Extraction d'Outliers dans des Cube de Données : une Aide à la Navigation

Marc Plantevit, Anne Laurent, Maguelonne Teisseire

Extraction d'outliers dans des cubes de données : une aide à la navigation

Marc Plantevit, Anne Laurent, Maguelonne Teisseire

LIRMM, Université Montpellier 2, CNRS,
161 Rue Ada 34392 Montpellier, France
nom.prenom@lirmm.fr

Résumé. La recherche d'algorithmes d'extraction de connaissances à partir de cubes de données est un domaine actuellement très actif qui trouve de très nombreuses applications dans les entrepôts de données disponibles maintenant dans la plupart des entreprises et milieux scientifiques (biologie, santé, etc.). Nous nous intéressons ici à l'extraction de comportements atypiques[1] (dénommés outliers) dans de tels cubes de données quand l'utilisateur veut identifier des séquences anormales. Par exemple, un directeur marketing aimerait savoir quelle zone géographique ne suit pas le même comportement que les autres afin de pouvoir y remédier. Pour ce faire, nous définissons une mesure de similarité capable d'appréhender de telles données complexes et définissons les algorithmes associés que nous avons testés sur différentes bases. Notons que nous considérons des cubes de données très denses, ce qui complexifie le problème de l'extraction.

1 Introduction

Les techniques d'extraction de connaissances apportent une aide non négligeable dans le contexte OLAP où l'utilisateur doit désormais prendre les décisions les mieux adaptées en un minimum de temps. De façon plus précise, la fouille de données constitue une étape clef dans le processus de décision face à de gros volumes de données multidimensionnelles en fournissant des motifs ou règles permettant une autre appréhension des données sources. Nous pouvons citer en particulier les travaux de recherche de motifs dédiés au contexte multidimensionnel (Pinto et al. (2001); Plantevit et al. (2006); Messaoud et al. (2006)). Néanmoins et en particulier lorsque les données sont fortement corrélées, la véritable connaissance n'est pas toujours celle associée aux comportements fréquents. C'est ainsi que les événements rares deviennent plus intéressants et font l'objet du processus même d'extraction. Par exemple, un directeur marketing préférera connaître quels sont les individus qui ne suivent pas les directives plutôt que de savoir que la quasi totalité des représentants suivent ses recommandations. De nombreuses applications ont été développées pour la détection de fraudes, la surveillance des activités criminelles dans le commerce électronique, le suivi des athlétes basées sur cette

[1]Ces travaux s'inscrivent dans le cadre d'un partenariat de recherche avec EDF R&D qui développe des méthodes d'Olap Mining. Cette collaboration s'intéresse en particulier à la détection d'évolutions temporelles atypiques dans des cubes.

recherche d'éléments, de motifs atypiques. Mais, à notre connaissance, il n'existe aucune proposition permettant d'extraire des séquences atypiques dans un contexte multidimensionnel. Il devient primordial d'être capable de fournir au décideur ce type de connaissances appelées motifs atypiques, rares ou outliers.
Notre contribution se situe dans ce contexte et propose une méthode de recherche de séquences multidimensionnelles atypiques. De plus, nous soulignons que de telles séquences peuvent constituer une aide à la navigation dans le cube de données. Nous proposons ainsi d'identifier les séquences qui se distinguent des autres à un niveau d'agrégation spécifié et ensuite de détecter quelles sont les causes de ces séquences atypiques en accédant au niveau plus fin. Les séquences atypiques des différents niveaux identifient ainsi des chemins de navigation dans le cube de données. La suite de cet article est organisée de la façon suivante. Section 2, nous décrivons les différentes approches de recherche de motifs atypiques puis nous détaillons le modèle de données dans lequel s'inscrit notre proposition. Après une illustration des motivations sur une base de données exemple Section 3, nous détaillons notre proposition de recherche guidée de séquences anormales Section 4. Nous précisons ainsi les définitions permettant de comparer les séquences afin d'obtenir des séquences atypiques et nous décrivons les algorithmes associés à la démarche de parcours guidé selon de telles séquences. Les expérimentations réalisées sont décrites et analysées Section 5. Enfin, Section 6, nous concluons sur le caractère générique de notre proposition, adaptable à d'autres mesures et dressons les perspectives de ces travaux.

2 Panorama des travaux existants et Modèle de données

Dans cette section, nous décrivons les travaux proposés pour l'extraction d'outliers tout d'abord dans un contexte classique puis dans un contexte OLAP. Nous définissons ensuite le modèle de données adopté dans le cadre de notre proposition.

2.1 Extraction d'outliers

Les outliers sont très répandus dans le monde réel. Ils ont plusieurs causes : erreurs dans la saisie ou l'enregistrement des données, événements vrais mais très rares (comportements volontairement ou involontairement non standards). Plus généralement, ils sont *tellement différents des autres observations qu'ils en sont suspicieux et ont dû être générés par un autre mécanisme* Hawkins (1980). Détecter des outliers est très important dans certains domaines tels que les détections de fraudes bancaires et les détections d'intrusions dans des réseaux informatiques. Les premiers travaux sur la détection d'outliers proviennent du monde des statistiques où de nombreuses approches ont été développées comme les tests de discordances Hawkins (1980); Barnett et Lewis (1994). En pratique, une règle 3σ est généralement adoptée. La règle 3σ est la suivante : Soient μ la moyenne et σ l'écart type, si une observation ne se situe pas dans l'intervalle $[\mu - 3\sigma, \mu + 3\sigma]$ alors on dit que cette observation est un outlier. Toutes ces méthodes sont développées pour détecter un unique outlier, et ne sont plus efficaces quand plusieurs outliers sont présents dans le jeu de données. Certains suggèrent d'utiliser la médiane ou la *mad scale* au lieu de la moyenne et de l'écart type afin de détecter des outliers multiples.

Cependant, il reste que ces approches ont été développées pour extraire des outliers dans un ensemble univarié où les éléments sont supposés suivre une distribution standard (Normale,

Poisson) alors que l'essentiel des données issues du monde réel sont multivariées et qu'il est difficile de définir la distribution que les régit.

De nombreux travaux proposent différentes méthodes pour détecter des outliers dans des données multivariées sans connaissance a priori de la distribution. Knorr et Ng donnent leur propre définition d'outlier basée sur la distance (Knorr et Ng (1997, 1998)). Un point est appelé un $DB(p, D)$ outlier si au moins une fraction p des points de l'ensemble de données sont à une distance supérieure à D. Ils prouvent aussi que leur définition est compatible avec les définitions d'outlier basées sur des distributions connues a prioiri (Hawkins ...). Ils définissent plusieurs algorithmes pour extraire des outliers basés sur la distance. Ramaswamy et al. (Ramaswamy et al. (2000))montrent que les DB outliers sont trop sensibles aux paramètres p et D. Ils définissent les outliers basés sur les k plus proches voisins. Ils calculent, pour chaque élément, les k plus proches voisins et ainsi la $k^{ème}$ distance. Ils ordonnent les éléments par rapport à cette distance et extraient les n données les plus déviantes. Breuning et al. (Breunig et al. (2000)) proposent la notion d'outlier local. Ils considèrent qu'un élément est un outlier seulement quand on considère son voisinage "local". Ils assignent à chaque objet un degré qu'ils appellent facteur d'outlier local. Ainsi ils utilisent un score continu pour mesurer les outliers au lieu de donner une réponse binaire : oui ou non. Aggarwal et Yu (Aggarwal et Yu (2001)) assurent que les deux précédentes approches (distance et local) ne fonctionnent pas très bien dans des ensembles contenant de nombreuses dimensions puisque les données sont "creuses" et les outliers doivent être définis dans une projection dans un sous-espace (sub space projection). Ils proposent un algorithme évolutif pour détecter les outliers. Fan et al. (Fan et al. (2006)) introduisent la notion d'outlier basé sur la résolution. Ils définissent un algorithme d'extraction d'outliers qui permet d'identifier facilement les top n outliers en prenant en compte les caractéristiques locales et globales d'un ensemble de données.

