$\mathcal{CND}$-$\mathcal{C}$ube : Nouvelle représentation concise sans perte d'information d'un cube de données

Hanen Brahmi*, Tarek Hamrouni*
Riadh Ben Messaoud**, Sadok Ben Yahia*

* Département des Sciences de l'Informatique, Faculté des Sciences de Tunis
{tarek.hamrouni,sadok.benyahia}@fst.rnu.tn,
** Faculté des Sciences Économiques et de Gestion
riadh.benmessaoud@fsegn.rnu.tn

Résumé. Le calcul des cubes de données est excessivement coûteux aussi bien en temps d'exécution qu'en mémoire et son stockage sur disque peut s'avérer prohibitif. Plusieurs efforts ont été consacrés à ce problème à travers les cubes fermés, où les cellules préservant la sémantique d'agrégation sont réduites à une cellule, sans perte d'information. Dans cet article, nous introduisons le concept du *cube de données non-dérivable fermé*, nommé $\mathcal{CND}$-$\mathcal{C}$ube, qui généralise la notion des modèles non-dérivables fermés fréquents bidimensionnels à un contexte multidimensionnel. Nous proposons un nouvel algorithme pour extraire le $\mathcal{CND}$-$\mathcal{C}$ube à partir des bases de données multidimensionnelles en se basant sur trois contraintes anti-monotones, à savoir "*être fréquent*", "*être non dérivable*" et "*être un générateur minimal*". Les expériences montrent que notre proposition fournit la représentation la plus concise d'un cube de données et elle est ainsi la plus efficace pour réduire l'espace de stockage.

1 Introduction

Depuis les années 90, l'émergence des besoins en aide à la décision a conduit aux développements *des entrepôts de données*. Ces derniers ont apporté une solution adéquate et efficace au problème de stockage et de gestion des données. Un entrepôt est une base centralisée de grands volumes de données, historisées, organisées par sujet et consolidées à partir de diverses sources d'informations. Son contenu est analysé par les applications *Online Analytical Processing* (OLAP) qui fournissent aux utilisateurs des moyens pour naviguer dans les données multidimensionnelles afin d'y découvrir des connaissances interprétables, exploitables et utiles à la prise de décision.

Dans le but de répondre efficacement aux requêtes OLAP, le calcul des cubes de données est une solution fréquemment adoptée. Cependant, il est bien connu que le calcul des cubes de données est un problème combinatoire (Wang et al., 2002). Le volume des agrégats générés peut être incomparablement plus important que celui des données initiales, elles-mêmes déjà très volumineuses.

Par exemple, étant donné un contexte d'extraction R contenant n attributs, le nombre de tuples dans un cuboïde (Group-By) à k-attributs, tel que ($0 \leq k \leq n$), est le nombre de tuples

dans R qui ont des valeurs d'attributs distinctes pour les k attributs. La taille d'un cuboïde est presque égale à la taille de R. Puisque le cube complet, construit à partir de R, consiste en 2^n cuboïdes, alors la taille de l'union des 2^n cuboïdes est beaucoup plus élevée que la taille de R.

En effet, la taille d'un cube augmente exponentiellement en fonction du nombre des dimensions. En outre, le problème s'aggrave, puisque nous traitons des jeux de données volumineux. Dans ce cadre, Ross et Srivastava donnent un exemple de ce problème en calculant un cube de données complet englobant plus que 210 millions de tuples à partir d'une relation initiale ayant 1 million de tuples (Ross et Srivastava, 1997). Généralement, le problème est dû aux deux raisons suivantes : le nombre exponentiel de combinaisons des dimensions et le nombre d'attributs par dimension. De plus, les cubes de données sont généralement épars (Ross et Srivastava, 1997). Ainsi en calculant un cube de données complet, les combinaisons de valeurs non fréquentes vont probablement être nombreuses et chaque exception doit être préservée. Dans un tel contexte, nous pouvons distinguer : (i) les approches favorisant l'efficacité des requêtes OLAP malgré l'espace de stockage, et (ii) celles favorisant des représentations optimales des cubes de données au lieu d'améliorer la performance des requêtes.

Bien que les contraintes liées à la taille des cubes de données aient attiré l'attention des chercheurs et divers algorithmes ont été développés visant un calcul rapide des cubes de données volumineux, moins de travaux se sont concentrés à la résolution du problème de la complexité de calcul des cubes de données à partir de la racine : la réduction de la taille d'un cube de données.

Dans cet article, nous examinons une autre façon afin d'attaquer ce problème. D'abord, nous introduisons le concept du *cube non-dérivable fermé* et nous montrons que le dernier réduit considérablement la taille d'un cube de données. Ensuite, nous introduisons un algorithme pour calculer efficacement les cubes non-dérivables fermés. Enfin, nous montrons l'efficacité de notre approche à travers une étude expérimentale critique menée sur des bancs d'essais et des bases réelles. Ces expérimentations portent à la fois sur l'efficacité de l'algorithme de calcul de la représentation et sur l'espace de stockage qui lui est nécessaire comparée aux approches s'inscrivant dans la même tendance.

