Adapting OLAP Analysis to User's Constraints through Semantic Hierarchies

Fadila Bentayeb and Rym Khemiri
ERIC Laboratory, University of Lyon, Lumière Lyon 2
5 av. P. Mendès-France, 69676 Bron Cedex, France

Keywords: Analysis Level, Constrained K-means Clustering, OLAP, Personalization, PRoCK, Semantic Dimension Hierarchy, User Constraints.

Abstract: The objective of this paper is to provide a personalized on-line aggregate operator, namely PRoCK (Personalized Rollup operator with Constrained K-means), based on data mining techniques. The use of data mining techniques, and more precisely constrained K-means clustering method, helps to discover new grouping sets with respect to users requirements. In the context of data warehouses, PRoCK allows to adapt dimension hierarchies to the user constraints. Indeed, applied on a given dimension hierarchy instances, constrained k-means clustering method gives a new natural classification. The obtained clustering results constitute a new hierarchy level semantically richer, namely personalized level on which user may elaborate more sophisticated OLAP analysis. PRoCK is integrated inside Oracle RDBMS (Relational DataBase Management System) and we have carried out some experimentation which validated the relevance of our operator.

1 INTRODUCTION

Dimension hierarchies represent a substantial part of the data warehouse model (Pedersen and Jensen, 2001). Indeed, hierarchies allow decision makers to examine data at different levels of detail with OLAP operators such as drill-down and roll-up. Furthermore, 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.

On the other hand, even though data warehouses and OLAP are considered to be user-centric systems, there is obviously a lack of involvement of the user in the system. In fact, only a few analysis possibilities are known at the design stage of a data warehouse according to the identified global analysis needs of the users. Although, business requirements often change over time at the client level where some specific constraints must be satisfied. In this case, the data warehouse must be user-centric to cope with user analysis requirements.

Therefore, to improve decision support systems and to give increasingly relevant information to the user, the need to integrate user requirements into the data warehouse is becoming unavoidable.

Unfortunately, OLAP does not provide automatic tools for structuring analysis axes. We thus base our research on data mining techniques that make possible integrating knowledge into OLAP process to create new relevant analysis axes by exploiting the data warehouse content. We show that combining OLAP technology with data mining techniques can provide more elaborated and more relevant analysis.

The objective of this paper is to adapt OLAP analysis to users by enriching existing hierarchies with derived semantic hierarchies. Indeed, one can need to define other semantic aggregates than those defined in the design step of the data warehouse. For this end, we propose a personalized on-line aggregate operator called PRoCK (Personalized Rollup operator with Constrained K-means). This operator creates automatically new roll-up functions based on user preferences and using the cop k-means clustering algorithm. The user preferences are defined by means of constraints which are specified in the form of must-link and cannot-link constraints (Wagstaff et al., 2001).

To achieve our objective, our idea consists in online personalization process of the data warehouse schema which follows these steps. Given a hierarchical level *l*, PRoCK classifies its instances by using the Cop K-means method clustering algorithm based on the user constraints. A new hierarchical level *pl* is

Bentayeb F. and Khemiri R..
Adapting OLAP Analysis to User's Constraints through Semantic Hierarchies.
DOI: 10.5220/0004444901930200
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 193-200
ISBN: 978-989-8565-59-4

then created by applying a rollup function which relates the instances of the level l with the instances of the level pl. The domain of the personalized level pl is composed of the k instances representing the k obtained clusters. The obtained personalized semantic hierarchies would provide new multidimensional ways for analyzing data and obtain more relevant analyses semantically richer.

Our operator is integrated inside Oracle DBMS where we carried out some experimentation which validated the relevance of our approach.

The remainder of this paper is organized as follows: Section 2 presents related works and compares our approach to existing ones in the literature. In section 3, we present our PRoCK operator. The framework of creating semantic hierarchies using PRoCK is described in section 4. After an illustrative example presented in section 5, we describe implementation with some preliminary experimental results in section 6. Finally, conclusions and our expected future work are given in section 7.