Les méthodes précédentes ont été définies pour détecter des objets singuliers au sein de la base de données. Elles ne permettent pas de caractériser des séquences d'objets comme outliers. Sun et al. (Sun et al. (2006)) proposent d'extraire des outliers dans des bases de données séquentielles. Pour approximer les mesures de distance, ils s'appuient sur des arbres probabilistes post-fixés.

Knorr et Ng (1997, 1998) proposent une version OLAP qui permet d'extraire des cellules outliers. Sarawagi et al. (Sarawagi et al. (1998)) proposent une exploration guidée par la découverte. Leur but est de découvrir des exceptions dans les cellules du cube. Ils définissent une cellule comme une exception si la mesure (agrégat) de la cellule diffère significativement de la valeur attendue. La valeur attendue est calculée par une formule et ils suggèrent une forme additive ou multiplicative. L'écart type peut être également estimé grâce à leur proposition. Quand la différence entre la cellule et la valeur attendue est supérieure à $2,5$ fois l'écart type, la cellule est une exception. Leur méthode peut donc être vue comme une version OLAP de la règle 3σ.

Lin et Brown (Lin et Brown (2003)) se focalisent aussi sur les cellules d'un cube OLAP. Ils définissent une fonction pour déterminer si la mesure d'une cellule est extrême. Quand la cellule est un outlier, les points contenus dans cette cellule sont associés. Ils combinent ainsi détection d'outlier et les concepts relatifs à OLAP afin d'établir des corrélations entre des événements (crimes).

Même si de nombreuses approches d'extraction d'outliers ont été proposées dans différents contextes, il n'existe pas d'approche permettant de caractériser des séquences outliers dans un

contexte multidimensionnel (plusieurs dimensions et une mesure) où les données sont définies à différents niveaux d'agrégation.

2.2 Modèle de données

Nous considérons les bases de données comme étant organisées en cubes eux-mêmes composés de *cellules*. Chaque cellule correspond à un n-uplet de la base défini sur un ensemble de dimensions $D = \{D_1, \ldots, D_n\}$ elles-mêmes définies sur leurs domaines actifs[2] respectifs $Dom(D_1), \ldots, Dom(D_n)$, et une mesure M (correspondant le plus souvent à une valuation numérique) définie sur le domaine $Dom(M)$. Le domaine de la mesure inclut la valeur nulle.

Définition 1 (HyperCube) *Un hypercube (ou simplement cube) de données défini sur les dimensions D_1, ..., D_n est un n-uplet $\langle Dom(D_1), \ldots, Dom(D_n), Dom(M), C\rangle$ où C est une application $C : Dom(D_1) \times \ldots \times Dom(D_n) \rightarrow Dom(M)$. Par abus de langage, on notera C un tel cube.*

Définition 2 (Cellule) *On appelle cellule d'un cube à n dimensions D_1,...,D_n, un n-uplet de la forme : $\langle (d_1, \ldots, d_n), \mu\rangle$ où :*
- $\forall i \in [1 \ldots n],\ d_i \in Dom(D_i)$
- $\mu \in Dom(M)$

Définition 3 (Projection) *Soit $cell = \langle (d_1, \ldots, d_n), \mu\rangle$ une cellule. On note $cell.D_i = d_i$ la restriction de $cell$ sur la dimension D_i.*

Définition 4 (Sous-Cube) *Soit C un cube à n dimensions $D = \{D_1, \ldots, D_n\}$, un sous-cube C' de C défini sur les k dimensions $\{D_{j_1}, \ldots, D_{j_k}\} \subseteq D$ dont les domaines actifs sont inclus, est un n-uplet $\langle d_{j_1}, \ldots, d_{j_k}\rangle$. Le sous-cube C' correspond à l'ensemble des cellules telles que : $c = \langle (c_1, \ldots, c_n), \mu\rangle$ avec $\mu \in Dom(M)$ et $\forall j \in [j_1, \ldots, j_k]\ \ c_j = d_j$.*

Un sous-cube est défini à partir de dimensions sur lesquelles les valeurs sont fixées. Un cube peut donc être partitionné en un ensemble de sous-cubes le long d'un ensemble de dimensions, tel que l'opération *split* du modèle multidimensionnel le réalise (Chaudhuri et Dayal (1997) et Marcel (1998)).

3 Base de données exemple et motivations

Afin d'illustrer notre approche, nous considérons l'exemple ci-dessous tout au long de cet article. Le cube de données C stocke les résultats commerciaux des diverses zones géographiques. Plus précisément, nous considérons que C est défini selon quatre dimensions comme l'indique la figure 2 où :
- $Date$ représente la date (cinq dates différentes notées $1, 2, 3, 4$ et 5).
- GEO représente le lieu. Il existe une hiérarchie au sein de cette dimension qui va du niveau le plus agrégé (national) vers des niveaux de granularité plus fins (régions, villes, magasin).

[2] Le domaine actif d'une dimension est, comme dans le modèle relationnel, la partie finie du domaine de la dimension contenant les valeurs présentes dans la base.

– $Fidélité$ représente le niveau de fidélité du client démarché. Sur ce cube, trois niveaux de fidélité sont disponibles : gold, silver et platinium.
– *Offre* représente l'option à laquelle le client a souscrit. Deux options sont présentes dans le cube de données exemple : opt1 et opt2.

Le cube de données C possède également une dimension particulière qu'est la *mesure*. La mesure est une application de : $Dom(GEO) \times Dom(date) \times Dom(Fidélité) \times Dom(Offre) \rightarrow Dom(Mesure)$. La mesure représente pour un lieu, le nombre d'options souscrites à une date donnée par des clients dans le cadre de leur Fidélité.

Par exemple, le premier 5-uplet $\langle (Paris, 1, Gold, opt1)800 \rangle$ indique que 800 clients $Gold$ à Paris, à la date 1, ont souscrit à l'option $opt1$. La figure 1 représente le cube de données exemple pour la date 1.

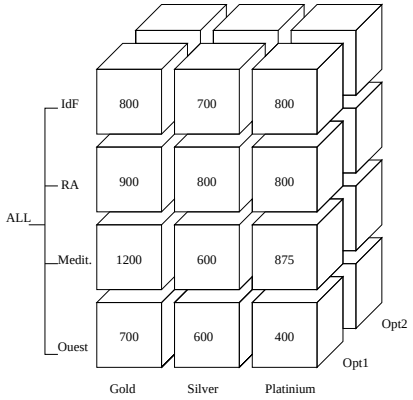

FIG. 1 – *Cube de données Exemple pour la date* 1

Motivations :
Nous souhaitons proposer à l'utilisateur un nouveau mode de navigation dans un cube de données. Cette navigation se base sur la recherche de séquences outliers. Nous proposons à l'utilisateur d'identifier à chaque niveau les n séquences qui diffèrent le plus des autres et ensuite de réitérer ce processus sur les séquences outliers à des niveaux inférieurs.