Le reste de l'article est organisé comme suit. Dans la section 2, nous passons en revue les travaux antérieurs. Notre proposition est détaillée dans la section 3. Nous définissons les concepts de notre représentation et nous introduisons l'algorithme CLOSENDMG dans la section 4. Les résultats des expérimentations montrant l'utilité de l'approche proposée sont présentés dans la section 5. La conclusion et les travaux futurs font l'objet de la section 6.

2 Revue critique des travaux de l'état de l'art

De nombreux travaux de recherche ont abordé la problématique d'une représentation concise (compacte) des cubes de données permettant d'en réduire notablement la taille. Nous pouvons distinguer deux grandes tendances dans ces travaux suivant que les représentations définies entraînent ou pas une perte d'information.

Les approches qui choisissent de ne pas restituer les données exactes ou complètes se basent sur le fait que l'utilisateur d'un entrepôt s'intéresse aux grandes tendances générales dans la "population" examinée (Casali et al., 2009b). À cet égard, les algorithmes BUC (Beyer et Ramakrishnan, 1999) et HCUBING (Han et al., 2001) calculent des résultats exacts mais incomplets : le cube de données partiel ou iceberg cube. L'idée sous-jacente de cette tendance

d'approches est d'intégrer, dès le calcul du cube, les contraintes anti-monotones des utilisateurs potentiels, de manière à ne calculer et ne stocker que les agrégats correspondant à des tendances suffisamment générales pour être pertinentes.

A l'opposé, les travaux s'intéressant à une représentation sans aucune perte d'information se subdivisent en deux sous catégories : (i) ceux qui recherchent le meilleur compromis entre l'efficacité des requêtes OLAP et l'optimisation de l'espace de stockage. Ces approches ont privilégié l'efficacité des requêtes récurrentes tout en permettant un calcul, qui se veut optimal ; (ii) par ailleurs, il existe trois approches ayant choisi de caractériser une représentation concise et exacte des cubes de données, en se basant sur les algorithmes d'extraction des motifs fermés fréquents. Ces approches sont les suivants :

- Le **cube quotient** : Un cube quotient est un résumé du cube de données (Lakshmanan et al., 2002). Cette représentation concise peut être efficacement construite et réalise une réduction significative de la taille du cube de données. L'idée principale derrière un cube quotient est de créer un résumé en partitionnant soigneusement l'ensemble des cellules d'un cube de données selon des classes d'équivalences. Le partitionnement des cellules se fait tout en gardant la sémantique ROLLUP et DRILLDOWN et la structure de treillis. Par ailleurs, Casali *et al.* ont prouvé que le cube quotient peut être calculé par l'application des algorithmes d'extraction des motifs fermés fréquents, tel que l'algorithme CLOSE (Pasquier et al., 1999).

- Le **cube fermé** : Un cube fermé offre une représentation de taille réduite d'un cube de données comparée au cube quotient (Casali et al., 2009a). Il est composé, uniquement, de cellules fermées. Une cellule *c*, est dite une *cellule fermée* s'il n'y a aucune cellule *d*, telle que *d* est une spécialisation (descendante) de *c*, qui a la même mesure que *c*. Casali *et al.* ont prouvé que le cube fermé peut être calculé en utilisant les algorithmes d'extraction des motifs fermés fréquents.

- La **représentation RSM** [1] : C'est une approche à trois phases, qui exploite les algorithmes d'extraction de motifs fermés fréquents pour construire un cube fermé fréquent (Ji et al., 2006). L'idée de base est de : (i) transformer un ensemble de données à trois dimensions (3D) à un ensemble de données à deux dimensions (2D) ; (ii) extraire les motifs fermés fréquents à partir des données 2D en utilisant des algorithmes existants de fouille de données ; (iii) et élaguer n'importe quel cube fréquent qui n'est pas fermé.

Représentations concises et exactes basées sur les algorithmes de fouille de données		**Représentations basées sur un compromis entre l'espace de stockage et l'efficacité des requêtes OLAP**	
Approche	**Algorithme**	**Approche**	**Algorithme**
Cube Quotient (Lakshmanan et al., 2002)	CLOSE	Cube CURE (Morfonios et Ioannidis, 2006)	CURE
Cube Fermé (Casali et al., 2009a)	CLOSE	Cube Condensé (Wang et al., 2002)	BST
RSM (Ji et al., 2006)	CLOSE	Cube Dwarf (Sismanis et al., 2002)	STA

TAB. 1 – *Approches de réduction des cubes de données sans perte d'information.*

[1] RSM est l'acronyme de **Representative Slice Mining**.