2 RELATED WORKS

Since data warehouses are characterized by voluminous data and are based on a user-centric analysis process, including personalization into the data warehousing process becomes a new research issue (Rizzi, 2007). Despite first approaches for personalization on data warehouses that focus on user definition with specific data as defined on traditional databases, there exists some approaches based on conceptual model and its multidimensional concepts (fact, dimension, hierarchy, measure, attribute). For example, using annotations, a new personalization technique based on user preferences model is proposed in which weights are associated to multidimensional databases components (Ravat and Teste, 2008). To assign priority weights to attributes of a multidimensional schema, the personalization rules are described using the Condition-Action formalism. More recently, this model has been used for handling the context notion in order to closely relating user requirements to their current context (Jerbi et al., 2009).

Moreover, the importance of dimension hierarchies was reflected in (Bentayeb, 2008) where the author used data mining techniques as aggregation operators to update dimension hierarchies in data warehouses without taking into account user preferences.

Garrigos et al. use the data warehouse multidimensional model, user model and rules for the data warehouse personalization (Garrigós et al., 2009). As a result, a data warehouse user is able to work with a personalized OLAP schema, which best matches his needs. Based on ECA-rules (Event-Condition-Action) (Thalhammer et al., 2001)), PRML (Personalization Rules Modeling Language is used in (Garrigós et al., 2009) for specification of OLAP personalization rules. The structure of such PRML rules can be presented with following statement: *when event do if condition then action endIf endWhen*.

After that, in (Kozmina and Niedrite, 2010), a new method was proposed which provides exhaustive description of interaction between user and data warehouse. A set of user-describing profiles (user preference, temporal, spatial, preferential and recommendational) have been developed. A metamodel which formulates user preferences for OLAP schema elements and aggregate functions has been proposed. This model reflects connections among user-describing profiles.

Recently, inspired by (Kießling and Köstler, 2002) and (Golfarelli and Rizzi, 2009), (Golfarelli et al., 2011) propose an approach to adapt preference constructors to multidimensional context. Formulated on schema, preferences can not only be expressed over attributes and thus over cuboids but also preferences can be expressed over numerical values (measures). The preferences composition is modeled using predicate logic attributes and expressed through Pareto composition (two preferences are equally relevant) or Prioritization (a preference is more relevant than another).

We argue that multidimensional structures such as dimension hierarchies have a strong impact in OLAP analysis and they should be considered in OLAP personalization. For this reason, users must be able to express their preferences on dimension hierarchies. In fact, preference model is considered a main open problem in OLAP personalization in (Rizzi, 2007).

Our proposal comes close to a previous work that proposes structural update of OLAP dimensions (Bentayeb, 2008). However, it is different, so that, it proposes personalizing hierarchies by exploiting user preferences. Our method aims at improving OLAP analysis process by taking into account the individual interests of users.

In this section we have reviewed the current approaches for personalization in data warehouses. We present a comparative table (table 1) confronting the panoply of the proposed approaches. We choose some criteria that we consider relevant to compare personalization approaches.

- *Source*: this criterion presents the object to exploit for personalization which can be a user profile (interests, preferences, constraints,...), query history (log file) or user context.

- *Personalization Time*: the time of personalization: before querying, while querying or after querying.
- *Personalization Object*: this criterion presents the object of the proposed method if it is a query, an interface or a content to personalize.
- *Input*: this criterion presents the inputs of the proposed method if it is DW schema or DW instance or both of them.
- *Output*: this criterion presents the outputs of the proposed method if it is a query, a set of tuples or a personalized schema.

3 PERSONALIZED SEMANTIC HIERARCHIES

Our method allows to enrich existing hierarchies with derived semantic hierarchies based on the current user needs (user preferences).