Par exemple, dans le cube de données exemple, *GEO Sud* ne suit pas le même comportement que les autres positions sur GEO (Paris, Ouest, RA). Nous allons donc nous repositionner à un niveau de granularité plus fin dans ce sous-cube afin d'essayer d'identifier les raisons pour lesquelles ce "lieu" est outlier, et ainsi extraire de nouveaux outliers dans ce sous-cube (Drill down de Sud). Si les outliers extraits dans ce sous-cube ne suivent pas le même comportement que les séquences "communes" du niveau supérieur, alors ces séquences peuvent être considérées comme responsables du fait que la séquence *Sud* soit outlier. On peut donc réitérer le processus à des niveaux d'agrégation plus fins sur les nouvelles séquences outliers. Si les séquences outliers suivent le même comportement que les séquences communes du niveau supérieur, alors elles ne peuvent pas êtres considérées comme cause de la rareté de la séquence supérieure. Dans ce cas, nous arrêtons la recherche d'outliers pour cet axe et donc la navigation à ce niveau.

On peut imaginer se situer dans un contexte où l'utilisateur est le directeur marketing national. Il veut vérifier dans un premier temps si tous les centres inter-régionnaux suivent ses

GEO	Date	Fidélité	Offre	Mesure
Paris	1	Gold	opt1	800
Paris	1	Gold	opt2	1000
Paris	1	Silver	opt1	700
Paris	1	Silver	opt2	700
Paris	1	Platinium	opt1	800
Paris	1	Platinium	opt2	900
Paris	2	Gold	opt1	1000
Paris	2	Gold	opt2	900
Paris	2	Silver	opt1	800
Paris	2	Silver	opt2	900
Paris	2	Platinium	opt1	900
Paris	2	Platinium	opt2	1300
Paris	3	Gold	opt1	1200
Paris	3	Gold	opt2	750
Paris	3	Silver	opt1	750
Paris	3	Silver	opt2	1000
Paris	3	Platinium	opt1	1300
Paris	3	Platinium	opt2	1000
Paris	4	Gold	opt1	1400
Paris	4	Gold	opt2	500
Paris	4	Silver	opt1	800
Paris	4	Silver	opt2	1200
Paris	4	Platinium	opt1	900
Paris	4	Platinium	opt2	1050
Paris	5	Gold	opt1	1500
Paris	5	Gold	opt2	500
Paris	5	Silver	opt1	690
Paris	5	Silver	opt2	1200
Paris	5	Platinium	opt1	850
Paris	5	Platinium	opt2	1100

(a) GEO Paris

GEO	Date	Fidélité	Offre	Mesure
RA	1	Gold	opt1	900
RA	1	Gold	opt2	1010
RA	1	Silver	opt1	800
RA	1	Silver	opt2	650
RA	1	Platinium	opt1	800
RA	1	Platinium	opt2	750
RA	2	Gold	opt1	1095
RA	2	Gold	opt2	910
RA	2	Silver	opt1	810
RA	2	Silver	opt2	870
RA	2	Platinium	opt1	900
RA	2	Platinium	opt2	1220
RA	3	Gold	opt1	1270
RA	3	Gold	opt2	730
RA	3	Silver	opt1	805
RA	3	Silver	opt2	1100
RA	3	Platinium	opt1	1300
RA	3	Platinium	opt2	1050
RA	4	Gold	opt1	1440
RA	4	Gold	opt2	580
RA	4	Silver	opt1	795
RA	4	Silver	opt2	1230
RA	4	Platinium	opt1	900
RA	4	Platinium	opt2	1070
RA	5	Gold	opt1	1490
RA	5	Gold	opt2	540
RA	5	Silver	opt1	720
RA	5	Silver	opt2	1220
RA	5	Platinium	opt1	850
RA	5	Platinium	opt2	1090

(b) GEO RA

GEO	Date	Fidélité	Offre	Mesure
Sud	1	Gold	opt1	1200
Sud	1	Gold	opt2	1300
Sud	1	Silver	opt1	600
Sud	1	Silver	opt2	750
Sud	1	Platinium	opt1	875
Sud	1	Platinium	opt2	850
Sud	2	Gold	opt1	1250
Sud	2	Gold	opt2	800
Sud	2	Silver	opt1	700
Sud	2	Silver	opt2	900
Sud	2	Platinium	opt1	910
Sud	2	Platinium	opt2	900
Sud	3	Gold	opt1	1160
Sud	3	Gold	opt2	950
Sud	3	Silver	opt1	550
Sud	3	Silver	opt2	1000
Sud	3	Platinium	opt1	975
Sud	3	Platinium	opt2	940
Sud	4	Gold	opt1	1080
Sud	4	Gold	opt2	1000
Sud	4	Silver	opt1	700
Sud	4	Silver	opt2	800
Sud	4	Platinium	opt1	950
Sud	4	Platinium	opt2	1400
Sud	5	Gold	opt1	1100
Sud	5	Gold	opt2	650
Sud	5	Silver	opt1	690
Sud	5	Silver	opt2	750
Sud	5	Platinium	opt1	880
Sud	5	Platinium	opt2	1000

(c) GEO Sud

GEO	Date	Fidélité	Offre	Mesure
Ouest	1	Gold	opt1	700
Ouest	1	Gold	opt2	870
Ouest	1	Silver	opt1	600
Ouest	1	Silver	opt2	500
Ouest	1	Platinium	opt1	400
Ouest	1	Platinium	opt2	800
Ouest	2	Gold	opt1	750
Ouest	2	Gold	opt2	800
Ouest	2	Silver	opt1	690
Ouest	2	Silver	opt2	745
Ouest	2	Platinium	opt1	600
Ouest	2	Platinium	opt2	1270
Ouest	3	Gold	opt1	900
Ouest	3	Gold	opt2	720
Ouest	3	Silver	opt1	740
Ouest	3	Silver	opt2	1050
Ouest	3	Platinium	opt1	1100
Ouest	3	Platinium	opt2	1050
Ouest	4	Gold	opt1	1200
Ouest	4	Gold	opt2	450
Ouest	4	Silver	opt1	810
Ouest	4	Silver	opt2	1150
Ouest	4	Platinium	opt1	700
Ouest	4	Platinium	opt2	1000
Ouest	5	Gold	opt1	1470
Ouest	5	Gold	opt2	460
Ouest	5	Silver	opt1	750
Ouest	5	Silver	opt2	1230
Ouest	5	Platinium	opt1	650
Ouest	5	Platinium	opt2	1060

(d) GEO Ouest

FIG. 2 – *Cube de données Exemple sous forme tabulaire*

ordres. Si un centre ne respecte pas ses directives, il veut voir si c'est au niveau du centre que les ordres ne sont pas passés ou si ce sont des sous-centres qui ne respectent pas ses directives et nuisent ainsi à la production du centre inter-régionnal.

4 Proposition d'une recherche guidée de séquences rares

Dans cette section, vous présentons notre contribution. Tout d'abord, il est nécessaire d'étudier les données que nous allons manipuler. Ensuite nous verrons comment nous mesurons d'une part, la distance entre deux séquences et d'autre part, entre une séquence et un ensemble de séquences. Enfin, nous présenterons les algorithmes permettant la recherche guidée d'outliers.