Le tableau 1 récapitule les approches proposées, consacrées à la réduction des cubes de données sans perte d'information. En particulier, l'objectif principal des approches, qui utilisent les algorithmes de la fouille de données, est la réduction de l'espace de stockage sur le disque. Vue son importance, la réduction de l'espace de stockage d'un cube de données sur le disque ne cesse de présenter une problématique faisant l'objet de plusieurs recherches. À cet égard, l'intérêt principal de cet article est de proposer une nouvelle représentation concise sans perte d'information, appelée *cube non-dérivable fermé* et notée $\mathcal{CND}$-$\mathcal{C}$ube, que l'on peut considérer comme une extension des modèles fermés non-dérivables à l'espace de recherche multidimensionnel. L'idée principale derrière notre approche vient de la conclusion tirée par la communauté de la fouille de données qui s'est concentrée sur la réduction sans perte d'information des motifs fréquents. En effet, il a été prouvé que ***les motifs non-dérivables fermés fréquents*** offrent des taux de compression très élevés sans perte d'information (Muhonen et Toivonen, 2006). À cet effet, nous essayons d'extraire un $\mathcal{CND}$-$\mathcal{C}$ube qui permet d'obtenir la représentation des données multidimensionnelles la plus petite afin de réduire notablement l'espace de stockage sur le disque. Pour construire le $\mathcal{CND}$-$\mathcal{C}$ube, nous introduisons un nouvel algorithme appelé CLOSENDMG [2].

3 $\mathcal{CND}$-$\mathcal{C}$ube : Nouvelle approche du cube de données

Le cube non-dérivable fermé, appelé $\mathcal{CND}$-$\mathcal{C}$ube, englobe toute l'information jointe dans un cube de données. En outre, nous appliquons un mécanisme simple qui réduit, d'une manière significative, la taille des agrégats à stocker. Notre but est de calculer la plus petite représentation concise et exacte d'un cube de données, comparée aux approches existantes. Casali et al. (2009a,b) ont prouvé qu'il y a un isomorphisme de treillis entre les cubes fermés et le treillis de Galois (treillis de concept) calculés à partir d'un contexte d'extraction *R*. Cet isomorphisme est important puisqu'il permet l'utilisation des algorithmes de la fouille de données. Il est aussi prouvé être efficace pour le calcul des représentations concises d'un cube de données. Par exemple, le calcul du *cube fermé*, du *cube quotient* et de la représentation RSM est basé sur les algorithmes de la fouille de données. D'autre part, l'approche de Casali *et al.* est basée sur le théorème de Birkhoff (Ganter et Wille, 1999) pour passer d'un treillis de concepts à un treillis cube fermé.

Dans notre approche, nous utilisons aussi cet isomorphisme et le théorème de Birkhoff afin d'appliquer notre algorithme, CLOSENDMG, pour calculer le $\mathcal{CND}$-$\mathcal{C}$ube. Plus précisément, à partir d'un contexte d'extraction *R*, nous extrayons les motifs non-dérivables fermés en calculant les fermetures des générateurs minimaux non-dérivables. Ensuite, en se basant sur l'isomorphisme, nous employons le théorème de Birkhoff pour obtenir le $\mathcal{CND}$-$\mathcal{C}$ube. Ainsi, nous proposons d'utiliser notre algorithme CLOSENDMG. Ce dernier fonctionne en deux étapes : la première étape extrait les motifs respectant trois contraintes anti-monotones, à savoir "*être fréquent*", "*être non-dérivable*" et "*être un générateur minimal*". Tandis que, la deuxième étape calcule les fermetures des générateurs minimaux non-dérivables.

[2] CloseNDMG est l'acronyme de Closed Non-Derivable Minimal Generators.

4 Comment calculer un $\mathcal{CND}$-$\mathcal{C}$ube ?

4.1 Fondements mathématiques

4.1.1 Motif fermé

L'ensemble des motifs fermés, basé sur le concept de fermeture, constitue une représentation concise de l'ensemble des motifs fréquents (Pasquier et al., 1999).

Définition 1 *Soit γ l'opérateur de fermeture affectant à un motif X son sur-ensemble maximal ayant la même valeur du support que X. Un motif X est fermé si seulement si $X = \gamma(X)$.*

4.1.2 Générateur minimal

Le concept de générateur minimal (Bastide et al., 2000) est défini comme suit.

Définition 2 *Un motif g est dit générateur minimal d'un motif fermé f, ssi $\gamma(g) = f$ et $\nexists\, g_1 \subset g$ tel que $\gamma(g_1) = f$. Pour un seuil minimal de support, Minsup fixé a priori par l'utilisateur, l'ensemble des générateurs minimaux fréquents inclut tous les générateurs qui sont fréquents.*

D'après la définition d'un motif fermé et celle d'un générateur minimal, il en découle qu'un motif fermé est l'élément maximal d'une classe d'équivalence induite par l'opérateur de fermeture γ, alors qu'un générateur minimal est un élément minimal de la classe.

4.1.3 Motif non-dérivable

La collection des motifs non-dérivables fréquents, notée $\mathcal{MND}$, est une représentation concise et exacte des motifs fréquents basée sur le principe d'inclusion-exclusion (Calders et Goethals, 2007).

