RoK: Roll-Up with the K-means Clustering Method for Recommending OLAP Queries

Fadila Bentayeb and Cécile Favre

Université de Lyon (ERIC - Lyon 2)
5 av. Pierre Mendès-France
69676 Bron Cedex, France
bentayeb@eric.univ-lyon2.fr
cecile.favre@univ-lyon2.fr
http://eric.univ-lyon2.fr

Abstract. Dimension hierarchies represent a substantial part of the data warehouse model. Indeed they allow decision makers to examine data at different levels of detail with On-Line Analytical Processing (OLAP) operators such as drill-down and roll-up. The granularity levels which compose a dimension hierarchy are usually fixed during the design step of the data warehouse, according to the identified analysis needs of the users. However, in practice, the needs of users may evolve and grow in time. Hence, to take into account the users' analysis evolution into the data warehouse, we propose to integrate personalization techniques within the OLAP process. We propose two kinds of OLAP personalization in the data warehouse: (1) adaptation and (2) recommendation.

Adaptation allows users to express their own needs in terms of aggregation rules defined from a child level (existing level) to a parent level (new level). The system will adapt itself by including the new hierarchy level into the data warehouse schema. For recommending new OLAP queries, we provide a new OLAP operator based on the K-means method. Users are asked to choose K-means parameters following their preferences about the obtained clusters which may form a new granularity level in the considered dimension hierarchy. We use the K-means clustering method in order to highlight aggregates semantically richer than those provided by classical OLAP operators. In both adaptation and recommendation techniques, the new data warehouse schema allows new and more elaborated OLAP queries.

Our approach for OLAP personalization is implemented within Oracle 10 g as a prototype which allows the creation of new granularity levels in dimension hierachies of the data warehouse. Moreover, we carried out some experiments which validate the relevance of our approach.

Key words: OLAP, personalization, adaptative system, recommendation, schema evolution, clustering, K-means, analysis level, dimension hierarchy.

1 Introduction

Traditional databases aim at data management, i.e., they help organizing, structuring and querying data. Data warehouses have a very different vocation: analyzing data by exploiting specific multidimensional models (star, snowflake and constellation schemas). Data are organized around indicators called measures, and analysis axes called dimensions. Dimension attributes can form a hierarchy which compose various granularity levels. They allow users (decision makers) to examine data at different levels of detail by using On-Line Analytical Processing (OLAP) tools. Indeed, OLAP allows users to acquire a dynamic manipulation of the data contained in the data warehouse, in particular through hierarchies that provide navigational structures to get summarized or detailed data by rolling up or drilling down.

The main objective of data warehouses is to facilitate decision making. In order to satisfy the whole analysis needs of the majority of the users, a promising issue consists in considering a personalization process for OLAP analysis. By personalization, we mean considering the user to be in the center of the decision system, taking into account his or her own preferences, needs, etc. Research concerning personalization constitutes an emerging topic for the data warehouse domain [1].

In a previous work [2], we proposed an original approach to allow schema evolution in data warehouses independently from data sources. In this paper, we extend this approach to support users' analyses personalization in an interactive way following two main techniques, namely adaptation and recommendation. We propose then a general framework to integrate OLAP personalization in data warehouses. The originality of our framework consists in including additional information and/or knowledge into the data warehouse for further analysis. The solution we propose is implemented by creating new dimension hierarchies into the data warehouse model in order to get new OLAP queries.

In the adaptation technique, users define their additional information under the form of aggregation rules from a child level (existing level) to a parent level (new level). Then, the system adapts to the data warehouse schema by creating the new granularity level in a dimension hierarchy which allows the user to get his/her own personalized analysis.

In the recommendation technique, classical tools are designed to help users to find items within a given domain, according to their own preferences (user profile). The recommendation technique we propose is slightly different from classical ones since we use data mining techniques to extract relevant clusters. These latter possibly represent significant and more elaborated OLAP queries. Hence, users can fix the algorithm parameters in an interactive way until the suggestion of the system coincides with the users' objectives, validating, therefore, the suggestion. We define more precisely a new Roll-up operator based on K-means (RoK) method that creates a new (parent) level to which, a child level rolls up in a dimension hierarchy. Our RoK operator is indeed different from classical OLAP operators since it combines data mining and OLAP tools.

To integrate efficiently our proposition in the OLAP process, we implemented the K-means method inside the Oracle 10g Relational DataBase Management System (RDBMS) under the form of a stored procedure. This allows treating efficiently large data sets directly inside the data warehouse, like an OLAP operator. In addition, we carried out some experiments which validate the relevance of our approach.