4.1 Données manipulées

Dans le cadre de ce travail et comme défini dans (Plantevit et al. (2005)), nous supposons que parmi toutes les dimensions définissant un cube de données, il existe au moins une ou un ensemble de dimensions D_t dont le domaine est ordonné(e.g. dimension temporelle).

Définition 5 (Partition des dimensions) *Pour tout cube défini sur les dimensions D, on considère une partition de D en quatre sous-ensembles notés respectivement :*

- *D_t pour la ou les dimensions temporelles*
- *D_A pour les dimensions dites d'*analyse
- *D_R pour les dimensions dites de* référence
- *D_F pour les dimensions oubliées.*

Il en découle que chaque cellule $cell = \langle (d_1, \ldots, d_n), \mu \rangle$ d'un cube peut être notée $cell = \langle (f, r, a, t), \mu \rangle$ où f, r, a et t correspondent respectivement aux restrictions de $cell$ sur D_F, D_R, D_A et D_t.

Dans le cadre de l'extraction de séquences outliers, l'ensemble D_R permet d'identifier les sous-cubes par rapport auxquels les séquences anormales seront extraites. Pour cette raison, cet ensemble est nommé *référence*. Chaque n-uplet défini sur D_R identifie une séquence. Nous désirons rechercher les séquences identifiées par des sous-cubes

Dans nos calculs, la séquence identifiée par un cube, sera dite anormale si elle est sensiblement différente des autres séquences identifiées par les autres sous-cubes.

L'ensemble D_t permet d'introduire une relation d'ordre entre les cellules. Cet ensemble permet donc d'introduire la notion de séquentialité.

L'ensemble D_A décrit les axes d'*analyse*, c'est-à-dire l'ensemble des dimensions apparaissant explicitement dans séquences extraites.

Il est aussi possible de définir un sous-ensemble D_F qui permet de décrire les axes *oubliés*, c'est-à-dire les dimensions qui ne servent ni à introduire une relation d'ordre, ni identifier un sous-cube, ni à définir la séquence elle-même. Ces dimensions dites *oubliées* peuvent être vues comme instanciées avec la valeur ALL.

Par rapport au cube de données exemple, nous choisissons :

- $D_t = \{date\}$
- $D_A = \{Fidélité, Offre\}$
- $D_R = \{GEO\}$
- $D_F = \{\}$

On note C_{D_R} l'ensemble des sous-cubes à partir des dimensions de référence. Ainsi, la figure 2 montre les quatre sous-cubes définis en fonction de $D_R = \{GEO\}$, c'est-à-dire C_{D_R}.

On note également $C_{(d_{r_1},d_{r_2},\ldots,d_{r_k})}$, le sous-cube identifié par le n-uplet $(d_{r_1}, d_{r_2}, \ldots, d_{r_k})$ défini sur les dimensions de références. Conformément au cube exemple, la figure 2(a) représente $C_{(Paris)}$.

4.2 Blocs et séquences

Définition 6 (Bloc) *On appelle bloc B un ensemble de cellules dont les positions sur D_R et D_T sont fixes. B est l'ensemble des n-uplets prenant leurs valeurs sur D_A :*

$$B = \{\langle({d_{i1}}^1, \ldots, {d_{im}}^1), \mu^1\rangle, \ldots, \langle({d_{i1}}^p, \ldots, {d_{im}}^p), \mu^p\rangle\}$$

On notera $B = \{c_1, \ldots, c_p\}$.

Définition 7 (Séquence) *On appelle séquence, une liste ordonnée non vide de blocs de la forme :*

$\varsigma = \langle\{\langle({d_{i1}}^1, \ldots, {d_{im}}^1), \mu^1\rangle, \ldots, \langle({d_{i1}}^p, \ldots, {d_{im}}^p), \mu^p\rangle\}, \ldots \{\langle({d_{i1}}^{1'}, \ldots, {d_{im}}^{1'}), \mu^{1'}\rangle, \ldots, \langle({d_{i1}}^{p'}, \ldots, {d_{im}}^{p'}), \mu^{p'}\rangle\}\rangle$

On note $\varsigma = \langle B_1, \ldots, B_l\rangle$.

Etant donnée la partition de l'ensemble de dimensions (D_A, D_R, D_t, d_F), un cube de données peut être vu comme un ensemble de séquences. Chaque séquence est identifiée par une valeur prise sur D_R. Chaque bloc d'une séquence correspond à une valeur sur D_t. Les blocs prennent ainsi leurs valeurs sur D_A.

La figure 3 illustre une séquence où $D_A = \{$A_1,A_2$\}$ pour une position sur D_R fixée. La séquence regroupe les blocs dont les positions des cellules varient sur D_A en fonction d'un ordre relatif à D_t. La séquence représentée dans la figure 3 contient quatre blocs différents. Les cellules colorées correspondent aux cellules non vides.

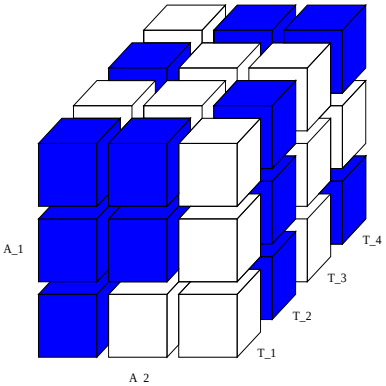

FIG. 3 – *Séquence de blocs pour une valeur de D_R*

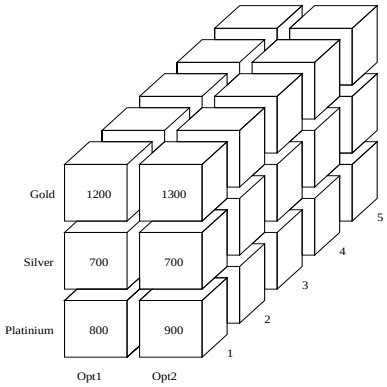

FIG. 4 – *Séquence pour GEO=Sud*

La figure 4 représente la séquence identifiée par GEO=Sud. Nous avons deux dimensions d'analyse. La séquence contient cinq blocs (les blocs aux dates $1, 2, 3, 4$ et 5).

4.3 Comparaison de séquences

Afin de définir si une séquence est outlier ou non, il est nécessaire de pouvoir calculer la similarité entre cette séquence et les autres. Nous introduisons ainsi une mesure de comparaison entre deux séquences. Cette mesure peut être une distance ou une mesure de similarité. Si la distance entre deux séquences est grande alors ces deux séquences seront considérées comme éloignées. De façon inverse, une mesure de similarité essaie de voir à quel point deux séquences sont proches. Si cette valeur est 1 alors ces séquences sont considérées comme identiques et si la mesure est égale à 0, alors les deux séquences n'ont rien en commun.

La distance la plus connue entre deux séquences est l'*edit distance*. L'edit distance correspond aux nombres d'opérations d'édition (insertion, suppression, déplacement) nécessaires pour transformer une séquence en une autre. Dans notre contexte, cette mesure n'est pas suffisante. En effet, la distance d'édition de deux séquences peut être très faible (1 opération) alors que les séquences sont en opposition de phase. Imaginons un centre qui suit les directives nationales, et qui a un comportement périodique. Un autre centre est en total décalage avec les directives nationale et se retrouve ainsi décalé d'une demi-période par rapport à la séquence précédente. La distance d'édition entre ces deux séquences est faible et ne traduit pas ce décalage.

