Evolution de schéma par classification automatique pour les entrepôts de données

Ony RAKOTOARIVELO *, Fadila BENTAYEB **

* ERIC, Université Lumière Lyon2, 05 av. Pierre Mendès-France, 69676 BRON Cedex, France
ony.rakotoarivelo@eric.univ-lyon2.fr, bentayeb@eric.univ-lyon2.fr
http://eric.univ-lyon2.fr

Résumé. Les modèles et outils OLAP actuels gèrent les dimensions d'analyse d'un entrepôt de données de manière statique. Par conséquent, les axes d'analyse restent souvent figés malgré l'évolution des besoins et des données. Dans cet article, nous proposons une approche d'évolution de schéma basée sur une technique de classification automatique. Pour cela, nous cherchons le meilleur regroupement des instances d'un niveau d'analyse choisi par l'utilisateur en utilisant la méthode des k-means. Un nouvel axe d'analyse est ensuite construit à partir du résultat de cette classification. Pour choisir les descripteurs du niveau d'analyse à classifier, nous proposons deux solutions: la première utilise directement les attributs décrivant le niveau à classifier. Par contre, la deuxième solution décrit le niveau d'analyse par les mesures dans la table des faits. Pour valider notre approche, nous l'avons intégrée et testée à l'intérieur du SGBD *(Système de Gestion de Bases de Données)* Oracle 10g.

1 Introduction

Un entrepôt de données est une base de données multidimensionnelle dont l'utilisation principale est l'analyse en ligne. Ce type d'analyse permet à l'utilisateur de naviguer à l'intérieur des données et effectuer des comparaisons dans le temps. Ainsi, elle impose deux contraintes majeures sur l'entrepôt : les données entreposées doivent être non volatiles et historisées. Pour gérer ces contraintes, les modèles OLAP actuels préconisent de proscrire toute forme de modification sur l'entrepôt. En conséquence, il est difficile d'adapter le schéma de l'entrepôt par rapport à l'évolution des besoins d'analyse.

Quelques travaux de recherches se sont alors penchés sur ce problème. Hurtado et al. sont parmi les premiers à proposer une algèbre d'évolution de schéma pour les entrepôts de données. (Hurtado et al., 1999a,b). Pour cela, ils modélisent une dimension d'analyse par un graphe acyclique direct où les noeuds représentent les attributs de la dimension et les arêtes représentent les liens hiérarchiques entre ces attributs. Ils proposent par la suite des opérateurs qui permettent de modifier la structure du graphe tout en préservant les propriétés du graphe

de départ (c'est-à-dire que le graphe modifié doit rester acyclique et direct). Ces modifications peuvent prendre plusieurs formes : ajout ou suppression de sommets, ajout ou suppression d'arêtes. De même, Blaschka et al. enrichissent les travaux ci-dessus en proposant un ensemble d'opérateurs qui sont indépendants de tout modèle logique et physique de l'entrepôt (Blaschka et al., 1999). L'idée est de proposer des opérateurs très élémentaires qui vont agir sur le modèle conceptuel de l'entrepôt et propager les modifications sur le modèle logique et le modèle physique. Toute modification complexe sera ensuite effectuée en combinant ces opérateurs. L'inconvénient principal des travaux que nous venons de citer est le fait qu'ils ne travaillent que sur la dernière version des données. Ils ne tiennent pas compte de l'historique des données. Pour pallier ce manque, les modèles multidimensionnels temporels sont apparus. Ces modèles permettent de tracer l'historique des évolutions avec des étiquettes temporelles. Ces étiquettes temporelles labélisent, soit les données elles-même (Bliujute et al., 1998), soit les liens d'agrégation (Vaisman et Mendelzon, 2000), soit les différentes versions de l'entrepôt (Morzy et Wrembel, 2003, 2004). Le langage de requête de chaque modèle exploite ensuite ces étiquettes pour répondre correctement aux requêtes d'analyse de l'utilisateur.