The rest of this paper is organized as follows. In Section 2, we present related work regarding personalization, combining OLAP and data mining and schema evolution in data warehouses. Then, in Section 3, we present our approach for personalized OLAP analysis in data warehouses. To illustrate our purpose, we provide an example from a real case study in Section 4. Section 5 details our data-mining based approach to recommend new OLAP queries and presents the data warehouse model evolution which supports our OLAP personalization approach. Section 6 presents the experiments we performed to validate our approach. We finally conclude this paper and provide some research perspectives in Section 7.

2 Related Work

Personalization in data warehouses is closely related to various research areas that we evoke in this section.

2.1 Personalization

Personalization has been studied since many years and constitutes always a hot topic in domains such as information retrieval (IR), databases (DB) and human-computer interaction (HCI). The general idea is to provide pertinent answers/ adapted interfaces to the user according to his/her individual preferences [3]. Personalization is usually based on the concept of profile [4]. This profile is used to model the user himself, his/her needs, the group he/she belongs to and so on.

This profile is not defined in a standard way. In the context of HCI, the profile contains information that allows the adaptation of the interface according to preferences [5]. In the context of IR, the profile can be represented as a set of key words with ponderation [6] or a set of utility functions to express in a relative way domains of interest [7]. In the context of DB, the profile can contain the usual queries of a user i.e. usual predicates, or order in these predicates [8, 9]. Thus, the system exploits these predicates to enrich queries and to provide more pertinent results.

Since data warehouses are characterized by voluminous data and are based on a user-centered analysis process, including personalization into the data warehousing process becomes a new research issue [1]. Works in this domain are inspired from those proposed for personalization in IR, DB, and HCI. For example, selecting data for visualization, based on users' preferences [10] or facilitating the navigation into the data cube [11, 12], or recommending some possible analyses according to navigation of other users [13].

2.2 Combining OLAP and data mining

OLAP operators have a powerful ability to organize and structure data allowing exploration and navigation into aggregated data. Data mining techniques are known for their descriptive and predictive power to discover knowledge from data. Thus OLAP and data mining are used to solve different kinds of analytic problems: OLAP provides summary data and generates rich calculations while data mining discovers hidden patterns in data. OLAP and data mining can complement each other to achieve, for example, more elaborated analysis.

In the context of data warehouses and OLAP, some data mining techniques can be used as aggregation operators. Thus many works are now focused on providing more complex operators to take advantages from the analysis capabilities of the data mining [14, 15]. In our approach, we are going beyond these proposals by exploiting data mining not only at the final stage as OLAP operators but also to consider the data warehouse evolution and take into account users' preferences.

2.3 Data warehouse model evolution

During OLAP analysis, business users often need to explore fact trends over the time dimension. This requires time-variant and non-volatile data. Thus, dimension updates and schema evolutions are logically prohibited because they can induce data loss or erroneous results. To deal with this problem, two categories of research emerged. The first category recommends extending the multidimensional algebra to update the model with a set of schema evolution operators [16, 17] while the second category proposes temporal multidimensional data models [18, 19]. These works manage and keep the evolutions history by time-stamping relations.

3 Personalization in Data Warehouses

3.1 General approach

Generally, to carry out OLAP analysis, the user generates a data cube by selecting dimension level(s) and measure(s) which will satisfy his/her needs. Then, the user explores the obtained cube to detect similarities between data facts or dimension instances. For that, he/she explores different levels within a dimension. To help the user in this step, we propose to personalize his/her analysis according to his(her) individual needs and preferences. In this context, we provide a general framework for OLAP personalization shown in Figure 1.

To achieve OLAP personalization, our key idea consists in integrating new information or knowledge inside the data warehouse. Hence, we consider two kinds of knowledge: (1) explicit knowledge expressed by users themselves, and (2) knowledge extracted from the data.

In our framework, we identify four main processes: (1) knowledge acquisition which requires either explicit information or extracted information from the data

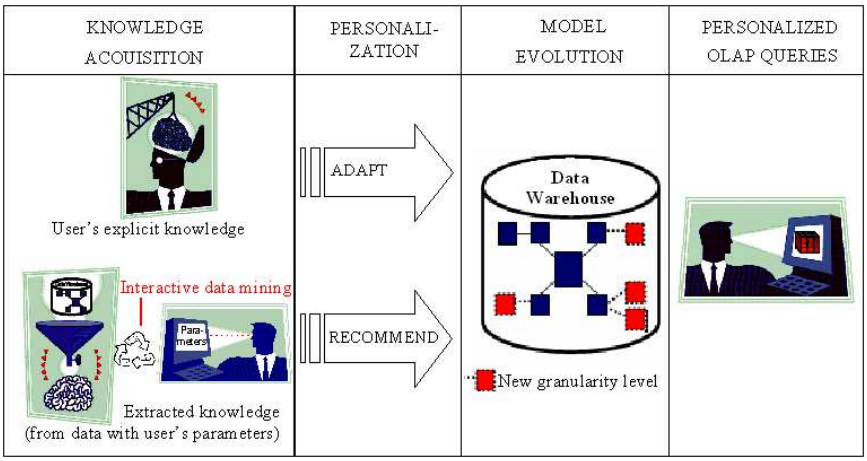

Fig. 1. Framework for OLAP personalization

using data mining techniques, (2) knowledge integration into the data warehouse, (3) data warehouse schema evolution, and (4) OLAP queries personalization.