Nous nous basons ici sur les distances les plus classiquement utilisées : la distance euclidienne, la distance de Manhattan et une mesure de similarité basée sur le cosinus.

Pour pouvoir établir des distances ou des mesures de similarité entre deux séquences, nous introduisons la notion de cellules comparables.

Définition 8 (Cellules comparables) *Deux cellules $c_1 = \langle (d_1, \ldots, d_n), \mu \rangle$ et $c_2 = \langle (d'_1, \ldots, d'_n), \mu' \rangle$ sont comparables si et seulement si $c_1.D_A = c_2.D_A$.*

Par exemple, les cellules $c_1 = \langle (Ouest, 1, Gold, opt1), 1200 \rangle$ et $c_2 = \langle (RA, 1, Gold, opt1), 900 \rangle$ sont comparables puisqu'elles ont la même restriction $(Gold, opt1)$ sur D_A. Par contre, les cellules c_1 et $c_3 = \langle (Sud, 1, Silver, opt2), 600 \rangle$ sont incomparables étant donné que leurs restrictions sur D_A sont différentes.

Dans cet article, nous nous situons dans un contexte de cube de données dense et nous supposons qu'il existe très peu de cellules vides. Pour calculer la distance entre deux blocs, nous essayons de construire des vecteurs de mesure, où chaque dimension sur les deux vecteurs correspond à des valeurs de mesures entre deux cellules comparables. L'algorithme 1 décrit comment deux blocs sont transformés en deux vecteurs contenant les valeurs de mesures des cellules comparables.

La représentation vectorielle de deux blocs permet d'appliquer les mesures de distances et de similarités telles que la distance euclidienne et le cosinus. Le calcul de la distance entre deux blocs nous permet de calculer la distance entre deux séquences :

Définition 9 (Distance entre 2 séquences) *Soient $s_1 = \langle b_1, b_2, \ldots, b_k \rangle$ et $s_2 = \langle b'_1, b'_2, \ldots, b'_k \rangle$ deux séquences multidimensionnelles, $dist$ une mesure de distance et Op un opérateur d'agrégation. La distance entre s_1 et s_2 se définit de la façon suivante :*
$d(s_1, s_2) = Op(dist(b_j, b'_j)$ *pour* $j = 1 \ldots k$

Dans cet article, nous utilisons les distances de Manhattan et euclidienne définies ci-dessous. Nous utilisons aussi la mesure de similarité basée sur le cosinus.

Algorithme 1: (TransBlocVec) Construction des vecteurs représentant les blocs

Data : b_1 et b_2 blocs
Result : Construction de deux vecteurs v_1 et v_2
begin
  $v_1 \leftarrow ()$
  $v_2 \leftarrow ()$
  foreach *cellule* $c_i \in b_1$ **do**
    if $\exists c_j \in b_2 \mid c_i$ *et* c_j *sont comparables* **then**
      $v_1.add(mesure(c_i))$
      $v_2.add(mesure(c_j))$
  return v_1, v_2
end

Distance de Manhattan : $Man(v_1, v_2) = \sum_{k=0}^{m} |v_{1_k} - v_{2_k}|$

Distance euclidienne : $Euclid(v_1, v_2) = \sqrt{\sum_{k=0}^{m} (v_{1_k} - v_{2_k})^2}$

Cosinus : $cos(v_1, v_2) = \frac{v_1.v_2}{||v_1||||v_2||} = \frac{\sum_{k=0}^{m} (v_{1_k} v_{2_k})}{\sqrt{\sum_{k=0}^{m} v_{1_k}^2 \sum_{k=0}^{m} v_{2_k}^2}}$

La définition 9 est suffisament générique pour appliquer n'importe quel opérateur d'agrégation pour calculer une distance entre deux séquences. La distance entre deux séquences peut être, par exemple, pour une mesure de distance donnée, la moyenne des distances entre chaque blocs, la médiane ou le max.

4.4 Comparaison de séquence par rapport à un ensemble de séquences

Pour déterminer si une séquence est un outlier, il est nécessaire de connaître sa similarité par rapport à toutes les autres séquences de la base. Nous établissons donc une matrice de distance représentant les distance entre chaque séquence de la base.

Sequence_Id	1	2	...	l
1	1	$sim(1,2)$	...	$sim(1,l)$
2	•	1	...	$sim(2,l)$
...	•	•	1	...
l	•	•	•	1

(a) Matrice de similarité

Sequence_Id	1	2	...	l
1	0	$d(1,2)$	...	$d(1,l)$
2	•	0	...	$d2,l)$
...	•	•	0	...
l	•	•	•	0

(b) Matrice de distance

FIG. 5 – *Comparaison d'une séquence par rapport aux autres*

Nous définissons la distance (resp. la similarité) d'une séquence par rapport à un ensemble de séquences, comme la moyenne des distances (resp. similarités) entre la séquence et les autres séquences. La distance d'une séquence s_α par rapport à un ensemble de séquences S est définie couramment définie dans la littérature de la façon suivante :

$$d(s_\alpha, S) = \frac{\Sigma_{i=1}^{i<\alpha} d(s_\alpha, s_i) + \Sigma_{j=\alpha+1}^{|s|} d(s_j, s_\alpha)}{|S| - 1}$$

Le calcul de la distance d'une séquence par rapport à un ensemble de séquences est primordial pour savoir si une séquence est un outlier ou non. Il est possible de définir un outlier par rapport à un seuil de distance fixé a priori par l'utlisateur. Définir ce seuil est très fastidieux et dépend fortement des séquences examinées. Il est, en conséquent, plus aisé pour l'utilisateur de définir un entier k qui correspond aux nombres d'outliers qu'il souhaite avoir. L'utilisateur veut voir ainsi les k séquences qui diffèrent le plus des autres.

Définition 10 (top n outliers) *Une séquence s_α est un top n outlier s'il n'existe pas plus de $n-1$ séquences telles que $d(s_i, C_{D_R}) > d(s_\alpha, C_{D_R})$*

Exemple 1 (top 1 outlier) *Etant donné le cube exemple (figure 2), nous voulons identifier la séquence la plus outlier. Il est nécessaire de calculer la matrice de distance entre les différentes séquences. La figure 6 (a) représente la matrice en fonction de la distance de Manhattan (moyenne). La figure 6 (b) représente la matrice de distance en fonction de la distance Euclidienne.*

Les figures 6 (c) et 6 (d) représente la distance moyenne d'une séquence par rapport aux autres en fonction de la mesure de distance utilisée.

Pour les deux types de distance, la séquence identifiée par $GEO = Sud.$ est considérée comme top 1 outlier puisque c'est la séquence la plus éloignée des autres.

Sequence_Id	$Paris$	RA	Sud	$Ouest$
$Paris$	0	243	1102	715
RA	•	0	1145	798
Sud	•	•	0	1437
$Ouest$	•	•	•	0

(a) Distance de Manhattan (moyenne)

Sequence_Id	$Paris$	RA	Sud	$Ouest$
$Paris$	0	127	576	365
RA	•	0	558	408
Sud	•	•	0	699
$Ouest$	•	•	•	0

(b) Distance Euclidienne (moyenne)

Sequence_Id	$Paris$	RA	Sud	$Ouest$
Distance	686	728	1228	983

(c) Distance par rapport à l'ensemble (Manhattan)

Sequence_Id	$Paris$	RA	Sud	$Ouest$
Distance	356	364	611	491

(d) Distance par rapport à l'ensemble (Euclidienne)

FIG. 6 – *Comparaison d'une séquence par rapport aux autres dans le cube exemple*

4.5 Algorithmes

Il s'agit ici de fournir les méthodes et outils à l'utilisateur pour qu'il soit capable, face à une séquence identifiée comme un outlier à un haut niveau de granularité, d'étudier plus en détail les sous-données associées à un niveau plus fin. Cette méthodologie permet de le guider dans sa recherche afin qu'il cible le plus directement possible les données susceptibles de l'intéresser.

