Vers une méthode automatique de construction de hiérarchies contextuelles

Dino Ienco[*****], Yoann Pitarch[**],
Pascal Poncelet[***],
Maguelonne Teisseire[*****]

[*]Irstea, UMR TETIS, 500 rue Jean-Francois Breton, F-34093 Montpellier, France
{Dino.Ienco,Maguelonne.Teisseire}@teledetection.fr,
[**] Département Informatique, Université d'Aalborg, Dk-9000 Aalborg, Danemark
ypitarch@cs.aau.dk
[***]LIRMM, 161 rue Ada, F-34090 Montpellier, France
Pascal.Poncelet@lirmm.fr

Résumé. Dans de nombreux domaines (*e.g.*, fouille de données, entrepôts de données), l'existence de hiérarchies sur certains attributs peut être extrêmement utile dans le processus analytique. Toutefois, cette connaissance n'est pas toujours disponible ou adaptée. Il est alors nécessaire de disposer d'un processus de découverte automatique pour palier ce problème. Dans cet article, nous combinons et adaptons des techniques issues de la théorie de l'information et du clustering pour proposer une technique *orientée données* de construction automatique de taxonomies. Les deux principaux avantages d'une telle approche sont son caractère totalement non-supervisé et l'absence de paramètre utilisateur à spécifier. Afin de valider notre approche, nous l'avons appliquée sur des données réelles et avons conduit plusieurs types d'expérimentation. D'abord, les hiérarchies obtenues ont été expertisées pour en examiner le pouvoir informatif. Ensuite, nous avons évalué l'apport de ces taxonomies comme support à des tâches de fouille de données nécessitant une définition hiérarchique des valeurs d'attributs : l'extraction de séquences fréquentes multidimensionnelles et multi-niveaux ainsi que la construction de résumés de tables relationnelles. Les résultats obtenus permettent de conclure quant à l'intérêt de notre approche.

1 Introduction

Les taxonomies fournissent une organisation hiérarchique des données couramment exploitée dans de nombreux domaines tels que la biologie (Ashburner et al. (2000)), le e-commerce (Kohavi et Provost (2001)), le web sémantique (T. Berners-Lee et Lassila (2001)). Elles permettent de réaliser différents types d'analyse de données telles que l'anonymisation (Samarati (2001)), l'exploration d'entrepôts de données (Pitarch et al. (2010)) ou le résumé de données (Candan et al. (2010)). Par exemple, produire un résumé de données peut être particulièrement utile pour un décideur qui souhaiterait obtenir un rapide aperçu des ventes à l'échelle nationale plutôt que de considérer les ventes individuelles de chaque magasin. Traditionnellement,

les hiérarchies sont définies manuellement par des experts. Malheureusement, cette spécification peut être très longue et exige des ressources humaines considérables pouvant parfois faire défaut. Face au nombre croissant d'attributs pouvants être hiérarchisés, la proposition de méthodes orientées données et non supervisées pour définir automatiquement des hiérarchies représente alors un enjeu majeur. Dans certaines travaux (Zhang et Honavar (2004); desJardins et al. (2005)), les auteurs construisent des hiérarchies contextuelles pour améliorer les performances de classifieurs basés sur des réseaux Bayésiens. Dès lors, ces approches dites *orientées classification* nécessitent des ressources humaines non négligeables et ne construisent les hiérarchies qu'en fonction de la classe associée. Notre travail se veut plus générique dans la mesure où il ne requiert aucune connaissance experte a priori et peut être utilisé comme brique à toute approche réclamant une vision hiérarchisée des valeurs d'attributs. A notre connaissance, très peu d'approches abordent cette problématique malgré le grand nombre d'applications pouvant en bénéficier.
Ainsi, les principales contributions de cet article sont : (1) une approche originale et efficace pour la construction automatique de hiérarchies contextuelles, (2) un processus indépendant des connaissances externes et (3) une méthode sans aucune spécification de paramètre. Afin de valider notre approche, nous l'avons appliquée sur des données réelles et avons conduit plusieurs types d'expérimentation. D'abord, nous avons évalué l'apport de ces taxonomies comme support à deux tâches de fouille de données nécessitant une définition hiérarchique des valeurs d'attributs : (1) l'extraction de séquences fréquentes multidimensionnelles et multiniveaux (Plantevit et al. (2010)), méthode qui exploite l'abstraction des données pour extraire des connaissances plus représentatives au sein d'une base de séquences ; (2) la construction de résumé de données basée sur les travaux de Samarati (2001) pour la préservation de la vie privée. Ensuite, les hiérarchies obtenues ont été expertisées pour en examiner le pouvoir informatif. La suite de cet article est organisée de la manière suivante. Section 2, nous présentons un cas d'étude. Puis, le cœur de notre méthode est décrit section 3. Les expérimentations menées sont rapportées section 4. Enfin, nous concluons et dressons quelques perspectives dans la dernière section.