Dans cet article, nous nous sommes intéressés à la manière d'identifier et de matérialiser de nouveaux axes d'analyse pertinents au sein d'une dimension. Bentayeb et al. proposent une approche d'évolution de schéma guidée par l'utilisateur (Bentayeb et al., 2007). Dans cette approche, l'utilisateur peut exprimer des règles d'analyse qui vont indiquer au modèle la manière d'agréger les instances d'un niveau d'analyse donné. De notre côté, nous proposons un opérateur d'évolution de schéma basé sur une technique de classification automatique : les k-means. Nous avons choisi la méthode des k-means parmis d'autres méthodes pour sa complexité algorithmique (linéaire) ainsi que pour le format des résultats obtenus (une partition de la population de départ). Notre opérateur utilise les k-means pour trouver un bon regroupement des instances d'un niveau d'analyse existant niv_{inf} choisi par l'utilisateur. Ce regroupement est effectué, soit à partir des descripteurs directs du niveau d'analyse niv_{inf}, soit sur les variables de mesures agrégées sur le niveau niv_{inf}. A l'issu de cette classification, l'opérateur construit un nouveau niveau d'analyse niv_{sup}. Pour ce faire, il crée les modalités du niveau niv_{sup} à partir du nombre de classes obtenues. Il établit ensuite le lien d'agrégation allant de niv_{inf} vers niv_{sup} en fonction de la classe d'affectation de chaque instance de niv_{inf}.

Nous avons validé notre approche en l'intégrant et en le testant à l'intérieur du SGBD Oracle 10g. En utilisant les descripteurs du niveau d'analyse de départ, nous avons obtenu un nouveau niveau d'analyse représentant le regroupement naturel du niveau d'analyse de départ. Par contre, l'utilisation des variables de mesures a mis en évidence la tendance des faits par rapport à un niveau d'analyse. Ainsi, notre approche enrichit considérablement l'analyse multidimensionnelle car elle offre de nouveaux angles de vues intéressants sur les faits. Par ailleurs, la possibilité de créer dynamiquement des niveaux de hiérarchie apporte une certaine flexibilité aux modèles multidimensionnels.

La suite de l'article est organisée comme suit : dans la section 2, nous exposons notre démarche d'évolution de schéma. Dans la section 3, nous formalisons cette démarche. La section 4 présente l'implémentation et l'expérimentation de l'approche. Dans la section 5, nous concluons notre travail et présentons quelques perspectives.

2 Evolution de schéma basée sur la méthode des k-means

Dans le cadre de l'évolution de schéma au sein des entrepôts de données, notre approche se positionne dans le courant des travaux qui proposent des opérateurs permettant de faire évoluer la structure hiérarchique d'une dimension. Toutefois, notre originalité réside dans l'utilisation de la fouille de données pour réaliser cette évolution. Pour présenter les détails de notre approche, nous allons d'abord présenter brièvement la méthode des k-means. Nous allons ensuite esquisser l'idée générale et l'argumenter avec des exemples. Nous présenterons la formalisation de l'approche dans la prochaine section.

2.1 La méthode des k-means

L'algorithme des k-means est un algorithme de classification automatique qui procède par réallocation dynamique (Forgy, 1965; MacQueen, 1967). On l'appelle aussi *la méthode des centres mobiles*. En effet, il s'agit d'un algorithme itératif qui partitionne une population X en k classes les plus homogènes possibles où chaque classe est modélisée par un individu de référence : *l'individu-centre de la classe*. Cet individu-centre n'est rien d'autre que le barycentre de sa classe (c'est-à-dire la moyenne arithmétique de tous les individus affectés à la classe).

Pour répartir la population X dans une partition à k classes, la démarche de l'algorithme des k-means peut être résumée comme suit :

1. *Prendre aléatoirement k individus-centres initiaux ;*
2. *Affecter chaque individu x_i au centre C_j qui lui est le plus proche (au sens de la distance euclidienne) ;*
3. *Recalculer les coordonnées des k centres ;*
4. *Réitérer (2) et (3) tant que les centres bougent ;*

La qualité de la classification peut être évaluée par la dispersion totale des individus à l'intérieur des classes obtenues. Cette dispersion est faible lorsque les individus d'une classe sont très proches de leur centre. Par conséquent, la meilleure partition de X en k classes est la partition qui minimise cette dispersion.

Il existe d'autres méthodes de classification automatique telles que la C.A.H *(Classification Ascendante Hiérarchique)* ou les cartes auto-organisatrices (Kohonen, 1995). Nous avons choisi la méthode des k-means car nous pensons que c'est la méthode la mieux adaptée aux exigences majeures de l'analyse en ligne pour sa complexité algorithmique qui est faible et linéaire ainsi que pour le format des classes fournies par la méthode (une partition de la population à classifier).