Définition 3 *Soit X un motif et Y un sous-ensemble de X. Si $|X \backslash Y|$ est impair, alors la règle de déduction correspondante à une borne supérieure pour le Supp(X) est :*

$$Supp(X) \leq \sum_{Y \subseteq I \subset X} (\text{-}1)^{|X \backslash I| + 1}\, Supp(I)$$

Si $|X \backslash Y|$ est pair, le sens de l'inégalité est inversé et la règle de déduction donne une borne inférieure au lieu d'une borne supérieure du support de X. Étant donné tous les sous-ensembles de X et leurs supports, nous obtenons un ensemble de bornes supérieures et inférieures pour X. Dans le cas où la plus petite borne supérieure est égale à la borne inférieure la plus élevée, le support de X est exactement dérivé. Un tel motif s'appelle *dérivable*. Dans le reste, les bornes inférieures et supérieures du support d'un motif X seront respectivement notées par $X.l$ et $X.u$.

4.1.4 Motif fermé non-dérivable

L'ensemble des motifs fermés fréquents non-dérivables, notés par $\mathcal{MFND}$, a été introduit par Muhonen et Toivonen (2006). Cette représentation concise exacte allie à la fois la notion du motif fermé fréquent et celle du motif non-dérivable fréquent. Elle consiste à appliquer à chaque motif non-dérivable fréquent l'opérateur de fermeture γ.

Définition 4 *Soit $\mathcal{MND}$ la collection des motifs non-dérivables fréquents. L'ensemble des motifs fréquents fermés non-dérivables est comme suit : $\mathcal{MFND} = \{\gamma(X) \mid X \in \mathcal{MND}\}$.*

4.2 Liaison entre les motifs fermés non-dérivables et les générateurs minimaux

Le calcul des motifs non-dérivables fermés fréquents peut être optimisé si nous utilisons les générateurs minimaux. En effet, il peut être prouvé que chaque motif non-dérivable fermé résulte du calcul de la fermeture d'un générateur minimal non-dérivable. Nous notons par un "générateur minimal non-dérivable", un motif qui est en même temps générateur minimal et non-dérivable. Par conséquent, au lieu de calculer l'ensemble des motifs non-dérivables fréquents puis leurs fermetures associées tel que le cas dans (Muhonen et Toivonen, 2006), nous pouvons seulement utiliser l'ensemble des générateurs minimaux fréquents non-dérivables. Pour obtenir l'ensemble des générateurs minimaux non-dérivables fréquents, une modification des algorithmes consacrés à l'extraction des motifs non-dérivables fréquents doit être effectuée. Son but principal est de maintenir seulement les motifs respectant la contrainte des générateurs minimaux parmi l'ensemble des motifs non-dérivables fréquents. Par conséquent, l'introduction des générateurs minimaux dans les algorithmes NDI et FIRM [3] optimisera l'étape de génération des candidats et les étapes de calcul de la fermeture. En effet, le nombre des générateurs minimaux fréquents non-dérivables est inférieur à celui des motifs non-dérivables.

4.3 L'algorithme CLOSENDMG

4.3.1 Générateurs minimaux fréquents non-dérivables

L'idée principale est de retenir seulement les motifs respectant la contrainte des générateurs minimaux parmi l'ensemble des motifs non-dérivables fréquents.

$\mathcal{GMC}_k$	:	Ensemble des générateurs minimaux non-dérivables candidats de taille k.
$\mathcal{GMF}_k$	:	Ensemble des générateurs minimaux non-dérivables fréquents de taille k.
$\mathcal{G}en$	:	Ensemble des générateurs minimaux non-dérivables de taille k à partir desquels seront générés les motifs non-dérivables candidats de taille $k+1$.
$\mathcal{P}reC_{k+1}$	:	Ensemble des générateurs minimaux non-dérivables fréquents de taille k+1.
$\mathcal{GMFND}$	:	Ensemble des générateurs minimaux non-dérivables fréquents.
$\mathcal{GMNDFF}$	:	Ensemble des générateurs minimaux non-dérivables fermés fréquents généré en utilisant l'algorithme CLOSENDMG avec leurs supports associés.
Supp-estimé	:	Ce champ contient le support estimé d'un candidat g de taille k. Ce support est égal au minimum du support des sous ensembles de taille (k-1) de g.

TAB. 2 – *Liste des notations utilisées dans l'algorithme* CLOSENDMG.

Définition 5 *Étant donné un motif X, l'ensemble des $\mathcal{GMFND}$ est défini comme suit : $\mathcal{GMFND} = \{X.l \neq X.u$ et X est un $\mathcal{GM}\}$.*

Nous pouvons conclure le théorème suivant, à propos de la cardinalité de l'ensemble des $\mathcal{GMFND}$:

[3] L'algorithme FIRM (Muhonen et Toivonen, 2006) extrait les motifs fermés non-dérivables.

Théorème 1 *La cardinalité de l'ensemble des générateurs minimaux non-dérivables, $\mathcal{GMFND}$, est toujours plus petite ou égale à la cardinalité de l'ensemble des motifs non-dérivables $\mathcal{MND}$, c.-à.-d.,* $|\mathcal{GMFND}| \leq |\mathcal{MND}|$.

Preuve. D'après la définition 5, l'ensemble des $\mathcal{GMFND}$ renferme les motifs qui respectent la contrainte des générateurs minimaux parmi l'ensemble des motifs non-dérivables fréquents. Par conséquent, nous avons $|\mathcal{GMFND}| \leq |\mathcal{MND}|$. $\diamondsuit$

4.3.2 Générateurs minimaux non-dérivables fermés fréquents

Données :
1. $\mathcal{D}$: Une base de données,
2. *Minsup* : Un seuil minimal du support.