2 Cas d'étude

Considérons la base de données exemple présentée dans le tableau 1 qui recense trois informations par individu (*Ville*, *Sport favori* et *Type de cuisine*). Nous supposons ces attributs définis sur un domaine discret. L'attribut *Ville* désigne le lieu de résidence de l'individu et est défini sur {Turin (Italy), Porto (Portugal), Miami (USA), Denver (USA)}. L'attribut *Sport favori* désigne le sport que l'individu pratique le plus et est défini sur {ski, surf, snowboard, beach volley}. Enfin, l'attribut *Type de cuisine* désigne les habitudes gastronomiques préférées d'un individu et prend ses valeurs dans {viande, poisson}.
Afin d'illustrer l'intérêt de générer des hiérarchies contextuelles, posons nous la question de résumer cette base à l'aide des hiérarchies des données (Candan et al. (2010)). L'objectif est de produire un résumé compact et informatif de la base source en agrégeant les données. Une table contiendra alors les même colonnes que la table initiale, à laquelle sera ajouté une colonne *Count* représentant le nombre de n-uplets associés à chaque ligne du résumé. Considérons les hiérarchies *usuelles* sur les attributs *Ville* (figure 1(a)) et *Sport favori* (figure 1(b)).

Ville	*Sport favori*	*Type de cuisine*
Denver	ski	viande
Porto	surf	poisson
Miami	beach volley	poisson
Porto	surf	poisson
Porto	surf	viande
Denver	ski	viande
Porto	surf	poisson
Turin	snowboard	viande
Miami	surf	poisson
Denver	snowboard	viande
Denver	snowboard	viande
Miami	surf	viande
Porto	surf	poisson
Turin	ski	viande
Porto	beach volley	poisson
Miami	surf	poisson
Denver	ski	poisson
Turin	ski	viande

TAB. 1 – *Base de données exemple*

Il est intéressant de noter que l'information géographique n'est pas vraiment adéquate pour la tâche proposée. Pour s'en convaincre définitivement, le tableau 2 présente le résumé obtenu à partir de cette hiérarchie. Notons que dans cet article, nous représentons une généralisation par l'ensemble des éléments y appartenant. Par exemple, en considérant la hiérarchie présentée dans la figure 1(a), la généralisation des éléments *Denver* et *Miami* est alors {*Denver*, *Miami*}). En effet, notre objectif n'est pas de labelliser les nœuds d'une hiérarchie [1] mais bien de les découvrir.

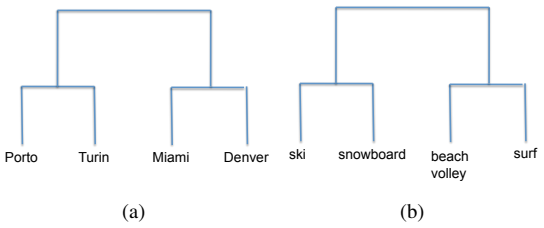

FIG. 1 – *Hiérarchies disponibles sur les attributs (a) Ville et (b) Sport favori*

Une analyse approfondie de la base de données exemple révèle que les habitudes des personnes vivant à *Turin* et à *Denver* sont très proches tout comme le sont celles des personnes vivant à *Porto* et *Miami*. Une explication à ces ressemblances est que *Turin* et *Denver* sont des villes de montagne alors que *Porto* et *Miami* sont des villes côtières. A partir de ces observations, nous construisons une hiérarchie (contextuelle) alternative représentée dans la figure 2. Alors, en s'appuyant sur cette hiérarchie alternative ainsi que sur la hiérarchie associée à l'attribut *Sport favori*, nous pouvons produire un nouveau résumé présenté dans le tableau 3. Nous observons que la prise en compte de la hiérarchie contextuelle permet la génération d'un

1. Nous encourageons les lecteurs intéressés par cette thématique à consulter les travaux de Carmel et al. (2009) ou Treeratpituk et Callan (2006).