2.2 Présentation de notre approche

2.2.1 Exemple de base

Dans toute la suite de l'article, nous nous baserons sur l'entrepôt de données que nous fournissons ici en exemple. Considérons un entrepôt de données *"Ventes"* (figure 1). Cet entrepôt

comporte deux mesures : le **revenu des ventes** et la **quantité vendue**. Ces mesures peuvent être étudiées sur trois dimensions : ***"Temps", "Produit"*** et ***"Région"***. La hiérarchie de la dimension *"Région"* possède trois niveaux : *magasin, ville* et *pays de localisation*. De même, la dimension *"Produit"* est hiérarchisée sur trois axes d'analyse : *article de produit, catégorie de produit* et *famille de produit*.

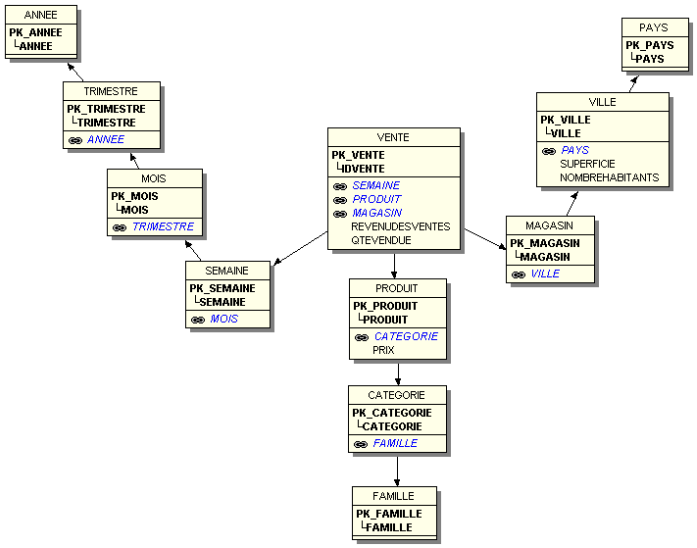

FIG. 1 – *Schéma de l'entrepôt de données "Ventes".*

2.2.2 Idée générale

La démarche classique d'analyse multidimensionnelle commence par la sélection des niveaux d'analyse et des mesures qui sont susceptibles de répondre au besoin d'analyse de l'utilisateur. Une fois que le cube de données associé à ce besoin est construit, l'utilisateur va explorer ce cube pour tenter de déceler rapidement des similarités entre les faits ou les dimensions qu'il étudie. Ce sont les niveaux d'analyse dans les dimensions qui permettent de détecter ces similarités. Pour aider l'analyste dans cette démarche, nous proposons alors un opérateur d'évolution de schéma permettant de créer un nouveau niveau d'analyse en se basant sur une méthode de classification automatique. Notre idée est d'ajouter un nouveau niveau d'analyse niv_{sup}, vers lequel un niveau d'analyse existant niv_{inf} peut être agrégé. Pour ce faire, l'opérateur va d'abord classifier les instances du niveau d'analyse niv_{inf} en utilisant la méthode des k-means. Il crée ensuite le nouveau niveau d'analyse niv_{sup} avec un nombre de modalités équivalent au nombre de classes obtenues. Il établit enfin le lien d'agrégation allant du niveau niv_{inf} vers le niveau niv_{sup} en fonction de la classe d'affectation de chaque instance de niv_{inf}. Ce nouveau niveau peut alors être intégré dans l'analyse ou servir à la construction de nouveaux cubes.

Pour choisir les descripteurs sur lesquels k-means va classifier les instances du niveau niv_{inf}, nous nous sommes penchés sur la manière d'explorer efficacement un cube de données. Classiquement, il en existe deux :

1. Soit l'utilisateur s'intéresse aux faits qui concernent un groupe d'individus qu'il connaît à l'avance. Il va donc focaliser son exploration sur les zones du cube où ces individus sont présents. Dans ce cas, il serait alors intéressant de classifier directement les instances du niveau d'analyse niv_{inf} avec ses propres descripteurs.

2. Soit l'utilisateur veut comprendre la tendance des faits que contient le cube. Dans ce cas, il va baser sa navigation sur la variation des mesures. Ceci lui permettra d'identifier les groupes d'individus qui expliquent cette tendance. Dans ce cas, nous proposons d'appliquer les k-means sur les variables de mesures dans la table des faits qui ont été préalablement agrégées sur le niveau d'analyse niv_{inf}.

Argumentons ces deux propositions au travers de deux exemples.

Proposition 1 : Classification basée sur les descripteurs dans la dimension

Considérons par exemple l'objectif d'analyse suivant : Faut-il fermer les points de ventes qui ne rapportent pas beaucoup ? Et faut-il ouvrir de nouveaux points de ventes dans les zones où les indicateurs sont très satisfaisants ?

