y, 2\rangle, c_3 = \langle b, x, 3\rangle$, and $c_4 = \langle b, y, 4\rangle$. Considering the representation R_1, we have $c_3 \prec_{R_1} c_1$, $c_3 \prec_{R_1} c_2$, $c_3 \prec_{R_1} c_4$, $c_1 \prec_{R_1} c_2$, $c_4 \prec_{R_1} c_2$. Note that c_1 cannot be compared with c_4 w.r.t. $\prec_{R_1}$. □

Now, we define what we call a *misplaced* cells.

Definition 2.5 Let $\langle C, dom_1, \ldots, dom_n, dom_m, m_C\rangle$ be a cube and R_C a representation of C. A cell $c = \langle m_1, \ldots, m_n, m\rangle$ of C is *misplaced w.r.t.* R_C if $m \neq \perp$, and

- $\exists c_1 = \langle m'_1, \ldots, m'_n, m'\rangle \in C$ such that $c \prec_{R_C} c_1$ and $m > m'$, or
- $\exists c_2 = \langle m''_1, \ldots, m''_n, m''\rangle \in C$ such that $c_2 \prec_{R_C} c$ and $m'' > m$.

For a cube C, a representation R_C of C and a cell $c \in C$, we define the function $f_{R_C}(c) = 1$ if c is misplaced w.r.t. R_C, 0 otherwise.

Then, the measurement we propose is simply the total number of misplaced cells in a cube.

Definition 2.6 Given a cube C and a representation R_C of C, we define $M_{R_C}(C)$ by $M_{R_C}(C) = \sum_{c_i \in C} f_{R_C}(c_i)$. $M_{R_C}(C)$ is the total number of misplaced cells in C w.r.t. the representation R_C.

With this measurement, we can characterize the representations of a cube.

Definition 2.7 Let $\langle C, dom_1, \ldots, dom_n,\ dom_m, m_C\rangle$ be a cube and let S_{R_C} be the set of all representations of C.

- A representation R_C of C is a *Perfect Representation (PR)* if $M_{R_C}(C) = 0$.
- A representation R_C of C is an *Optimal Representation (OR)* if $\nexists R'_C \in S_{R_C}, M_{R'_C}(C) < M_{R_C}(C)$.

Obviously for a given cube, a PR may not exist, and there exists at least one OR. Moreover, if a PR exists it may not be unique.

Example 2.4 Consider the representations R_1 and R_2 of the cube C in Example 2.1. The number of misplaced cells in R_1 is $M_{R_1}(C) = 4$, whereas R_2 is a PR of C (i.e., $M_{R_1}(C) = 0$). Now if we consider the table below as a representation of a cube, there exists no PR of this cube. This is so because the lowest and highest measures are on the same row. Since this must hold in every representation of the cube although this cannot hold in any PR, this cube has no PR.

2	3
1	4

□

3. THE PROBLEMS

In this section we study the problems of using the measurement of Definition 2.6 to find appropriate representations of cubes. We first define the operation used to change the representation of a cube.

3.1 Arranging the cube

The *switch* operation [5, 6] is an OLAP operation that consists in interchanging the positions of two members of a dimension of a cube. In our framework, the switch operation is the basic operation to go from one representation of a cube to another.

Definition 3.1 Let $\langle C, dom_1, \ldots, dom_n, dom_m, m_C \rangle$ be a cube and S_{R_C} the set of all representations of C. A switch on dimension j of members p and q, denoted by $switch(j,p,q)$, is a function from S_{R_C} to S_{R_C} such that, for every $R_C = \{rep_1, \ldots, rep_n\}$ in S_{R_C}, $switch(j,p,q)(R_C) = R'_C$ where $R'_C = \{rep'_1, \ldots, rep'_n\}$ is defined by:

- for every $i = 1, \ldots, n$, if $i \neq j$, then $rep_i = rep'_i$,
- $rep_j(p) = rep'_j(q)$ and $rep_j(p) = rep'_j(q)$
- for every m in dom_j different than p and q, $rep_j(m) = rep'_j(m)$.

Notice that according to the first point of Definition 3.1, applying a switch operation on two members in one dimension leaves unchanged the positions of the members in the other dimensions.