3.1 User Preferences

User profile is the important element for a personalization system. Nevertheless, user profile is reduced to user preferences. In the context of database systems, preference query was introduced for the first time in order to soften "the rigid way in which the researched data characteristics must be specified" (Lacroix and Lavency, 1987). In the case where any object (any record) doesn't reply to these characteristics, it's nevertheless possible in some applications to accept objects having less good characteristics against search criteria. After that, several extensive investigations were carried out and two major lines emerged in the literature for expressing preferences: quantitative and qualitative approaches (Chomicki, 2003). In the qualitative approach, preferences are specified directly, whereas, in the quantitative approach, preferences are expressed indirectly by using scoring functions.

In this paper, we are distinguished from classical definition of preferences by the user constraints. In fact, the user constraints are specified in the form of must-link (two instances must be placed together) and cannot-link (two instances must not be placed together) (Wagstaff et al., 2001). These kind of constraints are explicitly defined by the user.

3.2 Principle

Our personalization method consists in changing online the structure of the data warehouse by creating personalized semantic hierarchies. Then, we enrich existing hierarchies with derived semantic hierarchies to allow the user to get his own personalized analysis. To achieve this purpose, we use data mining techniques, whose parameters are fixed by the user in an interactive way, according to his/her own preferences in terms of aggregation constraints defined by COP K-means. We selected the cop K-means clustering method in order to highlight aggregates semantically richer than those provided by existing hierarchies with respect to user constraints.

In our method, user preferences are represented by user constraints. In fact, users are asked to provide their preferences about the obtained clusters which may form a new granularity level in the considered dimension hierarchy. We use a constrained clustering problem in which the user has some pre-existing knowledge about their desired partitions. Besides the number of clusters k, user can iteratively provide his/her constraints about how items should be grouped in the form of must-link and cannot-link constraints. A must-link constraint enforces that two instances must be placed in the same cluster while a cannot-link constraint enforces that two instances must not be placed in the same cluster. The user constraints refine the clusters towards the desired data.

Our PRoCK operator generates automatically the new roll-up function based on user constraints. Our PRoCK operator exploits user knowledge especially his hard constraints. Therefore, PRoCK provides a way to deal with the structure of the hierarchy and its data with respect to user preferences (user constraints).

To define the domain of the parent level and the aggregation function from a child to the parent level, our operator classifies all instances of a child level into k clusters with the cop k-means clustering algorithm. Therefore, users are asked to choose cop K-means parameters (k + constraints) following their preferences about the obtained clusters which may form a new granularity level in the considered dimension hierarchy.

4 FRAMEWORK FOR CREATING PERSONALIZED SEMANTIC HIERARCHIES

In this section, we present a declarative framework for creating semantic hierarchies that addresses the challenges discussed earlier in the introduction. We show the different definitions of used concepts, the clustering algorithm and the personalization algorithm.

Table 1: Survey of OLAP personalization approaches.

		Bellatreche et al. 2005	Bentayeb 2008	Jerbi et al. 2009, Ravat and Teste 2008	Garrigos et al. 2007	Kozmina et al. 2010	Golfarelli et al. 2009, 2011	Our approach
Source	User profile	×		×	×		×	×
	Query Log					×	×	
	Context			×				
Time (% querying)	Before	×	×	×		×		×
	While				×	×		
	After		×				×	
Object	Query	×	×	×	×	×		×
	Interface	×						
Input	DW schema		×	×	×	×		×
	DW instance	×					×	
Output	Query	×		×				
	Tuples						×	
	Schema		×		×			×

4.1 Basic definitions

Definition 1. Data warehouse. A data warehouse is a multidimensional database that can be defined as $\mu = (\delta, \varphi)$ where δ is a set of dimensions and φ is a set of facts (Hurtado et al., 1999).