Pour trouver une réponse à ces questions, l'analyste va essayer d'étudier les revenus des ventes à travers la dimension *"Région"* (figure 1). Pour améliorer la qualité de son analyse, il peut alors ressentir le besoin d'ajouter un nouveau niveau d'analyse qui doit regrouper les villes selon la densité de sa population. Dans ce cas, il serait intéressant de classifier directement chaque ville sur les descripteurs du niveau d'analyse *"Ville"* (*superficie* et *nombre d'habitants*) pour obtenir un regroupement en petite, moyenne et grande ville par exemple. A partir de cela, l'opérateur peut ensuite créer le niveau hiérarchique *"Groupe de Ville"* au dessus du niveau *"ville"* (figure 2) en se basant sur les résultats de la classification.

Proposition 2 : Classification basée sur les mesures dans la table des faits

Supposons que l'objectif d'analyse de l'utilisateur est de trouver la politique commerciale la mieux adaptée à chaque produit. Pour cela, l'utilisateur a besoin d'étudier le comportement d'achat des clients. Avec notre proposition, il peut créer un nouveau niveau d'analyse qui regroupe les produits en fonction du chiffre d'affaire qu'il rapporte. Pour ce faire, notre opérateur va d'abord agréger les mesures (*"revenus de vente"* et *"quantité vendue"*) sur le niveau d'analyse *"produit"*. Il exécutera ensuite la méthode des k-means sur le résultat de cette agrégation. A l'issu de cette classification, il va créer le niveau d'analyse "Groupe Produit" au dessus du niveau d'analyse "Produit" (figure 2).

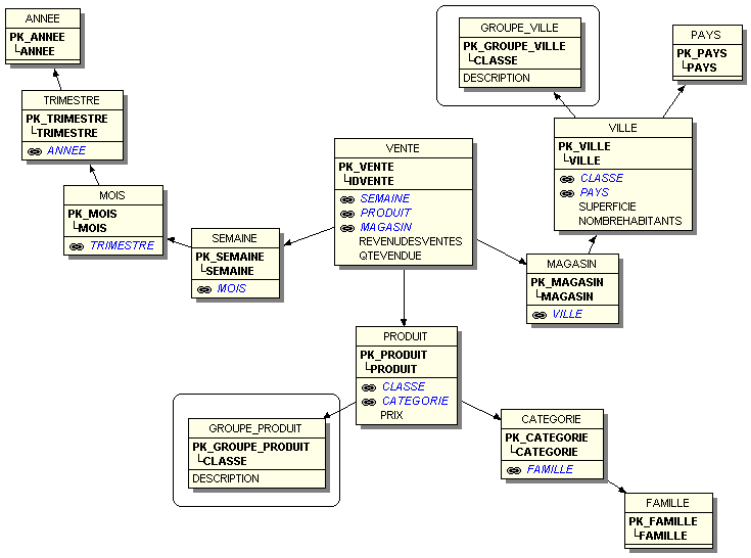

FIG. 2 – *Entrepôt "Ventes" après ajout des niveaux "Groupe Produit" et "Groupe Ville".*

3 Formalisation de l'approche

3.1 Rappels

Algébriquement, une base de données multidimensionnelle (c'est-à-dire, un entrepôt de données) est un couple $\mu = (\delta, \varphi)$ où δ est un ensemble de **dimensions** et φ est un ensemble de **faits**. Avant de formaliser notre problème d'évolution de schéma, rappelons brièvement les principales notions de l'algèbre multidimensionnelle (nous utilisons ici les notations qui ont été proposés par Hurtado et al. (Hurtado et al., 1999a)).

3.1.1 Dimensions

Le **schéma d'une dimension** est un couple $D = (L, \preceq)$ où L est un ensemble fini de niveaux hiérarchique et $\preceq$ est une relation binaire transitive et reflexive sur L traduisant le lien hiérarchique entre les éléments de L. Cette relation possède au moins deux niveaux spécifiques :

- l_{bottom} : représente l'unique niveau le plus bas de la relation $\preceq$,
- all : représente l'unique niveau le plus haut de la relation $\preceq$.