Example 3.1 Consider the cube of Example 2.1 and its representations R_1 and R_2. R_2 is the result of the operation $switch(1,a,b)$ applied to R_1. In other words, $R_2 = switch(1,a,b)(R_1)$. □

Definition 3.2 A finite composition of switches is called an *arrangement.*

Example 3.2 Consider the representations of Example 2.1. We have $switch(1,a,b)(R_1) = R_2$, $switch(2,x,y)(R_2) = R_3$. Thus $switch(2,x,y)(switch(1,a,b)(R_1)) = R_3$. Therefore, $R_3 = arr(R_1)$ where arr is the arrangement defined by $switch(2,x,y) \circ switch(1,a,b)$. □

As for the switch operation, it is obvious that applying an arrangement involving only one dimension leaves the position of the members of the other dimensions unchanged.

The following proposition shows that all representations of a cube can be obtained through arrangements.

Proposition 3.1. *Let $\langle C, dom_1, \ldots, dom_n, dom_m, m_C \rangle$ be a cube and let S_{R_C} be the set of all representations of C. Given any two representations R_1 and R_2 of S_{R_C}, there exists an arrangement arr such that $arr(R_1) = R_2$.*

3.2 The Perfect Representation problem

We are interested in the following problem that we call the Perfect Representation (PR) problem: For a given cube and a given representation of this cube, test whether there exists at least one PR, and if so, compute one PR. If more than one PR exist, then compute the number of PRs and list all the arrangements leading to these PRs. In this paper, we study the first part of the problem, namely the test of the existence of a PR of a given cube and its computation if it exists.

To deal with this problem, we consider separately three cases. We first consider the simple case of cubes containing no null values, and having no duplicates in their rows. This gives rise to a basic algorithm for solving the PR problem. Then we consider cubes containing no null values, but where rows may contain duplicates. Finally, we consider cubes containing null values, but where rows do not contain duplicates.

We now introduce formally the notion of row for the sake of readability. Intuitively, a row is a set of cells where all coordinates but one are fixed.

Definition 3.3 Let $\langle C, dom_1, \ldots, dom_n, dom_m, m_C \rangle$ be a cube. A row r in dimension k is the set of cells of C $\{\langle m_1, \ldots, m_{k-1}, j, m_{k+1}, \ldots, m_n, m \rangle \mid j \in dom_k\}$. This row is identified by the tuple $\langle m_1, \ldots, m_{k-1}, m_{k+1}, \ldots, m_n \rangle$, where $m_i \in dom_i$ for every i in $[1, \ldots, k-1, k+1, \ldots, n]$.

As for cells and cubes, we feel allowed to denote by $r \in C$ the fact that every cell belonging to r also belongs to C.

Given a representation $R = \{rep_1, \ldots, rep_k, \ldots, rep_n\}$ of a cube, a row r in dimension k, and a cell $c = \langle m_1, \ldots, m_k, \ldots, m_n, m \rangle$ of r, the position of c in r is simply $rep_k(m_k)$.

Definition 3.4 Let $\langle C, dom_1, \ldots, dom_n, dom_m, m_C \rangle$ be a cube, let R_C be a representation of C. A row r is sorted in R_C if $\forall c = \langle m_1, \ldots, m_n, m \rangle, c' = \langle m'_1, \ldots, m'_n, m' \rangle \in r$ with $m \neq \perp$ and $m' \neq \perp$, $c \prec_{R_C} c' \Longrightarrow m \leq m'$. Otherwise the row r is unsorted.

Given a representation R and a row r in dimension k, sorting r is simply changing rep_k. Note that in a sorted row, empty cells can appear anywhere. Based on usual algorithms for sorting one-dimensional arrays, we have the following lemma.

Lemma 3.2. *For a given cube C, a given representation R_C of C and a given row r there is an arrangement that sorts the row.*

If r is a row and R is a representation, sorting a row means applying an arrangement to R so that r is sorted in the resulting representation. Obviously, sorting a row in dimension k implies assigning a position to the members of dom_k.