Résultat : $\mathcal{GMNDFF}$: Ensemble des générateurs minimaux non-dérivables fermés fréquents.

début
  $k := 1$; $\mathcal{GMFND} := \emptyset$;
  $\mathcal{GMC}_1 := \{\{i\} \mid i \in \mathcal{I}\}$;
  pour chaque $i \in \mathcal{GMC}_1$ **faire**
    $i.l := 0$; $i.u := |\mathcal{D}|$;
    Supp-estimé$(i) := |\mathcal{D}|$;
  tant que $\mathcal{GMC}_k \neq \emptyset$ **faire**
    Déterminer les supports réels des motifs candidats à partir de $\mathcal{D}$;
    $\mathcal{GMF}_k := \{I \in \mathcal{GMC}_k \mid$ *Supp*$(I) \neq$ *Supp-estimé*(I) et *Supp*$(I) \geq$ *Minsup*$\}$;
    $\mathcal{GMFND} := \mathcal{GMFND} \cup \mathcal{GMF}_k$;
    $\mathcal{G}en := \emptyset$;
    pour chaque $I \in \mathcal{GMF}_k$ **faire**
      si ($Supp(I) \neq I.l$) *et* ($Supp(I) \neq I.u$) **alors**
        $\mathcal{G}en := \mathcal{G}en \cup \{I\}$;
    $\mathcal{P}reC_{k+1} :=$ *Apriori-Gen*$(\mathcal{G}en)$;
    $\mathcal{GMC}_{k+1} := \emptyset$;
    pour chaque $I \in \mathcal{P}reC_{k+1}$ **faire**
      Calculer la borne inférieure $I.l$ et la borne supérieure $I.u$ du support de I;
      si $I.l \neq I.u$ *et* $I.u \geq$ *Minsup* **alors**
        /*I est un motif non-dérivable et éventuellement fréquent puisque la borne supérieure de son support est supérieure ou égale à *Minsup*.*/
        Supp-estimé$(I) :=$ *min*$\{$*Supp*$(J) \mid J \subset I$ et $|J| = k\}$;
        $\mathcal{GMC}_{k+1} := \mathcal{GMC}_{k+1} \cup \{I\}$;
    $k := k + 1$;
  $\mathcal{GMNDFF} := \{\gamma(I) \mid I \in \mathcal{GMFND}\}$;
  retourner $\mathcal{GMNDFF}$;
fin

Algorithme 1 : CLOSENDMG($\mathcal{D}$, *Minsup*)

Nous présentons la définition suivante pour calculer l'ensemble des générateurs minimaux fermés non-dérivables.

Définition 6 *Soit $\mathcal{GMFND}$ l'ensemble des générateurs minimaux fréquents non-dérivables. L'ensemble des motifs fermés fréquents non-dérivables basés sur les générateurs minimaux est $\mathcal{GMNDFF}$= {$\gamma(X)$| $X \in \mathcal{GMFND}$}.*

Nous proposons dans cet article l'algorithme CLOSENDMG permettant l'extraction de l'ensemble $\mathcal{GMNDFF}$. Son pseudo-code est illustré par l'algorithme 1. Les notations utilisées sont récapitulées dans le tableau 2. Le théorème suivant fait le lien entre les ensembles $\mathcal{GMNDFF}$ et $\mathcal{MNDF}$.

Théorème 2 *L'ensemble $\mathcal{GMNDFF}$ comporte les mêmes motifs que l'ensemble $\mathcal{MFND}$.*

Preuve. L'ensemble $\mathcal{MFND}$ regroupe les fermetures des motifs non-dérivables fréquents. Soit I un motif non-dérivable fréquent. D'une part, d'après la définition d'un générateur minimal (*cf.* Définition 2), I admet nécessairement un générateur minimal g, inclus dans I, de même fermeture. Ainsi, $\gamma(I) = \gamma(g)$. D'autre part, étant donné que la contrainte "être un motif non-dérivable" est une contrainte anti-monotone, g est aussi un motif non-dérivable étant inclus dans un motif non-dérivable, à savoir I. Il en découle de ces deux résultats que g est un générateur minimal non-dérivable fréquent. Ainsi, tout motif non-dérivable fréquent admet un générateur minimal non-dérivable fréquent de même fermeture. D'ou, $\{\gamma(X) \mid X \in \mathcal{MND}\} = \{\gamma(X) \mid X \in \mathcal{GMFND}\}$. Il en résulte que l'ensemble $\mathcal{GMNDFF}$ est égal à l'ensemble $\mathcal{MFND}$. $\diamondsuit$

Il est à cet effet important de noter que l'utilisation seulement des motifs non-dérivables qui vérifient la contrainte d'être générateur minimal, au lieu de l'ensemble total des non-dérivables, admet pour principal avantage la réduction de la redondance dans le calcul des fermés associés.