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

Chaque niveau $l \in L$ possède un ensemble d'instances qui prend ses valeurs dans un domaine $dom(l)$ (le domaine de définition du niveau $all, dom(all) = \{all\}$). Pour toute paire de niveaux $(l, l') \in L$ telle que $l \preceq l'$, il existe obligatoirement une fonction de correspondance f qui associe chaque instance du niveau l à une instance du niveau l' :

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

Exemple : Considérons la dimension $Region$ de la figure 1. Nous avons :

$$L_{region} = \{magasin, ville, pays, all \mid magasin \preceq ville \preceq pays \preceq all\}$$

Pour la paire de niveaux $(ville, pays)$, nous avons $dom(ville) = \{Paris, Lyon, Berlin\}$, $dom(pays) = \{France, Allemagne\}$ ainsi que la fonction f_{ville}^{pays} :

$$f_{ville}^{pays} = \{(Paris; France), (Lyon; France), (Berlin; Allemagne)\}$$

3.1.2 Faits

Le **schéma d'un fait** est un couple $F = (L_{group}, M)$ où $L_{group} = l_{D_1} \cup .. \cup l_{D_q}$ est la réunion de q niveaux d'analyse appartenant respectivement à q dimensions différentes et M est un niveau spécifique qu'on appelle **mesure**. Le domaine $dom(M)$ est un ensemble sur lequel, des opérations d'agrégation sont possibles (somme, moyenne, ...). Une instance x du fait F est donc une mesure $m \in dom(M)$ décri sur q niveaux :

$$\begin{array}{rcl} F : dom(l_{D_1})X...Xdom(l_{D_q}) & \longrightarrow & dom(M) \\ x(l_{D_1}, ..., l_{D_q}) & \longmapsto & m \end{array}$$

Exemple : Pour la table de fait $Vente$ de la figure 1, nous avons

$$Vente = ((Semaine \cup Produit \cup Magasin), QteVendue)$$

Le tableau 1 nous présente cinq instances du fait $Vente$.

Semaine	Produit	Magasin	QteVendue
1	p3	m1	10
1	p4	m2	2
2	p1	m2	5
2	p2	m3	7
3	p2	m2	4

TAB. 1 – *Instances de la table de fait "VENTE".*

3.1.3 Cubes de données

L'algèbre mutidimensionnelle fournit un opérateur *"Cube"* qui peut être défini comme suit : soient une table de fait $F = (L_{group} = \{l_1 \in D_1 \cup ... \cup l_p \in D_p\}, M)$ et un ensemble de niveaux d'analyse $GL = \{l'_1 \in D_1, ..., l'_p \in D_p \mid l_i \preceq l'_i \; \forall i = 1..p\}$. $CUBE(F, GL)$ fournit une nouvelle table de faits $F' = (GL, M')$ où M' est le résultat de l'agrégation de la mesure M du groupe de niveau L_{group} vers le groupe de niveau GL.

3.2 Cadre formel de l'approche

3.2.1 Ajout d'un nouveau niveau d'analyse

Pour ajouter un nouveau niveau d'analyse dans la hiérarchie d'une dimension, Hurtado et al. (1999a) ont définis l'opérateur $generalize$ dont la définition formelle peut être résumée comme suit : considérons une dimension $D_i = (L = \{l_{bottom}, ..., l, ..., all\}, \preceq)$, deux niveaux hiérarchiques $l \in L$ et $l_{new} \notin L$, et enfin une fonction $f_l^{l_{new}}$ qui associe chaque instance de l à une instance de l_{new}. $Generalize(D, l, l_{new}, f_l^{l_{new}})$ fournit une nouvelle dimension $D' = (L', \preceq')$ où $L' = L \cup \{l_{new}\}$ et $\preceq' = \preceq \cup \{(l \rightarrow l_{new}), (l_{new} \rightarrow All)\}$ conformément à la fonction de correspondance $f_l^{l_{new}}$.

Exemple : Considérons la dimension ***Région*** de la figure 1 ainsi que la fonction $f_{pays}^{continent} = \{(France, Europe), (Espagne, Europe), ..., (Canada, Amerique), ..., (Chine, Asie), ...\}$. $Generalize(Region, pays, continent, f_{pays}^{continent})$ ajoute un nouveau niveau ***"continent"*** dans la hiérarchie de la dimension ***Région*** et fournit la nouvelle structure hiérarchique suivante :

$$magasin \rightarrow ville \rightarrow pays \rightarrow continent$$

L'originalité de notre approche réside dans la construction de la fonction de correspondance $f_l^{l_{new}}$. Pour construire cette fonction, nous avons défini un opérateur que nous avons nommé "classifyWithKMeans".

3.2.2 L'Opérateur "classifyWithKMeans"

Soient k un nombre entier strictement positif, $X = \{x_1, x_2, ..., x_n\}$ une population de n individus et $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$ un ensemble de k classes.