Example 3.3 Consider Example 2.1. The row $\langle y\rangle$ is the set $\{\langle a, y, 2\rangle, \langle b, y, 4\rangle\}$. Moreover, this row is sorted in R_2. □

The following theorem, of which the proof is an immediate consequence of Definition 2.5, is the basic result on which rely all proofs of the subsequent propositions and corollaries.

THEOREM 3.3. *A representation of a cube is a PR if and only if every row in every dimension is sorted.*

Now, we proceed to study the PR problem in the following three cases:

- Case 1: each row of the cube contains no duplicates and no null values,
- Case 2: each row of the cube can contain duplicates but no null values,
- Case 3: each row of the cube can contain null values but no duplicates.

The last case (each row can contain both duplicates and null values) is still an open issue.

Case 1: No duplicates and no null values in each row

We first consider the case where every row in every dimension contains no duplicates and no null values. In this case, we show that the existence of a PR can be efficiently tested by sorting only one row in each dimension. Moreover, when a PR actually exists, it is unique and our method computes it. Our method is based on the following two propositions and corollary.

PROPOSITION 3.4. *Let C be a cube such that each row contains no duplicates and no null values. There exists at most one PR of C.*

PROPOSITION 3.5. *Let C be a cube such that each row contains no duplicates and no null values. If there exists a representation such that for one dimension, a row r is sorted and another row r' is unsorted, then there exists no PR.*

COROLLARY 3.6. *Let C be a cube such that each row contains no duplicates and no null values, and for which a PR exists. Let R be a representation of C. If in R one row is sorted in each dimension, then R is a PR.*

At this point, a simple algorithm can be given to solve the PR problem for a cube where each row contains no duplicates and no null values.

Algorithm 3.1

```
Input: A representation of a cube $C$
Output: The PR of $C$ or the indication "no PR"

for each dimension $k$ of $C$ do
    choose a row $r$ in dimension $k$
    sort $r$
    for every other row $r'$ in dimension $k$ do
        check if $r'$ is sorted
        if $r'$ is unsorted then
            exit with output "no PR"
```

This algorithm is polynomial in the number of cells of the cube, since it only sorts one-dimensional arrays (one row in each dimension) or tests if one-dimensional arrays are sorted.

Case 2: Dealing with duplicates

If we allow duplicates, but no null values, to appear in a row, sorting a row in each dimension is necessary but no more sufficient for computing a PR. For instance, consider the cube of which representations R_1 and R_2 are depicted below. Sorting row $\langle a\rangle$ may lead to representation R_1 which is not perfect, since row $\langle b\rangle$ is unsorted. On the other hand, sorting row $\langle b\rangle$ leaves row $\langle a\rangle$ unchanged and gives a PR.

R_1

b	4	3
a	1	1
	x	y

R_2

b	3	4
a	1	1
	y	x

Definition 3.5 Let $C = \langle dom_1, \dots, dom_n, dom_m, m_C\rangle$ be a cube, $R = \{rep_1, \dots, rep_n\}$ be a representation of C, and $r = \langle m_1, \dots, m_{k-1}, m_{k+1}, \dots, m_n\rangle$ be a row of dimension k. A sequence of duplicates in r is an interval $I = [i_1, i_2]$ of $\mathbb{N}$ such that for all $i, j \in I$, $m_C(m_1, \dots, m_{k-1}, rep_k^{-1}(i), m_{k+1}, \dots, m_n) = m_C(m_1, \dots, m_{k-1}, rep_k^{-1}(j), m_{k+1}, \dots, m_n)$. Given a row r, a sequence of duplicates I in r is maximal if there is no sequence of duplicates J in r such that $I \subset J$.

Given a representation of a cube, a row r in dimension k, and an interval I of $\mathbb{N}$, the contiguous part of r w.r.t. I is defined by $r_I = \{c \in r \mid c = \langle m_1, \dots, m_k, \dots, m_n, m\rangle$ and $rep_k(m_k) \in I\}$.

PROPOSITION 3.7. *Let C be a cube and R a representation of C. Let r be a sorted row in R containing p maximal sequences of duplicates $I_1, \dots, I_p$. If there exists a row r' in the same dimension that is still unsorted after having sorted every contiguous part of r' w.r.t. $I_1, \dots, I_p$, then there exists no PR.*

Example 3.4 Consider a cube of which representations R_1 and R_2 are depicted below. Suppose we sort row $\langle b \rangle$ first, so as to obtain representation R_1. The next step is to sort row $\langle a \rangle$ *without affecting row* $\langle b \rangle$. The only possibility is to switch members x and y. Once done, we obtain representation R_2 where row $\langle a \rangle$ is still unsorted. Therefore there is no PR of this cube.