Dans cet article, nous essayons d'extraire un $\mathcal{CND}$-$\mathcal{C}$ube qui décrit "*des relations non-dérivables fermés*" entre les dimensions. En effet, la première étape, dans la phase de calcul, est l'extraction des générateurs minimaux fréquents non-dérivables (*cf.* lignes 2-23). L'idée principale derrière leur extraction est d'assurer un calcul efficace qui réduit l'espace de mémoire utilisé. Comme résultat, nous obtenons un cube de données composé, seulement, de générateurs minimaux fréquents non-dérivables. Lors de la deuxième étape, une compression du dernier cube obtenu est réalisée en calculant la fermeture des générateurs minimaux fréquents non-dérivables (*cf.* ligne 24). Le $\mathcal{CND}$-$\mathcal{C}$ube obtenu contient toute l'information jointe dans un cube de données. D'où, notre représentation fournit un mécanisme simple réduisant de manière significative la taille des agrégats à stocker.

5 Évaluation expérimentale

Nous avons choisi de comparer notre approche aux deux autres représentations concises sans perte d'information, *c.-à.-d.*, cube quotient, cube fermé et au cube complet. Pour calculer le cube quotient et le cube fermé, nous utilisons l'algorithme CLOSE considéré efficace dans la recherche de motifs fermés fréquents et pour lequel nous disposons des sources. Toutes les expériences ont été réalisées sur un PC équipé d'un Pentium 4 avec une fréquence d'horloge de 3 GHz et une mémoire RAM de 2 Go, utilisant la distribution de Linux Fedora Core 6 comme système d'exploitation.

Durant l'expérimentation effectuée, nous avons utilisé deux bancs d'essai denses : CHESS et MUSHROOM, deux bancs d'essai éparses : RETAIL et T10I4D100K [4], et deux bases réelles

[4]Disponible à l'adresse suivante : *http ://fimi.cs.helsinki.fi/data/.*

Base de test	Attributs	Tuples
COVTYPE	54	581 012
SEP85L	7 871	1 015 367
MUSHROOM	119	8 124
CHESS	75	3 196
RETAIL	16 470	88 162
T10I4D100K	1 000	100 000

TAB. 3 – *Descriptif des jeux de données.*

utilisées dans le contexte des cubes de données : COVTYPE [5], SEP85L [6]. Les caractéristiques de ces bases sont résumées par le tableau 3. La dernière colonne indique le nombre de tuples de chaque base de données. Notre étude expérimentale comprend deux volets : premièrement, nous comparons le temps d'exécution consommé par CLOSENDMG *vs.* celui de FIRM [7] pour calculer le $\mathcal{CND}$-$\mathcal{C}$ube. Deuxièmement, nous nous concentrons sur l'évaluation de la compacité, en termes de l'espace de stockage sur disque, de notre approche *vs.* celles proposées dans la littérature et s'inscrivant dans la même tendance.

5.1 Comparaison des performances de CLOSENDMG *vs.* FIRM

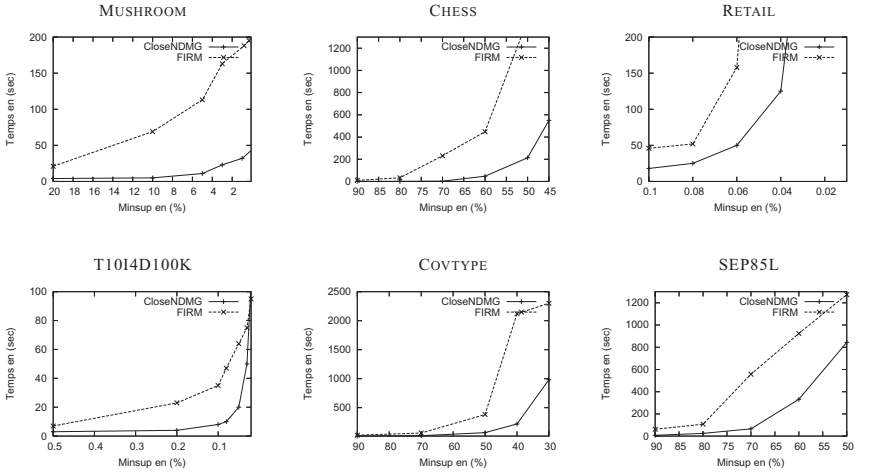

FIG. 1 – *Le temps d'extraction des $\mathcal{CND}$-Cubes en utilisant les algorithmes* FIRM *et* CLOSENDMG.

La figure 1 illustre le temps d'exécution consommé afin de générer le $\mathcal{CND}$-$\mathcal{C}$ube pour les bases de données considérées, en utilisant les algorithmes CLOSENDMG et FIRM. Nous constatons que l'algorithme CLOSENDMG est plus performant que FIRM sur les bases denses,

[5] Disponible à l'adresse suivante : *http ://ftp.ics.uci.edu/pub/machine-learning-databases/covtype.*
[6] Disponible à l'adresse suivante : *http ://cdiac.esd.ornl.gov/cdiac/ndps/ndp026b.html.*
[7] Disponible à l'adresse suivante : *http ://www.cs.helsinki.fi/u/jomuhone/.*

éparses et réelles pour toutes les valeurs de *Minsup*. Le mécanisme de comptage des générateurs minimaux adopté par CLOSENDMG s'avère plus efficace que FIRM. Ceci peut être expliqué par le nombre réduit des générateurs minimaux fréquents non-dérivables à prendre en considération lors du calcul de la fermeture. Cependant, l'algorithme FIRM est handicapé par un calcul redondant des fermetures.