Definition 2. Dimension. 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 $dom(all) = \{'all'\}$
- $\preceq$ 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*.
  $L = l_{bottom}, ..., l, ...all | \forall l, l_{bottom} \preceq l \preceq all$

A dimension instance is a tuple (D, f) where D is a dimension schema and f is a set of partial functions between instances of two adjacent hierarchical levels: $f = \{f_1, ..., f_n\}$ such that:
$\forall l, l' \in L \mid l \preceq l', \exists f \mid f_l^{l'} : dom(l) \rightarrow dom(l')$

Definition 3. Fact. A fact schema F is defined as $F = (I, M)$ where I is a set of dimension identifiers and M is a set of measures. A fact instance is a tuple where the set of values for each identifier is unique.

Definition 4. Cube. To create data cubes, we use the CUBE operator (Gray et al., 1996) which is defined as follows: for a given fact $F = (I = \{I_1 \in D_1, ..., D_p \in D_p\}, M)$, a set of levels $GL = \{l'_1 \in D_1, ...l'_p \in D_p | l_i \preceq l'_i \ \forall i = 1...p\}$ and a set of measures m with $m \subset M$, the operation $CUBE(F, GL, m)$ gives a new fact $F' = (GL, m')$ 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.

4.2 Constrained K-means Clustering

Cluster analysis, an important technology in data mining, is an effective method of analyzing and discovering useful information from numerous data. COP K-means algorithm groups the data into classes or clusters with respect to user constraints. COP-K-means is an iterative partitioning algorithm for semi-supervised clustering introduced in (Wagstaff et al., 2001). COP-K-means extends K-means (MacQueen, 1967) by applying constraints based on background knowledge.

Let $\Lambda = \{\lambda_1, ..., \lambda_n\}$ be the given set of instances which must be partitioned such that the number of clusters is not given beforehand. In the context of clustering algorithms, instance-level constraints are a useful way to express a priori knowledge that constrains a placement of instances into clusters. In general, constraints may be derived from partially labeled data or from background knowledge about the domain of real data set. We consider the clustering problem of the data set Λ under the following types of constraints.

- Must-Link constraints denoted by $ML(\lambda_i, \lambda_j)$ indicates that two instances λ_i and λ_j must be in the same cluster.

- Cannot-Link constraints denoted by $CL(\lambda_i, \lambda_j)$ indicates that two instances λ_i and λ_j must not be in the same cluster.
- Transitively derived Instance-Level constraints from:
  - $ML(\lambda_i, \lambda_j)$ and $ML(\lambda_j, \lambda_k)$ imply $ML(\lambda_i, \lambda_k)$,
  - $ML(\lambda_i, \lambda_j)$, $ML(\lambda_k, \lambda_l)$ and $CL(\lambda_i, \lambda_k)$ imply both $CL(\lambda_i, \lambda_l)$ and $CL(\lambda_j, \lambda_k)$.

We selected the COP K-means method because we want to exploit user knowledge especially his hard constraints about their desired partitions. The Constrained K-means Algorithm is as follows:

COP-KMEANS (dataset D, number of clusters k, must-link constraints $Con_= \subset D \times D$, cannot- link constraints $Con_{\neq} \subset D \times D$)

1. Let $C_1...C_k$ be the k initial cluster centers.
2. For each point $d_i \in D$, assign it to the closet cluster C_j such that VIOLATE-CONSTRAINTS $(d_i, C_j, Con_=, Con_{\neq})$ is false. If no such cluster exists, fail (return{}).
3. For each cluster C_i, update its center by averaging all of the points d_j that have been assigned to it.
4. Iterate between (2) and (3) until convergence.
5. Return $\{C_1...C_k\}$.

VIOLATE-CONSTRAINTS(data point d, cluster C, must-link constraints $Con_= \subset D \times D$, cannot-link constraints $Con_{\neq} \subset D \times D$)

1. For each $(d, d_=) \in Con_=$: if $d_= \neq C$, return true.
2. For each $(d, d_{\neq}) \in Con_{\neq}$: if $d_{\neq} \neq C$, return true.
3. Otherwise, return false.