R_1

a	4	3	1
b	1	1	2
	x	y	z

R_2

a	3	4	1
b	1	1	2
	y	x	z

□

At this point we can give an algorithm that outputs a PR of a cube where the rows contain duplicates but no null values, if any. Otherwise, the algorithm indicates that no PR exists.

Algorithm 3.2

```
Input: A representation of an $n$-dimensional cube $C$

Output: A PR of $C$ or the indication "no PR"

Variable: Two sets $D$ and $D'$ of sequences of duplicates

for each dimension $k$ of $C$ do
    let $D = \{I\}$ with $I = [1, n]$
    find the row $r$ in dimension $k$ having the lowest number of duplicates
    repeat until every row is marked
        sort $r_I$ for every $I \in D$
        check if $r$ is sorted
        if $r$ is unsorted then
            exit with output "no PR"
        else
            for each $I$ in $D$ do
                $D' = \emptyset$
                compute $I_1, \ldots, I_p$ the sequences of duplicates in $r$
                $D' = D' \cup \{I_j \cap I \mid I_j \cap I \neq \emptyset, I \in D, j = 1, \ldots, p\}$
            $D = D'$
            mark $r$
            find the unmarked row $r$ having the lowest number of duplicates
```

It is easy to see that this algorithm is polynomial in the number of cells of the cube.

Case 3: Dealing with null values

In what follows, we assume that the rows of a cube can contain null values but no duplicates. We recall from Definition 2.5 that changing the position of a null value in a row does not affect the fact that the row is sorted or not. Thus, a row containing null values can be sorted in different ways, which results in more flexibility when looking for PRs. For instance, consider the cube of which representations R_1 and R_2 are depicted below. Sorting row $\langle a \rangle$ may lead to representation R_1 which is not perfect, since row $\langle b \rangle$ is unsorted. On the other hand, sorting row $\langle b \rangle$ does not affect the fact that row $\langle a \rangle$ is still sorted, and gives a PR.

R_1

b	4	3
a	1	⊥
	x	y

R_2

b	3	4
a	⊥	1
	x	y

This flexibility for sorting rows imposes that many combinations have to be explored when looking for PRs. For example, suppose we must arrange the following representation.

⋮	⋮	⋮	⋮	⋮	⋮
b	1	⊥	4	2	⊥
a	1	2	3	⊥	⊥
	v	w	x	y	z

Suppose we have sorted row $\langle a \rangle$ and we must sort row $\langle b \rangle$. As $\perp$ can be placed anywhere, the following two possibilities are valid.

b	1	2	⊥	4	⊥
a	1	⊥	2	3	⊥
	v	y	w	x	z

b	1	⊥	2	4	⊥
a	1	2	⊥	3	⊥
	v	w	y	x	z

Looking for a PR means that each of these possibilities has to be checked when sorting the rows. Therefore, we conjecture that computing a PR is non-polynomial. However, based on Proposition 3.5, a polynomial algorithm can be given to indicate the non-existence of a PR of a given cube. To this end, we define sequences of null values analogously as sequences of duplicates.

Definition 3.6 Let $C = \langle dom_1, \ldots, dom_n, dom_m, m_C \rangle$ be a cube, $R = \{rep_1, \ldots, rep_n\}$ be a representation of C, and $r = \langle m_1, \ldots, m_{k-1}, m_{k+1}, \ldots, m_n \rangle$ be a row in dimension k. A sequence of null values in r is an interval $I = [i_1, i_2]$ of $\mathbb{N}$ such that for every $i \in I$, $m_C(m_1, \ldots, m_{k-1}, rep_k^{-1}(i), m_{k+1}, \ldots, m_n) = \perp$.