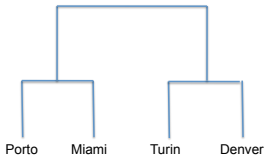

FIG. 2 – *Une hiérarchie contextuelle sur l'attribut Ville*

Ville	*Sport favori*	*Type de cuisine*	*Count*
{Denver, Miami}	{ski, snowboard}	{viande}	4
{Denver, Miami}	{ski, snowboard}	{poisson}	1
{Denver, Miami}	{surf, beach volley}	{viande}	1
{Denver, Miami}	{surf, beach volley}	{poisson}	3
{Porto, Turin}	{ski, snowboard}	{viande}	3
{Porto, Turin}	{surf, beach volley}	{viande}	1
{Porto, Turin}	{surf, beach volley}	{poisson}	5

TAB. 2 – *Résumé obtenu à partir des hiérarchies usuelles*

résumé plus compact mais tout aussi informatif. En effet, intuitivement, ce résumé capture la connaissante sous-jacente des données sources ainsi que l'information contextuelle qui en découle.

Ville	*Sport favori*	*Type de cuisine*	*Count*
{Denver, Turin}	{ski, snowboard}	{viande}	8
{Denver, Turin}	{ski, snowboard}	{poisson}	1
{Porto, Miami}	{surf, beach volley}	{viande}	2
{Porto, Miami}	{ski, beach volley}	{poisson}	9

TAB. 3 – *Résumé obtenu à partir de la hiérarchie contextuelle sur l'attribut Ville et de la hiérarchie usuelle sur l'attribut Sport favori*

3 Extraction de hiérarchies contextuelles

Pour un attribut donné, l'objectif est de générer une hiérarchie qui soit représentative et adaptée au jeu de données manipulées. Pour cela, nous adoptons un regroupement selon un clustering hiérarchique. Classiquement, une matrice de distances entre objets est construite et la représentation hiérarchique est générée en regroupant les objets les plus proches. Dans notre contexte, il est ainsi nécessaire de définir une distance permettant de mesurer la proximité entre chaque paire de valeurs d'un même attribut. Cette étape n'est pas triviale car les attributs manipulés sont catégoriels. Pour résoudre ce problème, nous utilisons une technique récente, *DILCA* ((DIstance Learning for Categorical Attributes), proposée dans Ienco et al. (2012), qui permet de calculer une matrice de distances intra-attribut en tirant profit des interactions avec les autres attributs du jeu de données. Chaque matrice obtenue représente alors la similarité entre chaque paire de valeurs de l'attribut considéré. L'approche que nous proposons se décompose donc en deux étapes : (1) le calcul de la matrice de distances et (2) l'application du clustering hiérarchique afin d'en extraire une hiérarchie.

3.1 Calcul de distance intra-attribut

Tout d'abord, nous rappelons le fonctionnement de *DILCA*, un système pour calculer une distance entre chaque paire de valeurs d'un attribut (Ienco et al. (2012)). Cette approche repose sur des techniques issues de la théorie de l'information. En particulier, elle fait usage de l'incertitude symétrique (Yu et Liu (2003)) qui peut être considérée comme une version normalisée de l'information mutuelle. Précisément, pour un attribut donné, cette approche fonctionne en deux temps : (1) sélection du contexte et (2) utilisation de ce contexte pour calculer la matrice de distances. Ce processus est répété pour chaque attribut catégoriel du jeu de données aboutissant ainsi à un ensemble de matrices de distance intra-attribut.

3.1.1 Selection du contexte

Soit Y un attribut sur lequel nous désirons construire une hiérarchie. L'ensemble des attributs participant au contexte de Y est noté $Contexte(Y)$. Dans Ienco et al. (2012), les auteurs proposent une technique entièrement automatique pour déterminer cet ensemble d'attributs. Plus précisément, l'attribut cible,*i.e.,* l'attribut Y, est considéré comme un attribut classe afin de réaliser une tâche supervisée de sélection d'attributs. La sélection d'attributs permet d'extraire, à partir de l'ensemble des attributs restants, une représentation des données qui permet la prédiction des valeurs de la variable Y. Les attributs sélectionnés par cette méthode, *i.e.,* ceux appartenant au contexte, sont donc liés à l'attribut Y. En particulier, les attributs du contexte de Y sont les plus informatifs pour réaliser la prédiction sur l'attribut Y. Si l'on considère l'exemple présenté dans la section 2, le contexte de l'attribut $Ville$ est $Contexte(Ville) = \{Sport\ favori\}$. L'attribut *Type de cuisine* n'est pas retenu car il est considéré comme peu informatif pour déduire les villes.

3.1.2 Utilisation du contexte pour calculer les distances intra-attribut

Soit $Contexte(Y)$ l'ensemble des attributs déterminé comme précédemment décrit pour l'attribut Y. La distance entre une paire de valeurs de l'attribut Y est calculé ainsi :

$$d(y_i, y_j) = \sqrt{\frac{\sum_{X \in Contexte(Y)} \sum_{x_k \in X} (P(y_i|x_k) - P(y_j|x_k))^2}{\sum_{X \in Contexte(Y)} |X|}}$$

où $P(y_j|x_k)$ est la probabilité conditionnelle que la valeur y_j de l'attribut Y apparaisse dans le même n-uplet que la valeur x_k de l'attribut X (où $X \in Contexte(Y)$). $|X|$ désigne la cardinalité de l'attribut X. A partir du cas d'étude présenté dans cet article, nous obtenons la matrice de distances présentée dans le tableau 4. Nous remarquons que ces distances correspondent à la hiérarchie contextuelle présentée dans la figure 2 qui regroupe *Denver* (resp. *Porto*) et *Turin* (resp. *Miami*) au premier niveau.