4.3 Formalization

The COP K-means method enables us to classify instances of a level l on its own attributes. We exploit then the COP K-means clustering results to create a new level pl and a roll-up function which relates instances of the child level l with the domain of the parent level pl with respect to user constraints.
Dimension Projection. The operator DimProject operator allows a projection of a dimension D from a hierarchical level l_k. Thus, the Level l_k becomes the finest new hierarchical level of the new dimension D' which summarizes D on the level of detail l_k. Assume a dimension $D = (L, \preceq, f)$ and a hierarchical level $l_k \in L$, $DimProjection(D, l_k)$ is a new dimension $D' = (L', \preceq', f')$ such that:

- $L' = L \{l_a \mid \forall l_a \in L, l_a \preceq l_k\}$,
- $\preceq' = \preceq \{(l_0 \rightarrow l_1), ..., (l_{k-1} \rightarrow l_k)\}$,
- $f' = f \{f^{l_1}_{l_0}, ..., f^{k}_{l_{k-1}}\}$.

Roll-up with Constrained K-means operator. In our case, the f^{pl}_l function is provided by our operator "PRoCK" (Roll-up with Constrained K-means). Assume a positive integer k, a population $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ composed by n instances, a set of k classes $C = \{C_1, ..., C_k\}$ and a set of constraints $Cons = Cons_= \cup Cons_{\neq}$. By using the Cop K-means algorithm described in section 4.2, $RoCK(\Lambda, k, Cons)$ calculates the set $C = \{c_1, ..., c_k | \forall i = 1..k, c_i = barycenter(C_i)\}$ and returns the roll-up function:
$f^c_{\lambda} = \{(\lambda_i \rightarrow C_j) | \forall i = 1..n$ and $\forall m = 1..k$, $dist(\lambda_i, c_j) \leq dist(\lambda_i, c_m)$ and violate-constraints $(\lambda_i, c_j) = False\}$.

Insert a personalized level in a dimension. The operator PRoCK creates a new level pl, to which a pre-existing level l rolls up. A function f must be defined from the instance set of l, to the domain of pl. We can summarize the formal definition of this operator as follows: given a dimension $D = (L = \{l_{bottom}, ..., l, ..., all\}, \preceq)$, two levels $l \in L$ and $pl \notin L$ and a function f^{pl}_l: instanceSet(l)$\longrightarrow$ dom(pl). PRoCK(D, l, pl, f^{pl}_l) is a new dimension $D' = (L', \preceq')$ where
$L' = L \cup \{pl\}$ and $\preceq' = \preceq \cup \{(l \rightarrow pl), (pl \rightarrow all)\}$, according to the roll-up function f^{pl}_l.

Our personalization approach is then original since the new roll-up function is generated automatically. It is 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.

4.4 Algorithm

We present in the following the input parameters and the different steps of the personalization algorithm. The first step of our algorithm consists in generating a learning set Λ_l from the instances of the pre-existing analysis level l. We consider a variable called *dataSource*. If the value of this variable equals to 'D', the population Λ_l is described by a part of attributes of the dimension D chosen by the user. Otherwise, Λ_l is generated by executing the operation $CUBE(F, Gl, m)$ whose parameters are also fixed by the user. Then, the algorithm applies the COP K-means method to the learning set Λ_l with respect to defined constraints $Cons$. It allows to every portioning plan to specify which are pairs having must-link or cannot-link constraints. Finally, our algorithm implements the new analysis level pl in the data warehouse model. It is

done after the validation of the user. To do this operation, our algorithm performs the *PRoCK* operator on the dimension D, from the level l by using the roll-up function f_l^{pl} generated during the previous step.

Algorithm 1: How to create a semantic hierarchy level.