In the following, we present our approach for OLAP personalization which is composed of two techniques, namely adaptation and recommendation. Each technique respects the four steps of our framework mentioned above.

3.2 Adaptation-based Personalization

Our adaptative data warehouse system aims to personalize analysis by integrating users's knowledge into the data warehouse, providing an answer to individual analysis needs. The user is asked to define his/her own knowledge in terms of if-then rules representing aggregations from a child level to a parent level. These rules are used to create a new granularity level in the considered dimension hierarchy. The if-clause, indeed, determines conditions on the attributes of the child level for grouping instances together forming a partition. The then-clause determines aggregates of the parent level, each one corresponds to a subset of the partition. In this case, the system is adaptative since it adapts itself by evolving the data warehouse schema to take into account new user's information.

3.3 Recommendation-based Personalization

Classical OLAP operators are designed to create intuitive aggregates. However, to help users to find non expected and relevant aggregates expressing deep relations within a data warehouse, we propose to combine data mining techniques and OLAP. We choose to use the K-means clustering method, because of the format of its result, which is defined as a partition. The user is asked to fix the algorithms' parameters in an interactive way for obtaining relevant clusters. Then, the system recommends to the user the obtained clusters. If these latter

are validated by the user, they are integrated into the data warehouse and a new hierarchy level is then created, allowing new OLAP queries which are proposed to the user.

To create a new level in a dimension hierarchy, we consider only classical hierarchies in both adaptation and recommendation techniques. In other words, each child occurrence in a child level is linked to a unique parent occurrence in a parent level but each parent occurence can be associated with several child occurrences as showed in Figure 2.

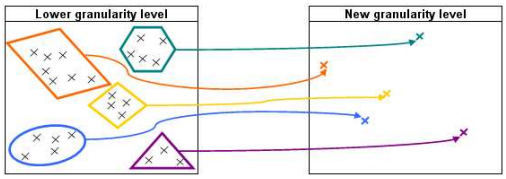

Fig. 2. Creation of a new granularity level.

4 Illustrative Example

To illustrate our approach for OLAP personalization in data warehouses, we use the example of the LCL company, which is a french bank we are collaborating with. We focus on an extract of the data warehouse concerning the management of accounts. We consider two measures which are the Net Banking Income (`NBI`) and the `COST`. The `NBI` is the profit obtained from the management of customers' accounts. As its name suggests, the second measure corresponds to the cost of customers' accounts. These measures are observed according to several dimensions: `CUSTOMER`, `AGENCY` and `YEAR` (Figure 3a). The dimension `AGENCY` is organized as a hierarchy which defines the geographical commercial structure of LCL, i.e. `AGENCY` is grouped into `COMMERCIAL UNIT`, which is grouped into `DIRECTION`.

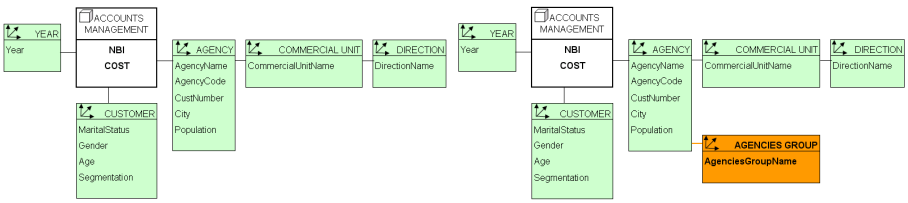

Fig. 3. a) Initial LCL data warehouse. b) Personalized LCL data warehouse.

Now, let us take the case of the person in charge of student products in the LCL french bank. He/she knows that there are three types of agencies: "student" for agencies which gather only student accounts, "foreigner" for agencies whose customers do not live in France, and "classical" for other agencies.

However, this information is not stored in the data warehouse and therefore it cannot be used to carry out analysis about "student" agencies. Our adaptation-based personalization approach consists then in allowing the user to integrate his specific knowledge into the data warehouse. Then the system adapts itself according to this new user's knowledge by generating a new granularity level: `AGENCIES GROUP` that corresponds to the desired level in the `AGENCY` dimension (Figure 3b).