Valeur	**Valeur**	**dist.**	**Valeur**	**Valeur**	**dist.**
Denver	Porto	0.60	**Porto**	**Miami**	**0.13**
Denver	Miami	0.55	Porto	Turin	0.47
Denver	**Turin**	**0.19**	Miami	Turin	0.41

TAB. 4 – *Matrice de distance intra-attribut associée à l'attribut* Ville

3.2 Construction des hiérarchies

Nous utilisons le résultat de l'étape précédente pour extraire et générer les hiérarchies contextuelles. A partir des matrices de distances calculées, l'algorithme de clustering hiérarchique Ward (Anderberg (1973)) est utilisé pour construire les hiérarchies. Nous avons retenu cet algorithme pour sa simplicité et parce qu'il n'introduit pas de biais dans le processus en favorisant des regroupements.

Algorithm 1 $WARD(distM)$

1: **for all** $i = 1$ to $|distM|$ **do**
2: $C_{1i} = \{i\}$
3: **end for**
4: $C1 = \{C_{11}, ..., C_{1n}\}$
5: $i = 1$
6: **while** $|C_i > 1|$ **do**
7: **for** $j = 1$ **to** $|C_i|$ **do**
8: **for** $k = j + 1$ **to** $|C_i|$ **do**
9: $d[j, k] = distM[C_{ij}, C_{ik}]$
10: **end for**
11: **end for**
12: (s,r) = $argmin(d[j, k])$
13: **for** $j = 1$ **to** $|C_i|$ **do**
14: **if** $j \neq r$ **and** $j \neq s$ **then**
15: $C_{i+1,j} = C_{i,j}$
16: **else if** $j = r$ **then**
17: $C_{i+1,j} = C_{ir} \cup C_{is}$
18: **end if**
19: **end for**
20: **end while**

La méthode Ward présentée dans l'algorithme 1 est une méthode hiérarchique agglomérative gloutonne produisant un diagramme, le dendogramme, qui enregistre les différentes étapes de fusion. Le processus est itératif et nécessite une matrice de distances en entrée. Initialement, autant de clusters sont créés que le nombre d'objets dans la matrice, *i.e.,* la cardinalité de l'attribut considéré. Le processus est ensuite guidé par une fonction objective qui mesure la cohésion au sein des clusters générés (ligne 12). Cette fonction repose sur le calcul d'un score global qui évalue la qualité de chaque clustering potentiel. A chaque étape, les deux clusters qui permettent l'augmentation du score global sont fusionnés (lignes 13 à 19). Ce processus implique de calculer le score associé à chaque regroupement potentiel. Parmi eux, le meilleur regroupement choisi est celui qui minimise l'augmentation du score global (ligne 12). A la fin du processus, une hiérarchie est ainsi obtenue d'une manière totalement non supervisée. Nous répétons alors cette méthode pour tous les attributs pour lesquels nous souhaitons calculer une hiérarchie contextuelle.

4 Expérimentations

Nous proposons une évaluation de l'approche proposée selon deux points de vue complémentaires. Dans un premier temps, nous exploitons les hiérarchies contextuelles pour réaliser deux tâches de fouilles de données : (1) l'extraction de séquences multidimensionnelles et multi-niveaux fréquentes et (2) le résumé de tables relationnelles. Pour ces deux tâches, nous confrontons les résultats obtenus avec des hiérarchies usuelles d'une part et contextuelles d'autre part. Dans un second temps, nous confrontons les hiérarchies contextuelles obtenues

au regard d'experts des données afin d'en évaluer leur qualité. Tout d'abord, nous décrivons brièvement les jeux de données réelles utilisés.

4.1 Description des jeux de données

Le jeu de données MALI recueille aussi bien des informations statiques (relevés terrains) que des informations dynamiques (séries temporelles multivariées) associées à 980 fermes du Mali. Extraites à partir d'images satellites, les séries temporelles représentent des mesures couramment utilisées pour la surveillance de zones cultivées. Chaque série temporelle, d'une longueur de 11 estampilles temporelles, est décrite grâce à 5 descripteurs. Les 5 attributs statiques (type de sol, distance au village, distance à la rivière, pluviométrie et groupe ethnique propriétaire) permettent de caractériser chaque ferme du jeu de données. Les hiérarchies contextuelles sont calculées sur ces attributs statiques.