L'opérateur $classifyWithKMeans(X, k)$ calcule (en utilisant l'algorithme des k-means) l'ensemble

$$C = \{c_1, ..., c_k \mid \forall i = 1..k, c_i = barycentre(\mathcal{C}_i)\}$$

et retourne la fonction de correspondance f_x^c telle que :

$$f_x^c = \{(x_i \rightarrow \mathcal{C}_j) \mid \forall i = 1..n \; et \; \forall m = 1..k, distance(x_i, c_j) \leq distance(x_i, c_m)\}$$

Exemple :

- $X = \{x_1 = 2, x_2 = 4, x_3 = 6, x_4 = 20, x_5 = 26\}$,
- $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2\}$,

classifyWithKeans (X, 2) retourne l'ensemble $C = \{c_1 = 4, c_2 = 23\}$ ainsi que l'application

$$f_x^c = \{(x_1 \rightarrow \mathcal{C}_1), (x_2 \rightarrow \mathcal{C}_1), (x_3 \rightarrow \mathcal{C}_1), (x_4 \rightarrow \mathcal{C}_2), (x_5 \rightarrow \mathcal{C}_2)\}$$

3.2.3 Algorithme

Paramètres en entrée :
- une dimension $D = (L, \preceq)$,
- un niveau d'analyse $l \in L$,
- un nouveau niveau d'analyse $l_{new} \notin L$,
- un nombre entier $k \geq 2$ qui va être le nombre de modalité de l_{new},
- une variable *dataSource* qui peut prendre deux valeurs : 'F' (pour *fait)* ou 'D' (pour *dimension).*

Etape 1 : Construction de la population d'apprentissage X_l

Cette première étape a pour objectif de constituer une population X_l à partir des instances du niveau d'analyse l. La population X_l sera décrite directement par les attributs de l si la valeur du paramètre *dataSource* est égale à 'D'. Dans le cas contraire (*dataSource* est égale à 'F'), X_l sera construite en exécutant l'opération $CUBE(F, l)$.

Exemple : Supposons que l'on désire créer un nouveau niveau *"groupe de villes"* au dessus du niveau d'analyse *"ville"*. Si le paramètre *dataSource* est égale à 'F', l'algorithme exécute l'opération $CUBE(Vente, ville)$. Nous obtenons ainsi la population décrite par le tableau 2. Dans le cas contraire, les villes seront décrites par leurs descripteurs dans l'entrepôt (tableau 3).

Ville	Revenues des ventes	Quantité vendue
Paris	10000	400
Chambéry	240	20
Lyon	120000	300
Saint-Etienne	1200	50

TAB. 2 – *Niveau d'analyse "ville" décrit par les mesures.*

Ville	Superficie	Nombre d'habitants
Paris	105	12 000
Chambéry	6	100
Lyon	60	6 000
Saint-Etienne	8	180

TAB. 3 – *Niveau d'analyse "ville" décrit par ses propres descripteurs.*

Etape 2 : Classification

Durant cette étape, l'algorithme utilise l'opérateur *classifyWithKMeans* sur la population d'apprentissage X_l qui a été créée durant l'étape précédente. Si, à titre d'exemple, le paramètre k est égale à 2, l'exécution de l'opérateur *classifyWithKMeans* sur le tableau 3 nous donne l'ensemble $\mathcal{C} = \{\mathcal{C}_1(82.5; 9000), \mathcal{C}_2(7, 140)\}$ ainsi que la fonction de correspondance

suivante :

$$f_{ville}^{groupedeville} = \{(Paris; \mathcal{C}_1), (Chambery; \mathcal{C}_2), (Lyon; \mathcal{C}_1), (SaintEtienne; \mathcal{C}_2)\}$$

Etape 3 : Création du nouveau niveau d'analyse

Cette étape consiste à matérialiser le nouveau niveau d'analyse l_{new} au coeur du schéma de l'entrepôt de données. Pour ce faire, notre algorithme utilise l'opérateur $Generalize$ sur la dimension D, à partir du niveau l et en utilisant la fonction de correspondance $f_l^{l_{new}}$ qui a été générée durant l'étape précédente de l'algorithme. En reprenant les exemples que l'on a pris dans les étapes 1 et 2, la création du niveau d'analyse *"groupe de villes"* consistera à exécuter l'opération $generalize(Region, ville, groupedeville, f_{ville}^{groupedeville})$.