Suppose now that the user wants to group agencies together according to the population of the city where the agency is located (Population) and the number of customers (CustNumber) but he/she doesn't kwow really how. To achieve this goal, our recommendation-based personalization approach consists then in extracting knowledge automatically from the data warehouse to provide possibly relevant clusters of agencies by using an unsupervised learning method, namely K-means. The system is then in charge of recommending to the user a new granularity level `AGENCIES GROUP` (Figure 3b) based on the obtained agencies clusters. The `AGENCIES GROUP` granularity level allows more elaborated OLAP queries. For instance, one may observe the evolution of `ACCOUNTS MANAGEMENT` (`NBI`) by `CUSTOMER` (Segmentation), `YEAR` (Year) and `AGENCIES GROUP` (AgenciesGroupName).

In the following, we detail our approach to recommend new OLAP queries based on the K-means method.

5 Framework for Recommending OLAP Queries

5.1 Basic Definitions

A data warehouse is a multidimensional database that can be defined as follows: $\mu = (\delta, \varphi)$, where δ is a set of **dimensions** and φ is a set of **facts** [17].

A **dimension schema** is a tuple $D = (L, \preceq)$ where L is a finite set of levels which contains a distinguished level named all, such that $dom(all) = \{all\}$ and $\preceq$ is a transitive and reflexive relation over the elements of L. The relation $\preceq$ contains a unique bottom level called l_{bottom} and a unique top level called all.

$$L = \{l_{bottom}, ..., l, ..., all \mid \forall\ l,\ l_{bottom} \preceq l \preceq all\}$$

Each level $l \in L$ is associated with a set of values $dom(l)$. For each pair of levels l and l' such that $l \preceq l'$, there exists a roll-up function f which is a partial function so that:

$$f_l^{l'} : dom(l) \longrightarrow dom(l')$$

A **fact table schema** F is defined as follows: $F = (I, M)$ where I is a set of dimension identifiers and M is a set of measures. A **fact table instance** is a tuple where the set of values for each identifier is unique.

To create data cubes, we use the *CUBE* operator [20] which is defined as follows: for a given fact table $F = (I = \{l_1 \in D_1, ..., l_p \in D_p\}, M)$, a set of levels

$GL = \left\{l'_1 \in D_1, ..., l'_p \in D_p \mid l_i \preceq\ l'_i\ \forall i = 1..p\right\}$, and a set of measures m with $m \subset M$, the operation $CUBE(F, GL, m)$ gives a new fact table $F' = (GL, m')$ where m' is the result of aggregation (with roll-up functions $f_{l_1}^{l'_1}, ..., f_{l_p}^{l'_p}$) of the set of measures m from I to GL.

5.2 K-means

K-means is known as a partitional clustering method that allows to classify a given data set X through k clusters fixed a priori [21, 22]. The main idea is to define k centroids, one for each cluster, and then assign each point to one of the k clusters so as to minimize a measure of dispersion within the clusters. The algorithm is composed of the following steps:

1. *Place k initial points into the space represented by the data set X ;*
2. *Assign each object x_i to the group that has the closest centroid c_j (the proximity is often evaluated with the euclidian metric);*
3. *Recalculate the positions of the k centroids when all objects have been assigned ;*
4. *Repeat Steps 2 and 3 until the centroids no longer move.*

The best grouping is the partition of the data set X that minimizes the sum of squares of distances between data and the corresponding cluster centroid.

We chose the K-means method for the following reasons: (1) its result format which is a partition that corresponds to the building process of the aggregation level in a dimension hierarchy, and (2) its low and linear algorithmic complexity which is crucial in the context of OLAP to provide the user with quick results.

5.3 Formalization

The K-means method enables us to classify instances of a level l, either on its own attributes, or on measure attributes in the fact table of the data warehouse. We exploit then the K-means clustering results to create a new level l_{new} and a roll-up function which relates instances of the child level l with the domain of the parent level l_{new}.

Roll-up with Generalize operator. An operator called **Generalize** is proposed in [17]. This operator creates a new level l_{new}, to which a pre-existent level l rolls up. A function f must be defined from the instance set of l, to the domain of l_{new}. We can summarize the formal definition of this operator as follows: given a dimension $D = (L = \{l_{bottom}, ..., l, ..., all\}, \preceq)$, two levels $l \in L$, $l_{new} \notin L$ and a function $f_l^{l_{new}} : instanceSet(l) \longrightarrow dom(l_{new})$. $Generalize(D, l, l_{new}, f_l^{l_{new}})$ is a new dimension $D' = (L', \preceq')$ where
$L' = L \cup \{l_{new}\}$ and $\preceq' = \preceq \cup \{(l \rightarrow l_{new}), (l_{new} \rightarrow All)\}$, according to the roll-up function $f_l^{l_{new}}$.