Le jeu de données ADULT provient d'un répertoire public [2] courament utilisé pour l'évaluation d'algorithmes de fouille. Les données sont issues d'un recensement et ont largement été utilisées pour évaluer différents algorithmes de classification et de k-anonymisation. Les paramètres adoptés sont identiques à ceux de Iyengar (2002). Nous obtenons ainsi un jeu de données de 8 attributs. Dans cet article, nous évaluons notre approche sur un échantillon de 4500 n-uplets. Notons que l'attribut *Age* a été discrétisé en 8 intervalles de tailles égales. Les hiérarchies de références sont reprises de Iyengar (2002) et nous calculons les hiérarchies contextuelles sur la totalité des attributs du jeu de données.

4.2 Hiérarchies contextuelles pour la découverte de motifs séquentiels multi-niveaux

Les hiérarchies sont de plus en plus utilisées dans le contexte de l'extraction de séquences fréquentes car elles permettent de capturer les tendances générales d'un jeu de données. En particulier, certains travaux, *e.g.,* Plantevit et al. (2010), Pinto et al. (2001), se sont concentrés sur l'exploitation des hiérarchies pour extraire des motifs séquentiels multidimensionnels et multi-niveaux. Schématiquement, ces approches permettent d'extraire des séquences qui représentent chacune un sous-ensemble du jeu de données. La taille minimale de ce sous-ensemble est indiquée par un paramètre utilisateur : le *support minimum*, noté $minSupp$ dans la suite. L'avantage d'intégrer les hiérarchies dans cette tâche est qu'elles permettent l'extraction de séquences plus longues et plus représentatives du comportement général des individus de la base. Par exemple, en s'appuyant sur la hiérarchie présentée dans la figure 1, l'item $(Porto,\ Surf)$ [3] peut ne pas être considéré comme fréquent alors qu'un item plus général tel que $(\mathbf{Europe},\ Surf)$ aurait pu l'être puisqu'il représente l'ensemble des villes européennes où le surf est pratiqué. Les expérimentations réalisées visent à étudier les bénéfices liés à l'utilisation de hiérarchies contextuelles lors de l'extraction de tels motifs. L'algorithme de recherche de motifs séquentiels multidimensionnels multi-niveaux utilisé est M3SP (Plantevit et al. (2010)). Les hiérarchies ont été obtenues de 3 façons distinctes : (1) à partir de notre approche, M3SP$_{CAVT}$, (2) à partir de connaissances expertes, M3SP$_{EXP}$ et (3) en les générant aléatoirement, M3SP$_{RAN}$. Pour cette dernière méthode, nous avons procédé comme suit.

2. http ://archive.ics.uci.edu/ml/
3. Cet item désigne le fait que le surf est pratiqué à Porto.

Pour un attribut donné, nous avons d'abord considéré un clustering tel que chaque valeur soit comprise dans n clusters distincts. Ces clusters sont ensuite fusionnés aléatoirement deux à deux pour simuler un processus agglomératif générant une hiérarchie binaire. Afin de fournir des résultats statistiquement pertinents, nous répétons ce procédé 30 fois et expérimentons l'extraction de séquences fréquentes autant de fois. Nous évaluons alors les performances de ces hiérarchies en considérant le nombre moyen de séquences fréquentes et la déviation standard qui en résulte. L'extraction de séquences fréquentes a été effectuée avec des support allant de 50% à 90% par palier de 10%. L'évaluation des performances se concentre sur l'étude de deux indicateurs : le nombre de motifs extraits et la proportion de motifs de taille supérieure à 1. Les résultats sur le nombre de motifs extraits sont reportés dans le tableau 5. Il existe une grande différence entre les résultats avec M3SP$_{RAN}$ et les deux autres techniques. Ceci est une observation intéressante car elle témoigne des distributions très similaires de M3SP$_{CAVT}$ et M3SP$_{EXP}$. Cela souligne le fait que notre approche propose une hiérarchie pouvant être une bonne alternative à une hiérarchie fournie par les experts (quand elle existe).