Dans cette section, on notera :

- $S_{v_{R_i}}$ la séquence identifiée par $D_R = v_{R_i}$;
- $C_{D_R=v_R}$ le sous-cube relatif à v_R.

Chaque valeur v_R sur D_R identifie une séquence. Ainsi si une séquence est un top n outlier, alors ce sont les actions sur v_R qui sont anormales par rapport aux actions relatives aux autres valeurs sur D_R. Comme v_R n'est pas le niveau le plus fin dans la hiérarchie, il est toujours possible de se demander pourquoi v_R est outlier. Nous pouvons donc nous placer dans le sous-cube identifié par v_R et rechercher les top n outliers.

L'algorithme 2 permet d'extraire les top n outliers à un niveau d'agrégation donnée. Pour chaque séquence top n outlier identifiée par sa valeur v_R sur D_R, le processus est réitéré sur les sous-cubes identifiés par chaque valeur v_R jusqu'à arrivé au niveau d'agrégation le plus fin.

Algorithme 2: RechTopn

Data : C_{v_R} Cube de données, n entier, L ensemble, $dist$ une mesure de distance
Result : Séquences outliers à chaque niveau de granularité
begin
  Calculer la matrice de distance
  foreach *séquence* $S_{v_{R_i}} \in C_{v_R}$ *top* n *outlier* **do**
    $add(v_{R_i}, L)$
    if v_{R_i} *is not* $leaf$ **then**
      $RechTopn(C_{DrillDown(v_{R_i})}, n, L, dist)$
  return v_1, v_2
end

Pour $k = 1$, cet algorithme permet de proposer à l'utilisateur un chemin de navigation dans le cube afin d'identifier des séquences anormales par rapport à l'ensemble des données. Pour $k = 1$, le chemin regroupe les valeurs v_R dont les séquences associées sont des top 1 outliers à un niveau donné. Ce chemin part d'un niveau d'agrégation élevé et se termine au niveau d'agrégation le plus fin. Grâce à ce chemin, l'utilisateur peut directement aller sur la valeur v_R la plus fine, ou avancer pas à pas.

Pour $k \geq 1$, l'algorithme propose un arbre de navigation. En effet, il n'existe plus un seul chemin, mais plusieurs chemin. L'utilisateur peut ainsi visualiser les séquences anormales par l'intermédiaire de cet arbre. Il peut directement situer au niveau d'agrégation le plus fin (les feuilles), ou naviguer à travers les différents nœuds de l'arbre.

Une séquence peut être un top n outlier pour plusieurs raisons :

- Une séquence à un niveau inférieur est sensiblement différente des autres. La séquence a ainsi une importance dans le fait que la séquence agrégée du niveau supérieur est outlier. Dans ce cas là, l'algorithme 2 permet d'extraire ces différents outliers pour chaque niveau.
- Une grande partie des séquences du niveau inférieur sont sensiblement différentes du comportement général du niveau supérieur des séquences non outliers. Ainsi, une séquence qui suit le comportement général peut être considérée comme un top n outlier. Nous proposons donc de calculer la distance de cette séquence avec les autres séquences non outliers afin de voir si cette séquence suit le comportement général (bien le seul). Comme nous ne nous situons pas au même niveau d'agrégation, il est nécessaire de normaliser les séquences afin de calculer la distance entre deux séquences de niveaux d'agrégation différents.

Nous pouvons adapter l'algorithme 2 afin d'arrêter la construction de chemins quand on arrive sur les niveaux les plus fins ou quand les outliers d'un niveau suivent le comportement général du niveau supérieur. L'algorithme 3 prend en compte ce type de navigation et inclut donc la comparaison avec le comportement au niveau supérieur.

Exemple 2 *Naviguons à travers le cube exemple, à l'aide de l'extraction des top* 1 *outliers. Dans un premier temps, nous nous fixons au niveau le plus agrégé de la dimension de référence, c'est-à-dire, les*

Algorithme 3: RechTopnUp

Data : C_{v_R} Cube de données, n entier, L ensemble, $dist$ une mesure de distance
Result : Séquences outliers à chaque niveau de granularité bis
begin
  Calculer la matrice de distance
  foreach *séquence* $S_{v_{R_i}} \in C_{v_R}$ *top* n *outlier* **do**
    $add(v_{R_i}, L)$
    if v_{R_i} *is not* $leaf \wedge Normal(S_{v_{R_i}})$ *is not top* n *outlier in* $C_{roll\ up(v_R)}$ **then**
      $RechTopn(C_{DrillDown(v_{R_i})}, n, L, dist)$
  return v_1, v_2
end

fils de la racine (Paris, Ouest, RA, Sud). La séquence identifiée par Sud *est un top* 1 *outlier. Nous nous situons donc sur les fils de* Sud*, c'est-à-dire les séquences identifiées par* Nice, Perpigan, Marseille *et* Montpellier *comme indiqué par la figure 7.*

La séquence identifiée par Montpellier *est un top* 1 *outlier. Nous vérifions si cette séquence est outlier par rapport aux séquences* normales *du niveau supérieur (Paris, RA, Ouest) normalisées. Cette séquence est également un top* 1 *outlier dans ce contexte normalisé. Nous nous situons au niveau le plus fin, l'algorithme s'arrête donc.*

5 Expérimentations

Nous avons effectué des expérimentations sur un cube de données simulant des données se rapportant à des inscriptions d'étudiants dans un institut d'enseignement supérieur associées au projet Expedo STIC Asia 2005/2007. Le cube se compose d'une dimension géographique choisie comme dimension de référence, d'une dimension temporelle, et quatre dimensions d'analyse. Le cube au niveau d'agrégation le plus élevé contient 35000 cellules organisées en 20 séquences de blocs. L'opérateur d'agrégation utilisé sur la mesure est la somme.

Les expérimentations rapportées dans cette section montrent le temps d'exécution et le nombre d'outliers extraits en fonction du nombre de top k outliers recherchés au niveau d'agrégation le plus élevé. Nous étudions ensuite, de manière plus précise, les outliers extraits. Nous regardons s'ils se démarquent uniquement de leur sous-cube ou s'ils se démarquent aussi du comportement général du niveau supérieur. Ces expérimentations sont menées avec trois mesures différentes (distance euclidienne, distance de Manhattan, et mesure de similarité cosinus) et rois topérateurs d'agrégation différents (moyenne, médiane, et min), les expérimentations avec la médiane ne sont pas reportées dans cet article par manque d'espace, toutefois les expérimentations relatives à la médiane sont très similaires à celles avec la moyenne.

Les figures 8 et 9 montrent respectivement le temps d'exécution et le nombre d'outliers extraits en fonction du nombre de top k outliers recherchés. Le temps d'exécution et le nombre d'outliers augmentent proportionnellement avec le paramètre k.