Input:

- A dimension $D = (L, \preceq)$, a level $l \in L$ and a set of measures $m \in M$ (if required)
- A level name $pl \notin L$
- A positive integer $k \geq 2$ which will be the modality number of pl
- Constraints *Cons* in the form of must-link or cannot-link constraints
- A variable *dataSource* that can take be *fact)* or *dimension)*

Output: Personalized hierarchy

$\Lambda \leftarrow \emptyset$
$PersDim \leftarrow \emptyset$
if $datasource =$ *'Dimension'* **then**
  $\Lambda \leftarrow DimProjection(D, l)$
  else
  if $datasource =$ *'Fact'* **then**
    $\Lambda \leftarrow CUBE(F, Gl, m)$;
  end
end
$f_l^{pl} \leftarrow$ COP K-means$(\Lambda_l, k, cons)$
$PersDim \leftarrow PRoCK(D, l, pl, f_l^{pl})$
return *PersDim*

5 ILLUSTRATIVE EXAMPLE

To illustrate our method, we present the analysis of Internet impact which constitute a development indicator. Indeed, Internet is a new vector of development and trade and we can measure the impact of the Internet for each country by measuring the number of Internet users in relation to the population.

Table 2 gives the number of users within a country that access the Internet. This table contains the number of Internet users and population of 9 African countries. Statistics vary from country to country and Nigeria occupies a rather exceptional place as the most populous country in Africa.

Assume the user analysis objective is to know whether the country is developed or not through the impact of Internet use. To find an answer to this question, he will try to explore the use of Internet across "*Country*" dimension whose actual hierarchy is organized as in Figure 1.

[1]https://www.cia.gov/library/publications/the-world-factbook/rankorder/2153rank.html

Table 2: Internet users in Africa.[1]

Country	Users(2009)	Population
Tunisia	3 500 000	10 589 025
Zimbabwe	1 423 000	11 651 858
Ouganda	3 200 000	31 367 972
Morroco	13 213 000	31 671 474
Algeria	4 700 000	36 057 838
Kenya	3 996 000	39 002 772
South of Africa	4 420 000	49 052 489
Egypte	20 136 000	84 474 000
Nigeria	43 989 000	149 283 240

Figure 1: Schema of Country dimension.

For more focused analysis, the user can then feel the need to add a new level of analysis *CountryGroup* which must group countries according to the rate of Internet use. To achieve this goal, our idea consists in extracting knowledge automatically from the data warehouse content to provide possibly relevant clusters of countries. In this case, it would be interesting to directly describe each country by the two following attributes: *population* and *number of Internet users*. Our operator PROCK is then in charge of grouping countries according to this new information and create a new granularity level *countryGroup* for further more elaborated OLAP queries.

However, each user may want a specific clustering of the data. In this case, the best way to find the personalized clustering for each user is to incorporate his/her preferences. As discussed earlier, in our method, the user's preferences are presented as must-link and cannot-link constraints between pairs of data instances. Our operator can then invoke the method cop k-means clustering to group automatically countries. To run the example, we present hereafter three application scenarios.

5.1 Scenario 1

Let Λ = {Tunisia, Kenya, Zimbabwe, Algeria, Ouganda, Morroco, Egypte, Nigeria, South of Africa}. By fixing k = 3 and without applying any constraints, we obtain the clusters C_1, C_2 and C_3 as illustrated in table 3.

5.2 Scenario 2

The user may want to find countries of north Africa Egypt, Morocco, Tunisia and Algeria in the same group. Thus, he can introduce the following constraints:

Table 3: Internet users with clusters.

Country	Users(2009)	Population	Cluster
Tunisia	3 500 000	10 589 025	C_3
Zimbabwe	1 423 000	11 651 858	C_3
Ouganda	3 200 000	31 367 972	C_3
Morroco	13 213 000	31 671 474	C_1
Algeria	4 700 000	36 057 838	C_3
Kenya	3 996 000	39 002 772	C_3
South of Africa	4 420 000	49 052 489	C_3
Egypte	20 136 000	84 474 000	C_1
Nigeria	43 989 000	149 283 240	C_2

- ML(Morroco, Egypte)
- ML(Morroco, Tunisia)
- ML(Morroco, Algeria)

Therefore, Λ = {Tunisia, Kenya, Zimbabwe, Algeria, Ouganda, Morroco, Egypte, Nigeria, South of Africa} and $Cons_{=}$ = {(Morroco, Egypte), (Morroco, Tunisia), (Morroco, Algeria)} and $Cons_{\neq} = \emptyset$.