Example: Consider the dimension `AGENCY` (Figure 3) and the roll-up function: $f^{\texttt{POTENTIAL GROUP}}_{\texttt{AGENCY}}$ = ((Charpennes, Big), ..., (Aubenas, Small), ..., (Lyon La Doua, Average), ...).

Then, Generalize(`AGENCY`, `AGENCY`, `POTENTIAL GROUP`, $f^{\texttt{POTENTIAL GROUP}}_{\texttt{AGENCY}}$) adds a new level called `POTENTIAL GROUP` in the `AGENCY` dimension.

Hence, `AGENCY` → `POTENTIAL GROUP` constitutes another hierarchy for the `AGENCY` dimension.

Roll-up with RoK operator. In our case, the $f_l^{l_{new}}$ function is represented by our "RoK" (*Roll-up with K-means*) operator. Assume a positive integer k, a population $X = \{x_1, x_2, ..., x_n\}$ composed by n instances and a set of k classes $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$. By using the K-means algorithm described in section 5.2, $RoK(X, k)$ calculates the set $C = \{c_1, ..., c_k \mid \forall i = 1..k, c_i = barycenter(\mathcal{C}_i)\}$ and returns the roll-up function:
$f_x^c = \{(x_j \rightarrow \mathcal{C}_i) \mid \forall j = 1..n \; and \; \forall m = 1..k, dist(x_j, c_i) \leq dist(x_j, c_m)\}$

Example: Let $X = \{x_1 = 2, x_2 = 4, x_3 = 6, x_4 = 20, x_5 = 26\}$ and $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2\}$. RoK(X, 2) returns the set $C = \{c_1 = 4, c_2 = 23\}$ with the roll-up function
$f_x^c = \{(x_1 \rightarrow \mathcal{C}_1), (x_2 \rightarrow \mathcal{C}_1), (x_3 \rightarrow \mathcal{C}_1), (x_4 \rightarrow \mathcal{C}_2), (x_5 \rightarrow \mathcal{C}_2)\}$

Discussion. Comparing with the Generalize operator, our RoK operator generates automatically the new roll-up function. Our RoK operator is then more than a conceptual operator and provides a way to deal not only with the structure of the hierarchy, but also with the data of this hierarchy.

5.4 Algorithm

We present in the following the input parameters and the different steps of the personalization algorithm for the recommendation system.

- A dimension $D = (L, \preceq)$, a level $l \in L$, a set of measure $m \in M$ (if required),
- A level name $l_{new} \notin L$,
- A positive integer $k \geq 2$ which will be the modality number of l_{new},
- A variable *dataSource* that can take two values: 'F' (for *fact)* or 'D' (for *dimension).*

Step 1. Construction of the learning set X_l: This first step generates a learning set X_l from the instances of the pre-existing analysis level l. We consider a variable called *dataSource*. If the value of the variable equals to 'D', the population X_l is described by a part of attributes of the dimension D chosen by the user. Otherwise, X_l is generated by executing the operation $CUBE(F, l, m)$ whose parameters are also fixed by the user.

Example: Let us consider the two examples presented previously about the creation of the POTENTIAL GROUP and the COST GROUP levels.

Let us consider that one user needs to create a new level POTENTIAL GROUP from the AGENCY level. If the *dataSource* parameter equals to 'D', each agency will be described by a part of its descriptors in the data warehouse chosen by the user. For instance, the user can choose the CustNumber and the Population attributes for the reasons presented before (Figure 4a).

Now, let us suppose that the user needs to create a new level COST GROUP from the CUSTOMER level. If the *dataSource* parameter equals to 'F', our algorithm performs the operator CUBE (ACCOUNTS MANAGEMENT, CUSTOMER, COST) according to the choice of the user. Thus, we obtain the learning set described in Figure 4b.

AgencyName	CustNumber	Population
Charpennes	105	12 000
Aubenas	6	100
Lyon La Doua	60	6 000
Annonay	8	180

a)

CUSTOMERid	COST
001	400
004	20
005	300
007	50

b)

Fig. 4. a) The `AGENCY` analysis level described by a part of its own attributes. b) The `CUSTOMER` analysis level described by a measure.

