K-MEANS BASED APPROACH FOR OLAP DIMENSION UPDATES

Fadila Bentayeb
ERIC, University of Lyon Lumière Lyon2, 5 avenue Pierre Mendès-France
69676 Bron Cedex, France

Keywords: OLAP, data warehouse, schema evolution, clustering, k-means, analysis level, dimension hierarchy.

Abstract: Actual data warehouses models usually consider OLAP dimensions as static entities. However, in practice, structural changes of dimensions schema are often necessary to adapt the multidimensional database to changing requirements. This paper presents a new structural update operator for OLAP dimensions, named *RollupWithKmeans* based on k-means clustering method. This operator allows to create a new level to which, a pre-existent level in an OLAP dimension hierarchy rolls up. To define the domain of the new level and the aggregation function from an existing level to the new level, our operator classifies all instances of an existing level into k clusters with the k-means clustering algorithm. To choose features for k-means clustering, we propose two solutions. The first solution uses descriptors of the pre-existent level in its dimension table while the second one proposes to describe the new level by measures attributes in the fact table. Moreover, we carried out some experimentations within Oracle 10 g DBMS which validated the relevance of our approach.

1 INTRODUCTION

A data warehouse is a consolidated view of enterprise data, used to analyze facts on various dimensions. The granularity levels of each dimension are fixed during the design of the data warehouse system. After deployment, these dimensions remain static because schema evolution is poorly supported in current OLAP (On-Line Analytical Processing) models. Thus, some new requirements are not satisfied and some trends are not explored.

To consider this problem, two categories of research emerged. There are works which propose a temporal multidimensional data model (Vaisman and Mendelzon, 2000; Bliujute et al., 1998; Morzy and Wrembel, ; Morzy and Wrembel, 2004). These works manage and keep the evolutions history by timestamping relations over levels (Vaisman and Mendelzon, 2000), data warehouse versions (Morzy and Wrembel,) or data themselves (Bliujute et al., 1998). Other works propose to extend the multidimensional algebra with a set of schema evolution operators (Blaschka et al., 1999; Hurtado et al., 1999; Pourabbas and Rafanelli, 1999).

In this paper, we propose a schema evolution operator allowing relevant structural updates on dimension hierarchies, named *RollupWithKmeans*. Given a hierarchical level l_n, our operator *RollupWithKmeans* classifies its instances by using k-means clustering algorithm. A new hierarchical level l_{new} is then created by applying a rollup function which relates the instances of level l_n with the instances of level l_{new} (the domain of the new level is composed of k instances representing the k obtained clusters). The choice of k-means method is justified by its low algorithmic complexity (linear) and by the format of its results (a partition). With regard to the feature selection, we propose two heuristics. The first heuristic uses directly attributes that describe the level l_n to be classified while the second heuristic uses measures attributes on the fact table aggregated over the level l_n. By using the first heuristic, *RollupWithKmeans* gives the natural classification of a dimension level while the second heuristic shows facts trends compared to a dimension level. To validate our approach and show its relevance, we performed experimentations within Oracle 10 g DBMS (DataBase Management System). The remainder of this paper is organized as follows. Section 2 presents related work about schema evolution on data warehouses. Section 3 details our k-means-based approach for schema evolution in data warehouses. Section 4 presents the experimentations we performed to validate our approach. Section 5 concludes our paper and presents some perspectives.

Bentayeb F. (2008).
K-MEANS BASED APPROACH FOR OLAP DIMENSION UPDATES.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - DISI*, pages 531-534
DOI: 10.5220/0001717905310534

2 RELATED WORK

In the context of data warehouses evolution, two categories of research emerged: the first one recommends extending the multidimensional algebra with a set of schema evolution operators while the second proposes temporal multidimensional data models.

Schema Evolution Operators. Hurtado et al. proposed a formal model of dimension updates in a multidimensional model, covering updates to the domains of the dimensions and structural updates to the dimension hierarchies with a collection of primitive operators to perform these updates (Hurtado et al., ; Hurtado et al., 1999). For example, they propose the *generalize* operator which creates a new level l_{new}, to which a pre-existent one, l_n, rolls up. Blaschka et al. improves works of Hurtado et al. by proposing a set of operators independent of every logical and physical model of the data warehouse (Blaschka et al., 1999).

Temporal Multidimensional Models. In temporal multidimensional database, the idea is to keep evolution history by using timestamps. Thus, Vaisman et al. proposed the TOLAP *(Temporal OLAP) (Vaisman and Mendelzon, 2000)*. In TOLAP, a dimension is designed with a DAG *(directed acyclic graph)* where a node represents a level and an edge represents a relation between two adjacent levels. In the TOLAP graph, edges are stamped with a time interval representing the validity period of the aggregation link. A similar approach is proposed by Bliujute et al. with the "Temporal Star Schema" (Bliujute et al., 1998). Morzy et al. proposed a multiversion data warehouse (Morzy and Wrembel, ; Morzy and Wrembel, 2004). With this versioning approach, a new version of the data warehouse is physically created when changes occur. These timestamps are then used to identify the good versions which will satisfy each analysis request.