Les figures restantes montrent le nombre de séquences outliers qui sont "totalement" outliers pour différents opérateurs d'agrégation (Fig 10 et 12), et le nombre de séquences outliers, pour différents opérateurs d'agrégation (Fig 11 et 13), qui suivent le comportement général du niveau supérieur par rapport au paramètre k. Quelques soient la mesure et l'opérateur d'agré-

GEO	Date	Fidélité	Offre	Mesure
Nice	1	Gold	opt1	200
Nice	1	Gold	opt2	250
Nice	1	Silver	opt1	150
Nice	1	Silver	opt2	175
Nice	1	Platinium	opt1	200
Nice	1	Platinium	opt2	225
Nice	2	Gold	opt1	250
Nice	2	Gold	opt2	225
Nice	2	Silver	opt1	250
Nice	2	Silver	opt2	225
Nice	2	Platinium	opt1	225
Nice	2	Platinium	opt2	300
Nice	3	Gold	opt1	300
Nice	3	Gold	opt2	187.5
Nice	3	Silver	opt1	180
Nice	3	Silver	opt2	250
Nice	3	Platinium	opt1	325
Nice	3	Platinium	opt2	250
Nice	4	Gold	opt1	350
Nice	4	Gold	opt2	125
Nice	4	Silver	opt1	200
Nice	4	Silver	opt2	300
Nice	4	Platinium	opt1	225
Nice	4	Platinium	opt2	275
Nice	5	Gold	opt1	375
Nice	5	Gold	opt2	125
Nice	5	Silver	opt1	172.5
Nice	5	Silver	opt2	300
Nice	5	Platinium	opt1	212.5
Nice	5	Platinium	opt2	275

(a) GEO Nice

GEO	Date	Fidélité	Offre	Mesure
Perpignan	1	Gold	opt1	210
Perpignan	1	Gold	opt2	300
Perpignan	1	Silver	opt1	130
Perpignan	1	Silver	opt2	165
Perpignan	1	Platinium	opt1	210
Perpignan	1	Platinium	opt2	220
Perpignan	2	Gold	opt1	270
Perpignan	2	Gold	opt2	215
Perpignan	2	Silver	opt1	200
Perpignan	2	Silver	opt2	225
Perpignan	2	Platinium	opt1	215
Perpignan	2	Platinium	opt2	275
Perpignan	3	Gold	opt1	285
Perpignan	3	Gold	opt2	197.5
Perpignan	3	Silver	opt1	190
Perpignan	3	Silver	opt2	235
Perpignan	3	Platinium	opt1	335
Perpignan	3	Platinium	opt2	245
Perpignan	4	Gold	opt1	320
Perpignan	4	Gold	opt2	150
Perpignan	4	Silver	opt1	195
Perpignan	4	Silver	opt2	250
Perpignan	4	Platinium	opt1	235
Perpignan	4	Platinium	opt2	270
Perpignan	5	Gold	opt1	350
Perpignan	5	Gold	opt2	145
Perpignan	5	Silver	opt1	160.5
Perpignan	5	Silver	opt2	245
Perpignan	5	Platinium	opt1	215.5
Perpignan	5	Platinium	opt2	275

(b) GEO Perpignan

GEO	Date	Fidélité	Offre	Mesure
Marseille	1	Gold	opt1	210
Marseille	1	Gold	opt2	300
Marseille	1	Silver	opt1	200
Marseille	1	Silver	opt2	185
Marseille	1	Platinium	opt1	195
Marseille	1	Platinium	opt2	230
Marseille	2	Gold	opt1	230
Marseille	2	Gold	opt2	220
Marseille	2	Silver	opt1	175
Marseille	2	Silver	opt2	225
Marseille	2	Platinium	opt1	220
Marseille	2	Platinium	opt2	305
Marseille	3	Gold	opt1	315
Marseille	3	Gold	opt2	177.5
Marseille	3	Silver	opt1	187.5
Marseille	3	Silver	opt2	245
Marseille	3	Platinium	opt1	315
Marseille	3	Platinium	opt2	255
Marseille	4	Gold	opt1	330
Marseille	4	Gold	opt2	175
Marseille	4	Silver	opt1	210
Marseille	4	Silver	opt2	250
Marseille	4	Platinium	opt1	215
Marseille	4	Platinium	opt2	280
Marseille	5	Gold	opt1	350
Marseille	5	Gold	opt2	125
Marseille	5	Silver	opt1	180
Marseille	5	Silver	opt2	200
Marseille	5	Platinium	opt1	205
Marseille	5	Platinium	opt2	285

(c) GEO Marseille

GEO	Date	Fidélité	Offre	Mesure
Montpellier	1	Gold	opt1	580
Montpellier	1	Gold	opt2	450
Montpellier	1	Silver	opt1	120
Montpellier	1	Silver	opt2	225
Montpellier	1	Platinium	opt1	270
Montpellier	1	Platinium	opt2	125
Montpellier	2	Gold	opt1	500
Montpellier	2	Gold	opt2	140
Montpellier	2	Silver	opt1	75
Montpellier	2	Silver	opt2	225
Montpellier	2	Platinium	opt1	250
Montpellier	2	Platinium	opt2	20
Montpellier	3	Gold	opt1	260
Montpellier	3	Gold	opt2	387.5
Montpellier	3	Silver	opt1	92.5
Montpellier	3	Silver	opt2	270
Montpellier	3	Platinium	opt1	0
Montpellier	3	Platinium	opt2	190
Montpellier	4	Gold	opt1	80
Montpellier	4	Gold	opt2	550
Montpellier	4	Silver	opt1	95
Montpellier	4	Silver	opt2	0
Montpellier	4	Platinium	opt1	275
Montpellier	4	Platinium	opt2	575
Montpellier	5	Gold	opt1	25
Montpellier	5	Gold	opt2	255
Montpellier	5	Silver	opt1	177
Montpellier	5	Silver	opt2	5
Montpellier	5	Platinium	opt1	247
Montpellier	5	Platinium	opt2	165

(d) GEO Montpellier

FIG. 7 – *Cube de données fils de Sud sous forme tabulaire*

gation utilisés, le nombre de séquences qui sont outliers par rapport à leur sous-cube et par rapport aux séquences du niveau supérieur augmente avec le paramètre k. Pour les séquences qui sont outliers dans leur sous-cube, mais qui suivent le comportement général du niveau supérieur, l'évolution est différente. Pour les distances de Manhattan et euclidiennes (médiane, min et moyenne), nous trouvons, à partir d'un certaine valeur de k, des séquences outliers qui suivent le comportement général du niveau supérieur. Ce nombre diminue ensuite dès que k tend vers le nombre de sous-cubes, c'est-à-dire quand on considère que toutes les séquences sont outliers. La mesure Cosinus identifie plus de séquences outliers à un niveau et communes au niveau supérieur.

Ces premières expérimentations sont encourageantes dans la mesure où le temps d'exécution de l'algorithme est quasiment linéaire par rapport au paramètre k et deux types d'outliers sont extraits, ce qui peut faciliter et enrichir la navigation de l'utilisateur dans le cube.