Step 2. Clustering: During this step, the algorithm applies the *RoK* operator to the learning set X_l. If, for example, the parameter k equals to 2, the operation *RoK* on the Figure 4a gives the set $\mathcal{C} = \{\mathcal{C}_1(82.5; 9000), \mathcal{C}_2(7; 140)\}$ and the roll-up function:

$f^{\texttt{POTENTIAL GROUP}}_{\texttt{AGENCY}}$ =((Charpennes; $\mathcal{C}_1$), (Aubenas; $\mathcal{C}_2$),
(Lyon La Doua; $\mathcal{C}_1$),(Annonay; $\mathcal{C}_2$)).

Step 3. Creation of the new level: This step implements the new analysis level l_{new} in the data warehouse model. It is done after the validation of the user. To do this operation, our algorithm performs a *Generalize* operation on the dimension D, from the level l by using the roll-up function $f_l^{l_{new}}$ generated during the previous step.

Example: To materialize the `POTENTIAL GROUP` level in the `AGENCY` dimension, our algorithm performs the operator Generalize:
Generalize(`AGENCY`, `AGENCY`, `POTENTIAL GROUP`, $f^{\texttt{POTENTIAL GROUP}}_{\texttt{AGENCY}}$).

5.5 Feature selection

To apply the K-means clustering method onto the data warehouse, we propose two strategies for the feature selection. The first one uses directly attributes

that describe the level l to be classified while the second one uses measure attributes on the fact table aggregated over the level l. We are going to illustrate these two proposals with examples extracted from the LCL case study presented previously.

Proposal 1: *K-means based on the dimension level features.* Let us consider the next analysis objective: *Is it necessary to close agencies which make little income? And is it necessary to open new agencies in places which make a lot of income?*

To try to answer these questions, the user is going to study the `NBI` through the `AGENCY` dimension (Figure 3). To improve his/her analysis, the user can feel the need to aggregate agencies according to their potential. For that purpose, our operator allows the user to classify instances of the `AGENCY` level according to the population of the city where the agency is located in and the customer number of the agency. The objective is to create a new level which groups the instances of the `AGENCY` level in small, average or big potential (Figure 5).

Proposal 2: *K-means based on data fact measures.* Assume that the analysis objective of the user is to identify a customer grouping according to the costs. The idea is that a customer can cost much compared to an average cost but also bring much more than an average and vice versa. Thus it would be interesting to analyse the `NBI` according to groups of customer costs. With our proposal, the user can concretize this need with a new level in the `CUSTOMER` dimension. For that, our operator will summarize `COST` measure on the `CUSTOMER` level of the dimension. K-means is then performed to the result of this aggregation operation. After this clustering, the creation of the new level allows analysis according to groups of costs (Figure 5).

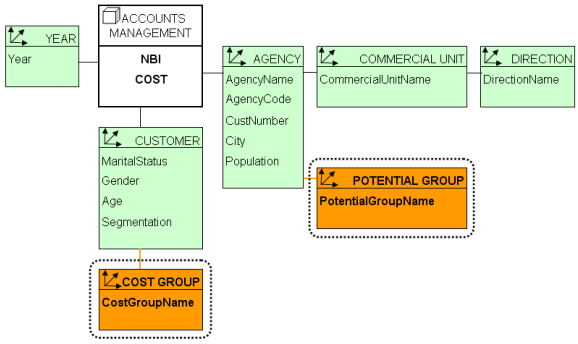

Fig. 5. LCL data warehouse model after addition of "`COST GROUP`" and "`POTENTIAL GROUP`" levels.

5.6 Data Warehouses Model Evolution for OLAP Personalization

Before the effective creation of the level, a validation phase by the user is required, since we are in a context of recommendation. The validation is given by the user only if the proposed level is an answer to his/her analysis needs. Note that, this personalization process provides the user with expressing his/her needs in terms of giving the value of the number of classes he/she wants and specifying the attributes involved in the K-means process.

Creating new granularity levels does not affect the integrity of existing data. The data warehouse is updated, allowing to share the new analysis possibilities with all decision makers, without requiring versions management.

6 Implementation and Experiments

We developped our approach inside the Oracle 10g RDBMS. Thus, we implemented the k-prototypes algorithm by using PL/SQL stored procedures. K-prototypes is a variant of the K-means method allowing large datasets clustering with mixed numeric and categorical values [23]. In our implementation, datasets are stored within a relational table. After the clustering process, the model evolution is performed by using SQL operators: the new level is created with the *CREATE TABLE* command and the roll-up function is established with a primary key/foreign key association between the new and the existing levels.

We carried out some experiments under the "Emode" data warehouse. Emode is an e-trade data warehouse which is used as a demonstration database for the tool *"BusinessObject 5.1.6"*. We standardized the schema of this data warehouse compared to the diagram of Figure 6.