5.2 Comparaison de l'espace de stockage des différentes propositions

Nous notons par "cube complet", un cube de données sans compression, *c.-à.-d.*, c'est un cube non réduit, généré à l'aide de l'opérateur "CUBE". Dans ce qui suit, nous comparons la taille du $\mathcal{CND}$-$\mathcal{C}$ube à stocker, respectivement *vs.* la taille d'un cube complet, d'un cube fermé et d'un cube quotient. Le tableau 4 présente l'espace sur le disque en (Ko) nécessaire pour stocker ces représentations de cube de données.

Base de test	Cube Complet	Cube Quotient	Cube Fermé	$\mathcal{CND}$-$\mathcal{C}$**ube**
MUSHROOM	10 147	4 021	2 972	**1 578**
CHESS	15 104	2 500	2 386	**1 009**
COVETYPE	20 825	6 900	5 410	**1 428**
SEP85L	32 912	7 383	5 925	**3 827**
T10I4D100K	15 398	12 890	10 987	**9 590**
RETAIL	13 025	11 986	11 913	**10 523**

TAB. 4 – *La taille des cubes de données générés* (*Ko*).

Les bases de données utilisées nécessitent trop de mémoire centrale (> 4 Go) pour calculer les représentations concises des cubes de données, *c.-à.-d.*, le $\mathcal{CND}$-$\mathcal{C}$ube, le cube quotient et le cube fermé, avec un seuil minimum fixé à 1 (situation où tous les motifs apparaissant dans une base donnée seraient extraits). Ainsi, nous avons dû induire un seuil minimum de fréquence pour chaque base de données, ce qui rend possible l'extraction des représentations concises des cubes de données. Selon le tableau 4, les expériences effectuées montrent que la représentation $\mathcal{CND}$-$\mathcal{C}$ube fournit une réduction importante de l'espace de stockage sur le disque comparée au cube complet, cube quotient et au cube fermé.

Le pourcentage de l'espace de stockage de notre représentation concise est plus petit que l'espace de stockage consommé par le cube complet. Pour les bases de données COVTYPE et SEP85L, notre représentation condensée exige, respectivement, 6,85% et 11,62% de l'espace requis pour stocker le cube de données complet. Comparés au cube quotient et au cube fermé, nos taux sont plus petits. Par exemple, le cube fermé exige, respectivement, 25,37% et 18,00%, de l'espace requis pour stocker un cube de données complet de COVTYPE et de SEP85L. Le cube quotient obtenu pour les deux dernières bases de données exige, respectivement, 33,13% et 22,43% de l'espace de stockage nécessaire pour stocker un cube complet. Nous concluons que pour les bases de données réelles, la compression est plus élevée en utilisant le $\mathcal{CND}$-$\mathcal{C}$ube *vs.* le cube fermé et le cube quotient. Les taux de compression obtenus pour les deux bases de données denses, MUSHROOM et CHESS, par le $\mathcal{CND}$-$\mathcal{C}$ube sont également significatifs. En outre, les taux de compression sont néanmoins beaucoup plus modestes pour les bases de données éparses, *c.-à.-d.* T10I4D100K et RETAIL exigent, respectivement, 62,28% et 90,57% de l'espace requis pour stocker le cube de données complet.

Considérons les trois représentations concises (cube fermé, cube quotient et $\mathcal{CND}$-$\mathcal{C}$ube), nous concluons que les meilleurs taux de compression d'un cube complet sont obtenus pour les

bases de données denses et réelles, *c.-à.-d.*, MUSHROOM, CHESS, COVTYPE et SEP85L. En outre, les taux de compression obtenus avec les bases de données éparses, *c.-à.-d.*, T10I4D100K, RETAIL sont plus petits et modestes mais ils sont toujours inférieurs à ceux obtenus par le cube fermé et le cube quotient.

6 Conclusion et perspectives

Dans cet article, nous nous sommes concentrés sur les approches de réduction des cubes de données, sans perte d'information utilisant les algorithmes de la fouille de données afin d'attaquer les défis suivants : un temps d'exécution coûteux aussi bien qu'un grand espace de stockage sur le disque. Ainsi, nous avons introduit un cube fermé appelé le $\mathcal{CND}$-$\mathcal{C}$ube basé sur un algorithme d'extraction efficace appelé CLOSENDMG. Les expériences, que nous avons réalisées, ont montré que le $\mathcal{CND}$-$\mathcal{C}$ube est plus performant que les représentations de réduction des cubes de données sans perte d'information existantes dans la littérature pour les différents types de contextes testés.