	# Sequences		
MinSupp	**M3SP$_{EXP}$**	**M3SP$_{CAVT}$**	**M3SP$_{RAN}$**
50%	105	183	116086.66±63705.46
60%	46	75	51206.66±25839.83
70%	28	27	20640.0±10227.56
80%	12	18	6843.33±3606.63
90%	12	10	1990.0±1306.73

TAB. 5 – *Nombre de motifs extraits avec différents type de hiérarchies*

Une des aptitudes essentielles à un algorithme de fouille de données est sa capacité à fournir un résultat à la fois concis et représentatif des données Vreeken et al. (2011). Afin de mesurer ce point, nous dénombrons le nombre de motifs contenant plus d'un item. Nous considérons ces séquences car ce sont elles qui représentent la temporalité au sein des données. Nous évaluons ainsi la capacité des hiérarchies étudiées à capturer les principales tendances séquentielles. Les résultats de cette série d'expérimentations sont reportés dans le tableau 6. Le comportement de M3SP$_{CAVT}$ est significativement différent du comportement de M3SP$_{RAN}$ témoignant de la grande différence structurelle de ces deux types de hiérarchies. En comparant M3SP$_{CAVT}$ et M3SP$_{EXP}$, on constate que notre approche permet l'extraction d'un nombre plus important de séquences de taille supérieure à 1. Ce résultat très appréciable est particulièrement évident lorsque l'on considère des valeurs élevées de $minSupp$. En conclusion de cette série d'expérimentations, nous pouvons souligner que les hiérarchies générées par notre approche se comportent au moins aussi bien que les hiérarchies fournies par des experts pour la phase d'extraction de motifs multidimensionnels.

4.3 Hiérarchies contextuelles pour le résume de tables relationnelles

Il s'agit maintenant de mesurer l'impact des hiérarchies contextuelles lors de la construction de résumés de tables relationnelles. Comme décrit dans Candan et al. (2010), l'objectif de cette tâche est d'exploiter les hiérarchies des données afin de fournir une représentation condensée mais représentative d'une table. Chaque n-uplet de la table résumée devra alors représenter au moins k n-uplets de la table d'origine. Ce problème présente de fortes similitudes avec celui de la k-anonymisation. Pour cette raison, nous adoptons l'algorithme présenté dans

MinSupp	**M3SP**$_{EXP}$	**M3SP**$_{CAVT}$	**M3SP**$_{RAN}$
50%	3.8%	5.46%	1.91±1.35%
60%	2.17%	5.33%	2.39±2.17%
70%	0%	0%	3.54±3.29%
80%	0%	27.77%	5.92±6.04%
90%	0%	20.0%	8.23±9.78%

TAB. 6 – *Proportion de motifs de taille > 1 extraits avec différents type de hiérarchies*

	$MinGen_{CAVT}$		$MinGen_{EXP}$	
K	**Perte d'info**(x10^3)	**# n-uplets**	**Perte d'info**(x10^3)	**# n-uplets**
10	**28.78**	67	30.44	55
20	**35.21**	33	718.37	37
30	**35.21**	33	35.36	22
40	37.798	18	**35.36**	22
50	**34.66**	20	35.36	22
60	**34.87**	14	35.36	22
70	**34.87**	14	723.29	16
80	**34.87**	14	723.29	16
90	**37.25**	10	723.29	16
100	**37.25**	10	723.29	16

TAB. 7 – *Perte d'information et nombre de tuples de la table résumée pour $MinGen_{CAVT}$ et $MinGen_{EXP}$*

Samarati (2001). Nous confrontons les résultats obtenus avec les hiérarchies contextuelles, $MinGen_{CAVT}$ et les hiérarchies fournies par les experts, $MinGen_{EXP}$. L'algorithme nécessite la définition de deux paramètres. Le premier, k, désigne le nombre minimum de n-uplets non distinguables sur les attributs *sensibles* (nommés *quasi-identifiants* dans Samarati (2001)). Ces attributs sont ceux sur lesquels le processus est exécuté et pour lesquels une hiérarchie est nécessaire. Pour cette étude, nous supposons que tous les attributs sont sensibles et faisons évoluer le paramètre k de 10 à 100 par palier de 10. Le second paramètre recquis est le pourcentage de perte tolérée pour que l'algorithme puisse atteindre son objectif final. Bien que plutôt atypique dans le contexte du résumé de tables, il est raisonnable de supposer que le résumé doive représenter la majorité des données et non son intégralité. Nous fixons ainsi ce paramètre à 1%. L'évaluation de la qualité du résumé est réalisée au travers d'une mesure non-uniforme d'entropie présentée dans Gionis et Tassa (2009) pour mesurer la perte d'information. Cette mesure suppose que les valeurs des attributs ne respectent pas une distribution uniforme et prend ses valeurs dans l'intervalle $[0; +\infty[$. Un résumé sera considéré comme de bonne qualité si cette mesure est proche de 0.