3 K-MEANS BASED APPROACH FOR DIMENSION UPDATES

3.1 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 (Forgy, 1965; Bradley and Fayyad, 1998; Likas et al., 2003). 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.

Among existing clustering methods, we chose k-means for its low and linear algorithmic complexity and for its result format (a partition). Indeed, we think that these two characteristics are important for OLAP analysis and dimension updates in data warehouses.

3.2 Illustrative Example

Let us consider a sales data warehouse (figure 1). This data warehouse contains two measures: **sales income** and **sold quantity**. These measures can be studied on three dimensions: ***"Time", "Product"*** and ***"Region"***. The hierarchy of the *Region* dimension has three levels: *store, city* and *country*. In the same way, the *Product* dimension consists of three levels: *product, product category* and *product family*. In addition, *Time* dimension is organized following four levels: *week, month, quarter* and *year*.

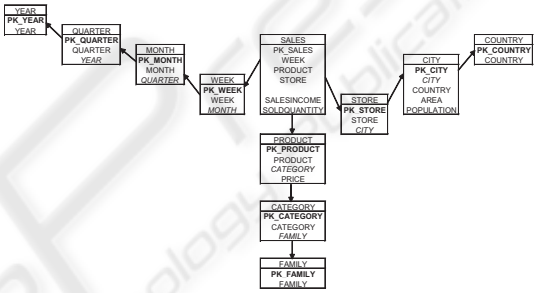

Figure 1: Schema of the sales data warehouse.

3.3 Principle of Our Approach

In our approach, we are distinguished from the existing ones by the use of data mining techniques to perform data warehouse schema evolution. Generally, to carry out OLAP analyses, the user generates a data cube by selecting dimension level(s) and measure(s) which will satisfy its needs. Then, the user explores the obtained cube to detect similarities in facts and dimension instances. For that, he exploits the different levels within a dimension. To help him in this step, we propose a schema evolution operator *RollupWithKmeans* allowing to create a new hierarchy level by using a clustering algorithm. Our idea is to add a new level, l_{new}, to which a pre-existent one, l_n, rolls up. To achieve our objective, our operator classifies initially the instances of the level l_n by using the k-means clustering algorithm. The operator *RollupWithKmeans* creates then the new level l_{new} composed of the k instances corresponding to the k obtained clusters. Finally, *RollupWithKmeans* defines a rollup function between level l_n and level l_{new} by relating the instances of the levels l_n and l_{new} according to the k-means clustering result. The originality of our schema evolution approach is that our rollup

function $f_{l_n}^{l_{new}}$ is generated automatically by using k-means clustering method.

3.4 Feature Selection

To choose features on which k-means will classify the instances set of the level l_n, we consider two strategies to explore a data cube efficiently.

1. Dimension attributes features. For example, to answer the question "*Is it necessary to close shops which make few sales?*", we study the sales incomes through the *Region* dimension. To improve analyses, we may feel the need to aggregate cities according to their size. Hence, we create a new level which groups the instances of the *city* level in small, average or big city (figure 2).
2. Measures attributes features. Assume that the analysis objective of the user is to find a product grouping according to the sales. We summarize then *sales income* and *sold quantity* measures on the *product* level of the *Product* dimension and perform the k-means onto the obtained aggregates. The new level is then created (figure 2).

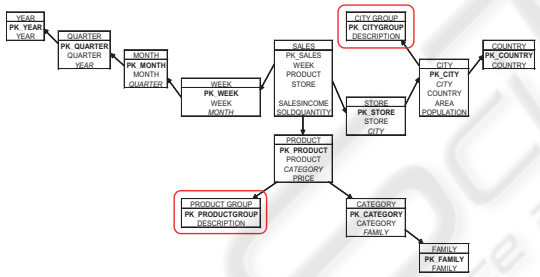

Figure 2: Sales data warehouse with additional "Product Group" and "City Group" levels.

3.5 Algorithm

Inputs: A dimension $D = (L, \preceq)$, a level $l_n \in L$, a level $l_{new} \notin L$, a positive integer $k \geq 2$ representing the modality number of l_{new}, and a variable *dataSource* that can take two values: F (for *fact*) or D (for *dimension*).

Step 1: Generating a learning set X_{l_n} from the instances of the pre-existent analysis level l_n. If the value of the *dataSource* variable equals to D, the population X_{l_n} is described directly by the attributes of the dimension D. Otherwise, X_{l_n} is generated by executing the operation $CUBE(F, l)$.

Step 2: Clustering. During this step, the algorithm applies the k-means clustering method on the learning set X_{l_n}.

Step 3: Creation of the new level. This step materializes the new hierarchy level l_{new} in the data warehouse schema by using the rollup function $f_{l_n}^{l_{new}}$ generated during the previous step.

4 IMPLEMENTATION AND EXPERIMENTATIONS

We implemented the k-prototypes algorithm by using PL/SQL stored procedures inside the Oracle 10g DBMS. K-prototypes is a variant of the k-means method allowing large datasets clustering with mixed numeric and categorical values (Huang, 1997). In our implementation, datasets are stored within a relational table. After clustering process, schema evolution is then performed by using SQL operators: the new level is created with the CREATE TABLE command and the rollup function is established with a primary key/foreign key association between the new and the existing levels.