Les perspectives de travaux futurs concernent : (1) les hiérarchies présentent plusieurs complications dans la construction d'un cube de données. Particulièrement les hiérarchies constituent un défi principal. En effet, le nombre de tuples qui doit être matérialisé dans le cube devient très élevé ce qui augmente la consommation de l'espace de stockage sur le disque. Nous proposons de tenir en compte les hiérarchies des dimensions dans le cube non-dérivable fermé de la même manière que pour le cube CURE. (2) les règles génériques présentent un intérêt particulier puisqu'elles offrent le meilleur rapport compacité/informativité. En effet, les règles génériques permettent de déterminer l'ensemble minimal des règles d'association génériques présentées à l'utilisateur tout en maximisant la quantité d'informations utiles véhiculées. Nous proposons l'extraction des règles associatives "génériques multidimensionnelles" en se basant sur le $\mathcal{CND}$-$\mathcal{C}$ube ; (3) la génération des règles associatives de classifications pour prédire la mesure de nouveaux faits et ceci, à travers l'utilisation des ensembles flous comme moyen de discrétisation des mesures.

Références

Bastide, Y., N. Pasquier, R. Taouil, G. Stumme, et L. Lakhal (2000). Mining Minimal Non-Redundant Association Rules Using Frequent Closed Itemsets. In *Proceedings of the First International Conference on Computational Logic, Springer-Verlag, London, UK*, pp. 972–986.

Beyer, K. et R. Ramakrishnan (1999). Bottom-Up Computation of Sparse and Iceberg CUBEs. In *Proceedings of the 1999 ACM-SIGMOD International Conference on Management of Data (SIGMOD'99), Philadelphia, Pennsylvania, USA*, pp. 359–370.

Calders, T. et B. Goethals (2007). Non-Derivable Itemset Mining. *Data Mining and Knowledge Discovery, 14*(1), 171–206.

Casali, A., R. Cicchetti, et L. Lakhal (2009a). Closed Cubes Lattices. *Annals of Information Systems 3*, 145–165. Special Issue on New Trends in Data Warehousing and Data Analysis.

Casali, A., S. Nedjar, R. Cicchetti, L. Lakhal, et N. Novelli (2009b). Lossless Reduction of Datacubes Using Partitions. *International Journal of Data Warehousing and Mining (IJDWM) 4*(1), 18–35.

Ganter, B. et R. Wille (1999). F*ormal* C*oncept* A*nalysis*. Springer-Verlag.

Han, J., J. Pei, G. Dong, et K. Wang (2001). Efficient Computation of Iceberg Cubes with Complex Measures. In *Proceedings of the International Conference on Management of Data, SIGMOD, Santa Barbara, California*, pp. 441–448.

Ji, L., K.-L. Tan, et A. K. H. Tung (2006). Mining Frequent Closed Cubes in 3D Datasets. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06), Seoul, Korea*, pp. 811–822.

Lakshmanan, L., J. Pei, et J. Han (2002). Quotient Cube : How to Summarize the Semantics of a Data Cube. In *Proceedings of the 28th International Conference on Very Large Databases (VLDB'02), Hong Kong, China*, pp. 778–789.

Morfonios, K. et Y. E. Ioannidis (2006). Cure for Cubes : Cubing Using a ROLAP Engine. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06), Seoul, Korea*, pp. 379–390.

Muhonen, J. et H. Toivonen (2006). Closed Non-Derivable Itemsets. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'06), Berlin, Germany*, pp. 601–608.

Pasquier, N., Y. Bastide, R. Taouil, et L. Lakhal (1999). Efficient Mining of Association Rules Using Closed Itemset Lattices. *Journal of Information Systems 24*(1), 25–46.

Ross, K. et D. Srivastava (1997). Fast Computation of Sparse Data Cubes. In *Proceedings of the 23rd International Conference on Very Large Databases (VLDB'97), Athens, Greece*, pp. 116–125.

Sismanis, Y., A. Deligiannakis, N. Roussopoulos, et Y. Kotidis (2002). DWARF : Shrinking the Petacube. In *Proceedings of the 2002 ACM-SIGMOD International Conference on Management of Data (SIGMOD'02), Madison, USA*, pp. 464–475.

Wang, W., H. Lu, J. Feng, et J. Yu (2002). Condensed Cube : An Effective Approach to Reducing Data Cube Size. In *Proceedings of the 18th International Conference on Data Engineering (ICDE'02), San Jose, USA*, pp. 213–222.

Summary

It is well recognized that data cubes often produce huge outputs. Several efforts were devoted to this problem through closed cubes, where cells preserving aggregation semantics are losslessly reduced to one cell. In this paper, we introduce the concept of *closed non-derivable data cube*, denoted $\mathcal{CND}$-$\mathcal{C}$ube, which generalizes bidimensional frequent closed non-derivable patterns to the multidimensional context. We propose a novel algorithm to mine $\mathcal{CND}$-$\mathcal{C}$ube from multidimensional databases considering three anti-monotone constraints, namely "*to be frequent*", "*to be non-derivable*" and "*to be minimal generator*". Experiments show that our proposal provides the smallest representation of a data cube and thus is the most efficient for saving storage space.