Le tableau 7 présente les résultats obtenus. Pour chaque type de hiérarchies, la perte d'information ainsi que le nombre distincts de n-uplets présents dans le résumé sont indiqués. L'analyse pour les hiérarchies contextuelles est intéressant. Par exemple, lorsque k se situe entre 20 et 70, la perte d'information associée aux hiérarchies contextuelles est environ 20 fois plus faible que dans le cas de hiérarchies expertes. Bien que moins significatifs, les résultats obtenus pour d'autre valeurs de k confirment la supériorité des hiérarchies contextuelles. En outre, les hiérarchies contextuelles permettent d'obtenir des résumés semblables ou plus compacts que ceux obtenus avec les hiérarchies fournies par les experts. Cela est particulièrement vrai lorsque k est supérieur à 40. Il est intéressant de noter que la perte d'information n'est pas proportionnelle à la taille du résumé (*e.g.,* quand $k = 30$, la taille du résumé avec $MinGen_{CAVT}$ est légèrement plus importante qu'avec $MinGen_{EXP}$ mais permet de

conserver bien plus d'informations). De cette étude, nous pouvons conclure que les hiérarchies contextuelles produisent des résultats au moins identiques et si ce n'est meilleurs pour réaliser un résumé représentatif d'une table relationnelle. Ces hiérarchies sont donc particulièrement adaptées quand (1) les hiérarchies usuelles sont trop générales et ne reflètent pas les particularités des données ou (2) lorsqu'aucune hiérarchie n'est disponible.

4.4 Evaluation qualitative

Dans cette section, nous analysons qualitativement quelques extraits de hiérarchies contextuelles obtenues sur le jeu de données MALI. Nous avons ainsi proposé à des scientifiques spécialisés en télédétection de comparer les hiérarchies qu'ils avaient fournies avec les hiérarchies contextuelles extraites automatiquement par notre approche. Le but d'une telle évaluation est double. Tout d'abord, il s'agit de savoir si les hiérarchies contextuelles permettent d'extraire de nouvelles connaissances. Ensuite, il est intéressant d'analyser comment les spécialistes du domaine peuvent valider de telles hiérarchies. Une partie de la hiérarchie contextuelle associée au type de culture est présentée dans la figure 3. Par souci de clarté, nous ne représentons que les deux premiers niveaux de la hiérarchie contextuelle. La mise à disposition de cette hiérarchie à l'expert a permis d'identifier quelques corrélations bien connues dans le domaine. Par exemple, elle a pu valider la pertinence du groupe (a) où le *sorgho* et le *mil* sont des cultures poussant dans des environnements semblables.

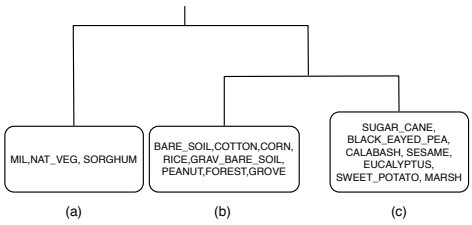

FIG. 3 – *Deux niveaux d'une hiérarchie contextuelle sur le type de culture selon* $CAVT$

De plus, le *coton* et le *maïs* appartiennent tous deux au groupe (b) car il sont usuellement cultivés ensemble. Enfin, l'expert a pu identifier le fait que les cultures appartenant au groupe (c) interviennent dans la nutrition des populations humaines. Des résultats très intéressants ont également été obtenus sur la hiérarchie associée au type de sol, entièrement décrite dans la figure 4. Une première observation faite par l'expert est que la hiérarchie produite est significativement différente de la hiérarchie usuelle. Ce résultat surprenant est en fait cohérent pour la zone géographique étudiée. En effet, le *EC* (Sol d'érosion) et le *GR* (Sol gravillonnaire), même s'ils sont structurellement différents, ont été regroupés ensemble. Dans le contexte agricole africain, aucune culture ne peut pousser sur ces deux types de sol qui correspondent à de l'herbe en friche. Notre méthode regroupe également le *GR_su* (sol superficiel gravillonnaire) et *SU* (sols superficiel) qui sont des types de sols particulièrement bien adaptés à la culture de *riz* ou d' *arachide*. Des résultats tout aussi intéressants ont également été produits sur les hiérarchies associées aux autres attributs de ce jeu de données. Cette étude montre donc la pertinence de notre proposition qui intègre de manière complètement non supervisée les in-

formations contextuelles afin de produire des hiérarchies faisant particulièrement sens dans les jeux de données étudiés.