4 Implémentation et Expérimentation

4.1 Environnement technique

L'algorithme que nous venons de présenter a été intégré à l'intérieur du SGBD Oracle 10g. Ainsi, nous avons programmé l'algorithme des k-prototypes avec le langage PL/SQL du SGBD Oracle 10g. La méthode des *k-prototypes* est une variante des k-means permettant de traiter simultanément des descripteurs numériques et catégoriels (Huang, 1997). Le choix d'intégrer l'approche à l'intérieur d'un SGBD est motivé par la volumétrie des entrepôts de données. En effet, les entrepôts sont souvent de très grandes bases données. En intégrant l'algorithme à l'intérieur d'un SGBD en utilisant les procédures stockées du SGBD, nous pouvons traiter de grosses volumes de données qui dépassent la taille de la mémoire de l'ordinateur de test.

4.2 Scenarii de test

Nos tests ont été effectués avec l'entrepôt de données *Emode* qui sert de base de démonstration de l'outil BusinessObject. Nous avons normalisé le schéma cet entrepôt pour qu'il soit identique au schéma de la figure 1. Sa table des faits "VENTE" contient 89200 enregistrements et la table de dimension *"Produit"* contient 663 enregistrements. Le niveau de granularité le plus fin de la dimension *"Produit"* contient 213 articles regroupés sur 34 catégories de produits et sur 12 lignes de produits. Notre principal objectif de test a été d'apprécier la pertinence des résultats de l'opérateur sur des données réelles. Ainsi, nous avons prévu les deux scénarii de tests suivants :

1. Créer un axe d'analyse *"groupe de prix"* qui classifie les articles selon leur prix.
2. Créer un axe d'analyse *"groupe d'article"* qui regroupe les articles selon les mesures dans la table de fait (revenu de ventes et la quantité vendue).

La figure 3 illustre les résultats de ces deux scenarii de test.

4.3 Discussion

Les résultats que nous avons obtenus met en évidence les points suivants :

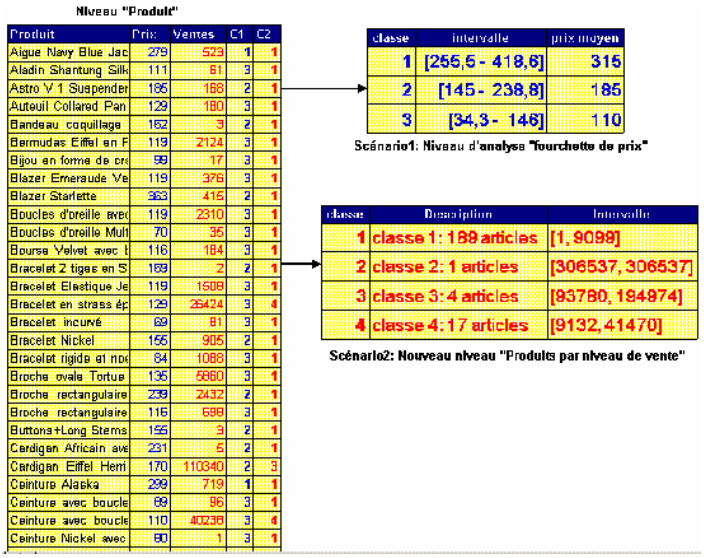

FIG. 3 – *Résultat des deux scenarii de test.*

- L'axe d'analyse créé avec le scénario numéro 1 permet d'étudier efficacement l'influence des prix sur les ventes. L'utilisation de ce nouvel axe dans l'analyse montre une corrélation assez forte entre le niveau des ventes et le prix des produits (figure 4).
- L'axe d'analyse qui a été créé avec le scénario numéro 2 a permis à l'utilisateur de voir les articles de produits qui se vendent bien et ceux qui se vendent moins bien (figure 4).
- Les classes obtenues avec les deux scenarii sont plus ou moins identiques. On a remarqué que la valeur des mesures pour individus qui ont été classifiés différemment par les deux scenarii sont assez atypiques. De cette remarque, on peut dire que notre approche permet d'identifier les individus à comportement atypique.

5 Conclusion et perspectives

Dans cet article, nous avons proposé une approche d'évolution de schéma qui permet d'ajouter dynamiquement un nouveau niveau d'analyse pertinent dans la hiérarchie d'une dimension donnée en utilisant la méthode des k-means. Cette méthode nous a permis de classifier les instances d'un niveau d'analyse niv_{inf} soit sur ses propres descripteurs, soit sur les mesures dans l'entrepôt de données. Nous avons ensuite transformé le résultat de cette classification en un nouveau niveau d'analyse niv_{sup}. Par ailleurs, nous avons présenté l'algorithme de notre approche et nous l'avons intégrée à l'intérieur du SGBD Oracle 10g. Nous avons ensuite effectuée une expérimentation qui a fourni des résultats encourageants.