$PRoCK(\Lambda, 3, Cons)$ returns the set $C = \{c_1 = \{(Egypte, Morroco, Algeria, Nigeria, Tunisia\}, c_2 = \{Ouganda, SouthofAfrica, Kenya\}, c_3 = \{Zimbabwe\}\}$.

Rollup function $f^c_\lambda = \{(Morroco \rightarrow C_1), (Algeria \rightarrow C_1), Nigeria \rightarrow C_1), (Tunisia \rightarrow C_1), (Egypte \rightarrow C_1), (Ouganda \rightarrow C_2), (Kenya \rightarrow C_2), (SouthofAfrica \rightarrow C_2), (Zimbabwe \rightarrow C_3)\}$.

5.3 Scenario 3

We can see that Nigeria is invited in the cluster C_1, which is not real if we know the atypical character of this country. Thus, we can introduce the following constraint: cannot-link (Morroco, Nigeria). By applying these constraints, we can obtain the desired result.

Let $\Lambda = \{Tunisia, Kenya, Zimbabwe, Algeria, Ouganda, Morroco, Egypte, Nigeria, SouthofAfrica\}$, $Cons_{=} = \{(Morroco, Egypte), (Morroco, Tunisia), (Morroco, Algeria)\}$ and $Cons_{\neq} = \{(Morroco, Nigeria)\}$.

$PRoCK(\Lambda, 3, Cons)$ returns the set $C = \{c_1 = \{(Egypte, Morroco, Algeria, Tunisia\}, c_2 = \{Ouganda, SouthofAfrica, Kenya, Zimbabwe\}, c_3 = \{Nigeria\}\}$.

Rollup function $f^{CountryGroup}_{Country} = \{(Morroco \rightarrow C_1), (Algeria \rightarrow C_1), Tunisia \rightarrow C_1), Egypte \rightarrow C_1), (Ouganda \rightarrow C_2), (Zimbabwe \rightarrow C_2), (Kenya \rightarrow C_2), (SouthofAfrica \rightarrow C_2), Nigeria \rightarrow C_3)\}$.

5.4 Discussion

At the end of the classification, our algorithm create a new analysis level *CountryGroup* of *country* dimension as in Figure 2.

To materialize the "*CountryGroup*" new level in the "*Country*" dimension, our algorithm performs the operator PRoCK (*Country*, *Country*, *CountryGroup*, $f^{CountryGroup}_{Country}$).

Therefore, PRoCK provides a way to deal with the structure of the hierarchy and its data with respect to user constraints. In fact, it generates automatically the new roll-up function based on user constraints. Based on this new personalized semantic level, the user can have personalized hierarchy and of course personalized dimension that allow him/her to directly target personalized analyses.

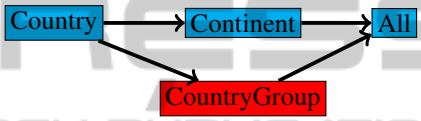

Figure 2: Enriched schema of Country dimension.

6 IMPLEMENTATION AND EXPERIMENTS

We developed our approach within the Oracle 11g 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.

In order to assess the relevance of results of PROCK operator on real data, our tests were conducted with the "Foodmart" data warehouse where sales are represented as a fact table namely "*Sales_fact*" and the axis of analysis are represented as dimension tables namely *Product*, *Promotion*, *Time*, *Store* and *Customer*. Thus, we expected the following test scenario: create an axis of analysis which classifies the 1560 products according to their weight into 3 clusters with respect to user constraints. Let us consider a marketing manager who wants to have products of the same brand together but he wants not to have recyclable package with non recyclable ones. One way to find the best clustering for this user is to incorporate must-link and cannot-link constraints between pairs data instances of *Product* dimension in order to have a personalized analysis level *ProductGroup*. The results of our test are in Figure 3.