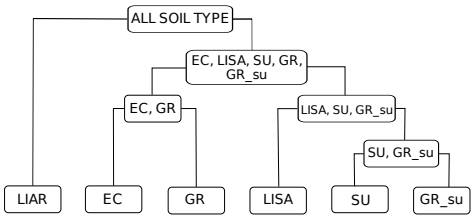

FIG. 4 – *Une hiérarchie pour les types de sol obtenue avec $CAVT$*

5 Conclusion

Cet article présente une méthode originale de génération automatique de hiérarchies contextuelles à partir des données sources. Afin de souligner la qualité des hiérarchies générées, indépendante des applications qui peuvent en être faites, nous avons conduit une étude expérimentale approfondie. Nous avons ainsi confronté les hiérarchies contextuelles aux hiérarchies fournies par les experts dans le cadre de l'extraction de motifs séquentiels multi-niveaux et de la construction de résumés de tables relationnelles. En outre, les hiérarchies obtenues ont également été expertisées par des spécialistes des données traitées. Les résultats obtenus à l'issue de ces expérimentations ont montré que l'utilisation de telles hiérarchies est une alternative extrêmement pertinente lorsque les hiérarchies expertes sont trop générales ou non disponibles. A court terme, nous prévoyons de confronter nos hiérarchies à d'autres tâches de fouille de données. A plus long terme, il serait intéressant d'étudier de nouvelles techniques de clustering hiérarchiques pour construire non plus des hiérarchies à structure binaire mais à structure n-aire plus concise.

Références

Anderberg, M. R. (1973). Cluster analysis for applications.

Ashburner, M. et al. (2000). Gene ontology : tool for the unification of biology. *The Gene Ontology Consortium. Nat Genet. 25*(1), 25–29.

Candan, K. S., M. Cataldi, et M. L. Sapino (2010). Reducing metadata complexity for faster table summarization. In *EDBT*, pp. 240–251.

Carmel, D., H. Roitman, et N. Zwerdling (2009). Enhancing cluster labeling using wikipedia. In *SIGIR*, pp. 139–146.

desJardins, M., P. Rathod, et L. Getoor (2005). Bayesian network learning with abstraction hierarchies and context-specific independence. In *ECML*, pp. 485–496.

Gionis, A. et T. Tassa (2009). K-anonymization with minimal loss of information. *IEEE Trans. Knowl. Data Eng. 21*(2), 206–219.

Ienco, D., R. G. Pensa, et R. Meo (2012). From context to distance : Learning dissimilarity for categorical data clustering. *TKDD to appear*.

Iyengar, V. S. (2002). Transforming data to satisfy privacy constraints. In *KDD*, pp. 279–288.

Kohavi, R. et F. Provost (2001). Applications of data mining to electronic commerce. *Data Min. Knowl. Discov. 5*(1/2), 5–10.

Pinto, H., J. Han, J. Pei, K. Wang, Q. Chen, et U. Dayal (2001). Multi-dimensional sequential pattern mining. In *CIKM*, pp. 81–88.

Pitarch, Y., C. Favre, A. Laurent, et P. Poncelet (2010). Context-aware generalization for cube measures. In *DOLAP*, pp. 99–104.

Plantevit, M., A. Laurent, D. Laurent, M. Teisseire, et Y. W. Choong (2010). Mining multidimensional and multilevel sequential patterns. *TKDD 4*(1), 1–37.

Samarati, P. (2001). Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng. 13*(6), 1010–1027.

T. Berners-Lee, J. H. et O. Lassila (2001). The semantic web. *Scientific American*.

Treeratpituk, P. et J. P. Callan (2006). An experimental study on automatically labeling hierarchical clusters using statistical features. In *SIGIR*, pp. 707–708.

Vreeken, J., M. van Leeuwen, et A. Siebes (2011). Krimp : mining itemsets that compress. *Data Min. Knowl. Discov. 23*(1), 169–214.

Yu, L. et H. Liu (2003). Feature selection for high-dimensional data : A fast correlation-based filter solution. In *ICML*, pp. 856–863.

Zhang, J. et V. Honavar (2004). Avt-nbl : An algorithm for learning compact and accurate naïve bayes classifiers from attribute value taxonomies and data. In *ICDM*, pp. 289–296.

Summary

In many domains, a hierarchical organization of attribute values can help the data analysis process. Nevertheless, such hierarchical knowledge does not always available or even may be inadequate or useless when exists. Starting from this consideration, in this paper we tackle the problem of the automatic definition of data-driven taxonomies. To do this we combine techniques coming from information theory and clustering to obtain a structured representation of the attribute values: the Contextual Attribute-Value Taxonomy (CAVT). The two main advantages of our method are to be fully unsupervised and parameter-free. We experiments the benefit of use CAVTs in the two following tasks: (i) multidimensional sequential pattern mining problem in which hierarchies are needed, (ii) table summarization problem, in which the hierarchies are used to aggregate the data. To validate our approach we use real world datasets in which we obtain appreciable results regarding both quantitative and qualitative evaluation.