Once a row is sorted, we can assume without loss of generality that this row contains at most one sequence of null values. Given a row r and an interval $I = [i, p]$ of $\mathbb{N}$, where $p = |dom_k|$, we call r_{p-I} the contiguous part of r defined by $r_{p-I} = \{c \in r \mid c = \langle m_1, \ldots, m_k, \ldots, m_n, m\rangle$ and $rep_k(m_k) \in [1, i]\}$.

Algorithm 3.3

```
Input: A representation of a cube $C$

Output: The indication "no PR" if there exists no PR of $C$

for each dimension $k$ of $C$ do
    let $p$ be the number of cells of all rows in dimension $k$
    choose a row $r$ in dimension $k$
    sort $r$ so that $I = [i, p]$ is the only sequence of null values in $r$
    for every other row $r'$ in dimension $k$ do
        check if $r'_{p-I}$ is sorted
        if $r'_{p-I}$ is unsorted then
            exit with output "no PR"
```

4. CONCLUSION

In this paper we have introduced an approach to enhance the query-driven analysis of multidimensional data, based on representations of cubes according to their measures. We have introduced a measurement to compute the quality of the representation, and we have proposed an algorithm to find the representation of a cube containing no null values, for which this measurement is optimal, if it exists.

Our current and future work encompasses the following open issues:

- Study the PR problem in the case of null values. We conjecture that computing a PR in the case of null values is not polynomial, and therefore heuristics should be given to deal with the problem efficiently.
- Computing all the PRs of a cube. As stated in Section 2, if a PR of a cube exists, it may not be unique. We conjecture that outputing every PR of a cube is not polynomial, but that computing the total number of PRs is polynomial.
- Implementation of the approach discussed in the paper. The algorithms given Section 3 are naive algorithms, that should be reworked in order to propose an efficient implementation.
- Study of other problems in this framework. As stated in Section 2, a PR may not exist. Thus we can define two other problems that we shall study in the future:
  - The OR problem (cf. Definition 2.7): for a given cube and a given representation of this cube, find all ORs, and list all arrangements leading to these ORs.
  - The t-OR problem: given a cube C and a threshold t, find a representation R_C of C such that $M_{R_C}(C) \leq t$ if it exists. If there exists at least one such representation, list all arrangements leading to these representations.
- Use of other OLAP operations to solve the problems. In this paper we restrict ourselves to the switch operation to compute appropriate representations. It would be interesting to study how the other OLAP operations [4, 5, 6] behave w.r.t. the problems introduced above. For example in the presence of hierarchies, can we use the roll-up operator to reach a PR?

5. REFERENCES

[1] Surajit Chaudhuri and Umeshwar Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26(1):65–74, 1997.

[2] Yeow Wei Choong, Dominique Laurent, and Patrick Marcel. Computing appropriate representations for multidimensional data. Research report, Laboratoire d'Informatique, Universit de Tours, 2001. To appear.

[3] E. F. Codd, S. B. Codd, and C. T. Salley. Providing OLAP (On-Line Analytical Processing) to user-analysts: An IT mandate [on-line]. 31p. White Paper, 1993.

[4] Marc Gyssens and Laks V. S. Lakshmanan. A foundation for multi-dimensional databases. In *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*, pages 106–115. Morgan Kaufmann, 1997.

[5] Marc Gyssens, Laks V. S. Lakshmanan, and Iyer N. Subramanian. Tables as a paradigm for querying and restructuring. In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, 1996, Montreal, Canada*, pages 93–103. ACM Press, 1996.

[6] P. Marcel. Modeling and querying multidimensional databases: An overview. *Networking and Information Systems Journal*, 2(5-6):515–548, 1999.

[7] Sunita Sarawagi. Explaining differences in multidimensional aggregates. In *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 42–53. Morgan Kaufmann, 1999.

[8] Sunita Sarawagi, Rakesh Agrawal, and Nimrod Megiddo. Discovery-driven exploration of OLAP data cubes. In *Advances in Database Technology - EDBT'98, 6th International Conference on Extending Database Technology, Valencia, Spain, March 23-27, 1998, Proceedings*, volume 1377 of *Lecture Notes in Computer Science*, pages 168–182. Springer, 1998.

[9] Panos Vassiliadis and Timos K. Sellis. A survey of logical models for OLAP databases. *SIGMOD Record*, 28(4):64–69, 1999.