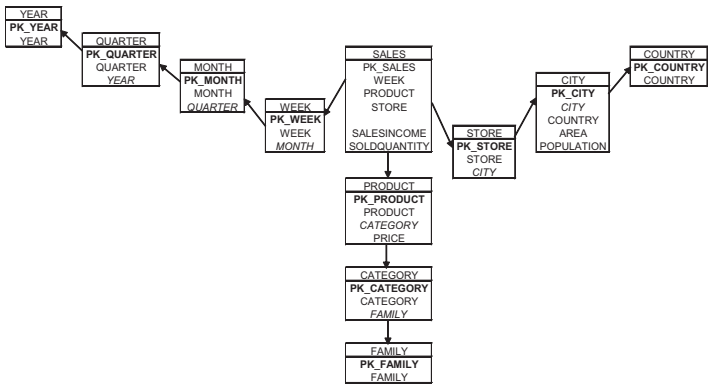

Fig. 6. Schema of the "Emode" data warehouse.

The *sales* fact table stores 89200 records and the *article* level of the *product* dimension contains 213 instances. According to our two proposals for "feature selection", we envisaged two scenarios:

1. Creation of an *article price grouping* level which classifies the 213 articles according to their price,
2. Creation of another level *article sales grouping* which groups the articles according to the sales income.

Figure 7 shows the results of the two scenarios.

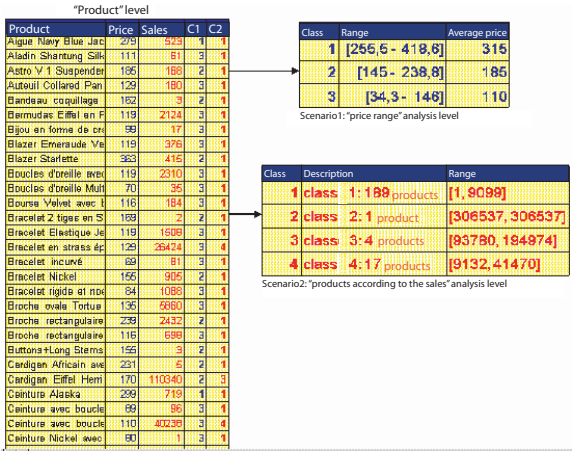

Fig. 7. Results of the two scenarios.

We created the *article price grouping* level with three possible values. With this level, we can analyse the influence of the prices on sales. Figure 8 shows the quantity sold for the 3 price categories. For instance, we can conclude that the products of the lower price (category 3) are those that are sold in larger quantities.

For the *article sales grouping* level, we obtain a level allowing to gather articles into four classes of sales income. Figure 8 shows the quantity sold according to the sales income information. Thus we can for instance affirm that the products that are the subject of the best sales (category 2) are not sold in the lowest quantities. Such a created level allows to confirm or deny the 80-20 rule.

We mention that a drill down allows to know more about the products composing the various created classes.

7 Conclusion and Perspectives

In this paper, we proposed a general framework to integrate knowlegde inside a data warehouse in order to allow OLAP personalization. Our personalization approach is supported by the data warehouse model evolution, independently of the data sources, and it provides to the users new analysis possibilities.

We exploit two types of knowledge: explicit knowledge which is directly expressed by users and implicit knowledge which is extracted from the data. In the first case, the system adapts itself by creating a new granularity level according

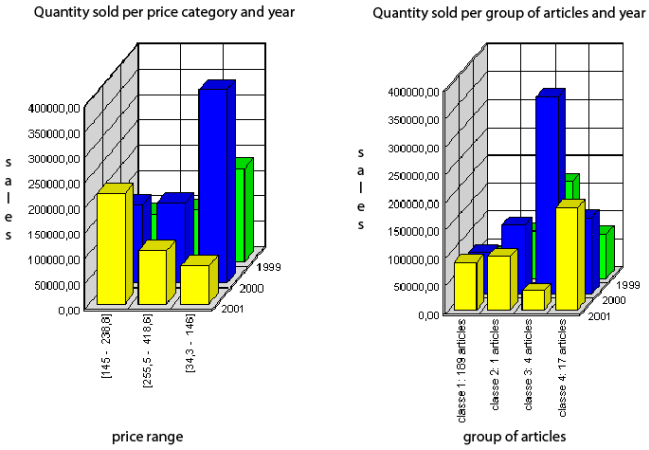

Fig. 8. Analysis results of the two scenarios.

to the user's needs. In the second case, the system recommends to the user a new analysis axis based on automatically extracted clusters from the data warehouse. If the user validates the proposition, a new granularity level is creatad in the dimension hierarchy.

Our recommendation system is based on a definition of a new OLAP operator, called RoK based on a combination between the K-means clustering method and a classical roll-up operator. RoK operator computes significant and more elaborated OLAP queries than the classical ones.