Avec ce travail, nous avons montré l'intérêt de combiner la fouille de données et l'analyse multidimensionnelle pour la création de nouveaux axes d'analyse dans un entrepôt de

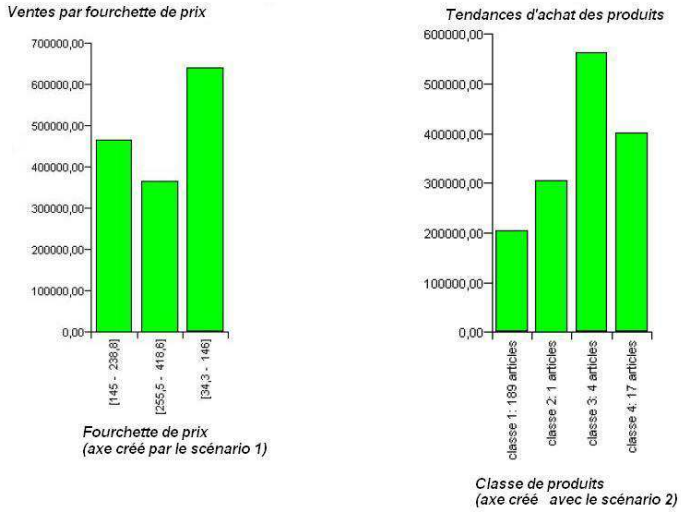

FIG. 4 – *Cubes de données générés avec les deux nouveaux niveaux d'analyse.*

données. Ainsi, quelques perspectives sont envisageables. En premier lieu, il serait intéressant de permettre à notre opérateur de créer des axes d'analyse de tendance qui tiennent compte de l'évolution des données dans le temps. Pour ce faire, nous proposons un "découpage horizontal" de la table des faits sur une unité de temps choisi par l'utilisateur. Un ensemble d'apprentissage sera ensuite extrait dans chaque sous-table de faits. On applique alors la classification sur chaque sous-population et l'on fusionne les résultats au sein d'un axe d'analyse unique. En deuxième lieu, il serait aussi intéressant d'étendre l'approche à l'apprentissage supervisé. On peut par exemple construire des modèles de prédiction à partir des résultats de la classification. Ces modèles peuvent être exploités pour identifier des règles d'analyse, pour donner une sémantique plus forte aux résultats de l'opérateur ou pour prédire la valeur des nouvelles données. Cette extension pourrait être suivie d'une généralisation de la fouille de données en ligne pour enrichir l'analyse multidimensionnelle avec des opérateurs d'extraction de connaissances intégrés dans les bases de données. Finalement, nous envisageons également de compléter notre approche par la création d'opérateurs de suppression et de modification de niveaux d'analyse.

Références

Bentayeb, F., C. Favre, et O. Boussaid (2007). A user-driven data warehouse evolution approach for concurrent personalized analysis needs. *Integrated Computer-Aided Engineering (ICAE), Special Issue (to appear)*.

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

Bliujute, R., S. Saltennis, G. Slivinskas, et C. Jensen (1998). Systematic change management in dimensional data warehousing. Technical report, Time Center - Computer Science Department - University of Arizona. Technical Report TR-23.

Forgy, E. (1965). Cluster analysis of multivariate data : efficiency versus interpretability of classification. In *Biometrics num 21*, pp. 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., A. Mendelzon, et A. Vaisman (1999a). Maintaining data cubes under dimension updates. In *Proc. 15th Int'l Conf. on Data Engineering, (ICDE'99)*, pp. 346–355.

Hurtado, C., A. Mendelzon, et A. Vaisman (1999b). Updating olap dimensions. In *DOLAP'99*, pp. 60–66.

Kohonen, T. (1995). *Self Organizing Maps*. Éditions Springer.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings 5th Berkeley Symposium*, pp. 281–297.

Morzy, T. et R. Wrembel (2003). Modeling a multiversion data warehouse : A formal approach. In *ICEIS (1)*, pp. 120–127.

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

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

Summary

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 article presents a new structural update operator for OLAP dimensions. This operator can 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. On the other hand, the second solution proposes to describe the level by measures attributes in the fact table. As data warehouses are very large databases, these solutions were integrated inside a RDBMS: the Oracle database system. In addition, we carried out some experimentations which validated the relevance of our approach.