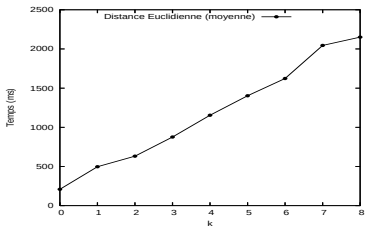

FIG. 8 – *Temps d'exécution en fonction du nombre de top k outliers recherchés*

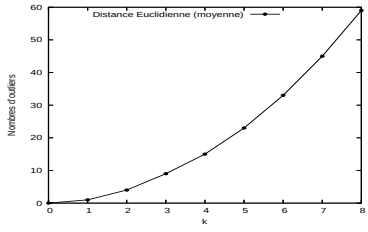

FIG. 9 – *Nombre d'outliers extraits en fonction du nombre de top k outliers recherchés.*

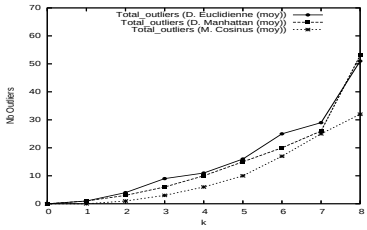

FIG. 10 – *Nombre d'outliers "totalement outliers" en fonction du nombre de top k outliers recherchés(moy.)*

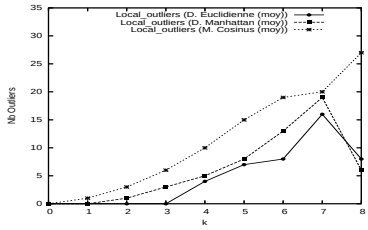

FIG. 11 – *Nombre d'outliers "localement" en fonction du nombre de top k outliers recherchés (moy.).*

6 Conclusions et perspectives

Dans cet article, nous avons proposé une méthode originale d'aide à la navigation dans un cube de données. Nous avons défini des algorithmes permettant de définir les top n séquences outliers à un niveau de granularité et d'étudier plus en détails les sous-données associées à un

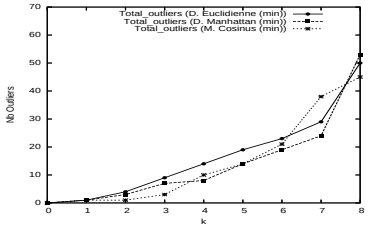

FIG. 12 – *Nombre d'outliers "totalement outliers" en fonction du nombre de top k outliers recherchés(min)*

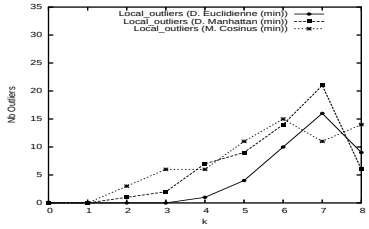

FIG. 13 – *Nombre d'outliers "localement" en fonction du nombre de top k outliers recherchés (min).*

niveau plus fin. Ainsi une séquence outlier à un niveau plus fin peut suivre ou ne pas suivre le comportement général du niveau supérieur. Des chemins de navigation sont ainsi proposés et permettent de guider l'utilisateur dans sa recherche afin qu'il cible le plus directement possible les données susceptibles de l'intéresser. Les algorithmes mis en œuvre sont suffisamment génériques pour être utilisés avec différentes mesures de distance et avec différents opérateurs d'agrégation. Dans cet article, nous utilisons trois mesures (distance euclidienne, distance de Manhattan et mesure de similarité cosinus) couplées à trois opérateurs (moyenne, médiane et min). Il est évidemment possible d'utiliser d'autres mesures. Les expérimentations menées sur des cubes de données réels ont permis de mettre en évidence l'intérêt de notre proposition. Ces résultats encourageants nous incitent à approfondir ces travaux par une analyse et une utilisation plus fine de la mesure. Dans le monde réel, les cubes de données contiennent en effet souvent un nombre important de cellules vides. La prise en compte des cellules vides est une véritable problématique de recherche. Dans certains cas, la cellule vide peut être considérée comme une mesure égale à 0. Toutefois, une cellule vide est rarement équivalente à zéro. En effet, il serait injuste de mettre un zéro à un étudiant non inscrit à un module, ou de considérer qu'aucune vente n'a eu lieu dans un magasin alors que le produit n'y est pas proposé.

Références

Aggarwal, C. C. et P. S. Yu (2001). Outlier detection for high dimensional data. In *SIGMOD Conference*, pp. 37–46.

Barnett, V. et T. Lewis (1994). *Outliers in Statistical Data*. John Wiley & Sons.

Breunig, M. M., H.-P. Kriegel, R. T. Ng, et J. Sander (2000). Lof : Identifying density-based local outliers. In *SIGMOD Conference*, pp. 93–104.

Chaudhuri, S. et U. Dayal (1997). An overview of data warehousing and olap technology. *ACM SIGMOD Record 26*(1), 65–74.

Fan, H., O. R. Zaïane, A. Foss, et J. Wu (2006). A nonparametric outlier detection for effectively discovering top-n outliers from engineering data. In *PAKDD*, pp. 557–566.

Hawkins, D. (1980). *Identification of Outliers*. Chapman and Hall, London.

Knorr, E. M. et R. T. Ng (1997). A unified notion of outliers : Properties and computation. In *KDD*, pp. 219–222.

Knorr, E. M. et R. T. Ng (1998). Algorithms for mining distance-based outliers in large datasets. In *VLDB*, pp. 392–403.

Lin, S. et D. E. Brown (2003). Criminal incident data association using the olap technology. In *ISI*, pp. 13–26.

Marcel, P. (1998). *Manipulations de Données Multidimensionnelle et Langages de Règles*. Ph. D. thesis, I.N.S.A. Lyon.

Messaoud, R. B., S. L. Rabaséda, O. Boussaid, et R. Missaoui (2006). Enhanced mining of association rules from data cubes. In *DOLAP*, pp. 11–18.

Pinto, H., J. Han, J. Pei, K. Wang, Q. Chen, et U. Dayal (2001). Multi-dimensional sequential pattern mining. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001*, pp. 81–88. ACM.

Plantevit, M., Y. W. Choong, A. Laurent, D. Laurent, et M. Teisseire (2005). M^2sp : Mining sequential patterns among several dimensions. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, et J. Gama (Eds.), *Knowledge Discovery in Databases : PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005, Proceedings*, Volume 3721 of *Lecture Notes in Computer Science*. Springer.

Plantevit, M., A. Laurent, et M. Teisseire (2006). Hype : Prise en compte des hiérarchies lors de l'extraction de motifs séquentiels multidimensionnels. In *EDA 2006, Actes de la deuxième journée francophone sur les Entrepôts de Données et l'Analyse en ligne, Versailles, 19 juin 2006*. Cépaduès.

Ramaswamy, S., R. Rastogi, et K. Shim (2000). Efficient algorithms for mining outliers from large data sets. In *SIGMOD Conference*, pp. 427–438.

Sarawagi, S., R. Agrawal, et N. Megiddo (1998). Discovery-driven exploration of olap data cubes. In *EDBT*, pp. 168–182.

Sun, P., S. Chawla, et B. Arunasalam (2006). Mining for outliers in sequential databases. In *SDM*.

Summary

Knowledge discovery in datacube is an important data mining problem with broad applications in datawarehousing in companies and scientific organization (biology, health). In this paper, we focus on atypical behaviors (called outliers) in such datacubes. We consider that users want to identify anormal sequences. For instance, a marketing manager would like to know which geographical area does not follow the same behavior as the others in order to find a better solution. We define similarity and distance measure to suit to such complex data. We define the associated algorithm which are carried out on several datacubes. Let us note that we consider very dense datacubes and then, the knowledge discovery problem becomes more complicated.