To validate our approach for OLAP personalization, we developed a prototype within the Oracle 10g RDBMS and carried out some experiments which showed the relevance of our personalized data warehouse system. We mainly implemented the RoK operator in the form of a stored procedure using PL/SQL language.

This work opens several promising issues and presents new challenges in the domain of personalization in data warehouses. Firstly, instead of recommending to the user to create only one hierarchy level, we plan to generalize our recommendation approach to be able to recommend a fully dimension hierarchy by using for example the Agglomerative Hierarchical Clustering. The user will be asked to choose a number of classes after the learning process. In this case, the number of classes is not an input parameter. Secondly, we plan to refine the recommendation process. A promising issue consists in combining data mining and the concept of user profile for personalization. Hence, we suggest to consider users' analysis sessions which are composed of a set of queries. For each user profile, our key idea is to use frequent itemset mining methods for extracting the frequently asked queries. These latter are recommended to the user according to his/her profile. Finally, as a consequence of our data warehouse personalization and evolution approach, it is interesting to evaluate the performance of material-

ized views maintenance. In other words, once a new level is created, how existing materialized views are updated and what is the cost?

References

1. Rizzi, S.: OLAP Preferences: A Research Agenda. In: DOLAP 07. (2007) 99–100
2. Bentayeb, F., Favre, C., Boussaid, O.: A User-driven Data Warehouse Evolution Approach for Concurrent Personalized Analysis Needs. Journal of Integrated Computer-Aided Engineering **15**(1) (2008) 21–36
3. Domshlak, C., Joachims, T.: Efficient and Non-Parametric Reasoning over User Preferences. User Modeling and User-Adapted Interaction **17**(1-2) (2007) 41–69
4. Korfhage, R.R.: Information storage and retrieval. John Wiley & Sons, Inc. (1997)
5. Manber, U., Patel, A., Robison, J.: Experience with personalization of yahoo! Communications of the ACM **43**(8) (2000) 35–39
6. Pretschner, A., Gauch, S.: Ontology Based Personalized Search. In: ICTAI 99, Chicago, Illinois, USA. (1999) 391–398
7. Cherniack, M., Galvez, E.F., Franklin, M.J., Zdonik, S.B.: Profile-Driven Cache Management. In: ICDE 03, Bangalore, India. (2003) 645–656
8. Chomicki, J.: Preference Formulas in Relational Queries. ACM Transactions on Database Systems **28**(4) (2003) 427–466
9. Koutrika, G., Ioannidis, Y.: Personalized Queries under a Generalized Preference Model. In: ICDE 05, Tokyo, Japan. (2005) 841–852
10. Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., Laurent, D.: A Personalization Framework for OLAP Queries. In: DOLAP 05. (2005) 9–18
11. Ravat, F., Teste, O.: Personalization and OLAP Databases. Annals of Information Systems, New Trends in Data Warehousing and Data Analysis (2008)
12. Jerbi, H., Ravat, F., Teste, O., Zurfluh, G.: Management of context-aware preferences in multidimensional databases. In: ICDIM 08. (2008) 669–675
13. Giacometti, A., Marcel, P., Negre, E.: A Framework for Recommending OLAP Queries. In: DOLAP 08. (2008) 73–80
14. BenMessaoud, R., Boussaid, O., Rabaseda, S.: A new OLAP aggregation based on the AHC technique. In: DOLAP 04. (2004) 65–72
15. Kaya, M., Alhajj, R.: Extending OLAP with Fuzziness for Effective Mining of Fuzzy Multidimensional Weighted Association Rules. In: ADMA 06. (2006) 64–71
16. Blaschka, M., Sapia, C., Hofling, G.: On Schema Evolution in Multidimensional Databases. In: DaWaK 99. (1999) 153–164
17. Hurtado, C., Mendelzon, A., Vaisman, A.: Maintaining Data Cubes under Dimension Updates. In: ICDE 99. (1999) 346–355
18. Morzy, T., Wrembel, R.: Modeling a Multiversion Data Warehouse: A Formal Approach. In: ICEIS 03. Volume 1. (2003) 120–127
19. Vaisman, A., Mendelzon, A.: Temporal Queries in OLAP. In: VLDB 00. (2000) 242–253
20. Gray, J., Bosworth, A., Layman, A., Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. In: ICDE 96. (1996) 152–159
21. Forgy, E.: Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification. In: Biometrics. Number 21
22. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Vth Berkeley Symposium. (1967) 281–297
23. Huang, Z.: Clustering Large Data Sets with Mixed Numeric and Categorical Values. In: PAKDD 97. (1997)