We carried out some experimentations under the emode data warehouse. Emode is an e-trade data warehouse which is used as demonstration database for the tool *"BusinessObject 5.1.6"*. We standardized the schema of this data warehouse to obtain the diagram of figure 2. Thus, the *sales* fact table stores 89200 records and the *product* level of the *Product* dimension contains 213 instances. According to our two heuristics of "feature selection", we defined two scenarii:

1. Creation of a *product price grouping* level which classifies the 213 articles according to their price into 3 clusters. Hence, we can analyze the influence of the prices on sales (Figure 4).
2. Creation of another level *product sales grouping* which gathers the products according to sales into 4 clusters. Hence, we can analyze the sales trend according to sales information (Figure 5).

The figure 3 shows the results of the two scenarii.

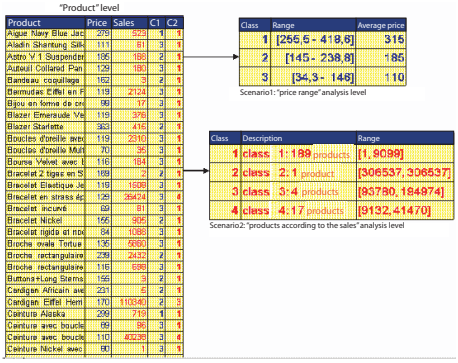

Figure 3: Results of the two scenarii.

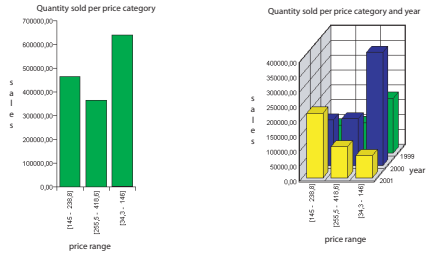

Figure 4: Sales trend according to the "product price grouping" level.

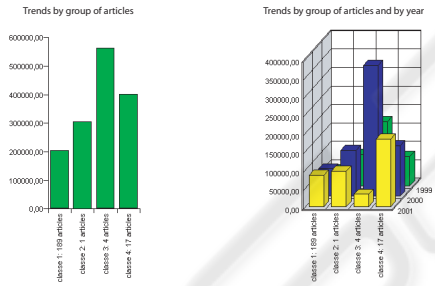

Figure 5: Sales trend according to the "product sales grouping" level.

5 CONCLUSIONS

In this paper, we proposed an original approach which consists in using data mining techniques as aggregation operators to update dimension hierarchies in data warehouses. Indeed, in certain cases, one can need to define other semantic aggregates than those defined in the design step of the data warehouse. We defined then a new structural update operator, named *RollupWithKmeans* for OLAP dimensions based on both the k-means clustering method and the hierarchical relationship which links a child member to a parent member in a dimension hierarchy. Our operator *RollupWithKmeans* applied the k-means method to extract semantic relations from either dimension descriptors or measures to enrich hierarchies by creating new aggregation levels. Decision makers will thus be able to achieve their information needs for analyses. To validate our approach, we carried out some experimentations which showed the relevance of our approach.

To generalize our approach, we plan to use the Agglomerative Hierarchical Clustering (AHC) method to create not only a new hierarchy level but a complete dimension hierarchy. Indeed, AHC method creates a hierarchy of partitions under the form of a tree which coincides with the format of the dimension hierarchy in data warehouses models.

REFERENCES

Blaschka, M., Sapia, C., and Höfling, G. (1999). On schema evolution in multidimensional databases. In *DaWaK 1999*, pages 153–164.

Bliujute, R., Saltennis, S., Slivinskas, G., and Jensen, C. (1998). Systematic change management in dimensional data warehousing. Technical report, University of Arizona.

Bradley, P. and Fayyad, U. (1998). Refining initial points for k-means clustering. In *ICML*, pages 91–99.

Forgy, E. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classification. In *Biometrics num 21*, pages 768–780.

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. In *First Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

Hurtado, C., Mendelzon, A., and Vaisman, A. Maintaining data cubes under dimension updates.

Hurtado, C., Mendelzon, A., and Vaisman, A. (1999). Updating olap dimensions. In *DOLAP 1999*, pages 60–66.

Likas, A., Vlassis, N., and Verbeek, J. (2003). The global k-means clustering algorithm. *Pattern Recognition Letters 36(2)*, pages 451–461.

Morzy, T. and Wrembel, R. Modeling a multiversion data warehouse: A formal approach.

Morzy, T. and Wrembel, R. (2004). On querying versions of multiversion data warehouse. In *DOLAP 2004*, pages 92–101.

Pourabbas, E. and Rafanelli, M. (1999). Characterization of hierarchies and some operators in olap environment. In *DOLAP 1999*, pages 54–59.

Vaisman, A. and Mendelzon, A. (2000). Temporal queries in olap. In *VLDB 2000*.