As a consequence, the marketing manager may have personalized analysis possibilities over the se-

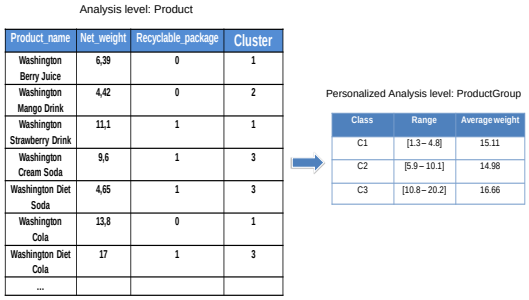

Figure 3: Test results.

mantic hierarchy of the dimension *Product* and especially on the personalized level *ProductGroup*.

7 CONCLUSIONS AND FUTURE WORKS

In this paper, we defined a personalized aggregation operator PRoCK which allows to change on-line the data warehouse structure by enriching existing hierarchies with derived semantic hierarchies. Thus, user may have new analysis possibilities over the semantic hierarchies especially the new aggregation levels. To define the domain of the new level and the aggregation function from an existing level to the personalized level, our operator PROCK classifies all instances of an existing level into k clusters according to user constraints with the Cop k-means clustering algorithm.

Finally, let us point out that as such operator mature, there are additional issues of research that need to be pursued. To provide users with only relevant data from the huge amount of available information, personalization systems use preferences to allow users to express their interest on specific data. Most often, user preferences vary depending on the circumstances. For instance, decision maker requirements can change from a context to another. As a consequence, currently, we think of supporting constraints that depend on user context.

REFERENCES

Bentayeb, F. (2008). K-means based approach for olap dimension updates. In *ICEIS (1)*, pages 531–534.

Chomicki, J. (2003). Preference formulas in relational queries. *ACM Trans. Database Syst.*, 28(4):427–466.

Garrigós, I., Pardillo, J., Mazón, J.-N., and Trujillo, J. (2009). A conceptual modeling approach for olap personalization. In *ER*, pages 401–414.

Golfarelli, M. and Rizzi, S. (2009). Expressing olap preferences. In *SSDBM*, pages 83–91.

Golfarelli, M., Rizzi, S., and Biondi, P. (2011). myolap: An approach to express and evaluate olap preferences. *IEEE Trans. Knowl. Data Eng.*, 23(7):1050–1064.

Gray, J., Bosworth, A., Layman, A., and Pirahesh, H. (1996). Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. In *ICDE*, pages 152–159.

Hurtado, C. A., Mendelzon, A. O., and Vaisman, A. A. (1999). Maintaining data cubes under dimension updates. In *ICDE*, pages 346–355.

Jerbi, H., Ravat, F., Teste, O., and Zurfluh, G. (2009). Modèle de préférences contextuelles pour les analyses olap. In *EGC*, pages 253–258.

Kießling, W. and Köstler, G. (2002). Preference sql - design, implementation, experiences. In *VLDB*, pages 990–1001.

Kozmina, N. and Niedrite, L. (2010). Olap personalization with user-describing profiles. In *BIR*, pages 188–202.

Lacroix, M. and Lavency, P. (1987). Preferences; putting more knowledge into queries. In *VLDB*, pages 217–225.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Pedersen, T. B. and Jensen, C. S. (2001). Multidimensional database technology. *IEEE Computer*, 34(12):40–46.

Ravat, F. and Teste, O. (2008). *Personalization and OLAP Databases*, volume New Trends in Data Warehousing and Data Analysis, chapter chapter 4, pages 1–22.

Rizzi, S. (2007). Olap preferences: a research agenda. In *DOLAP*, pages 99–100.

Thalhammer, T., Schrefl, M., and Mohania, M. K. (2001). Active data warehouses: complementing olap with analysis rules. *Data Knowl. Eng.*, 39(3):241–269.

Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *ICML*, pages 577–584.