Vers une plate-forme interactive pour la visualisation de grands ensembles de règles d'association

Olivier Couturier*, Tarek Hamrouni**, Sadok Ben Yahia**, Engelbert Mephu Nguifo*

*CRIL CNRS FRE 2499, IUT de Lens
Rue Jean Souvraz, SP-18
62307 Lens Cedex France
{couturier,mephu}@cril.univ-artois.fr
**Faculté des Sciences de Tunis, Université El-Manar
Campus Universitaire 1060 Tunis, Tunisie
{tarek.hamrouni,sadok.benyahia}@fst.rnu.tn

Résumé. La recherche de règles d'association est une question centrale en Extraction de Connaissances dans les Données (ECD). Dans cet article, nous nous intéressons plus particulièrement à la restitution visuelle de règles pertinentes dans un corpus très important. Nous proposons ainsi un prototype basé sur une approche de type "wrapper" par intégration des phases d'extraction et de visualisation de l'ECD. Tout d'abord, le processus d'extraction génère une base générique de règles et dans un second temps, la tâche de visualisation s'appuie sur un processus de regroupement ("clustering") permettant de grouper et de visualiser un sous-ensemble de règles d'association génériques. Le rendu visuel à l'écran exploite une représentation de type "Fisheye view" de manière à obtenir simultanément une représentation globale des différents groupes de règles et une vue détaillée du groupe sélectionné.

1 Introduction

L'Extraction de Connaissances dans les Données (ECD) a été proposée afin d'aider les utilisateurs à mieux comprendre et appréhender des quantités de données de plus en plus volumineuses. La recherche de règles d'association constitue une question centrale de l'ECD. La plupart des travaux se sont focalisés sur la tâche d'extraction de règles d'association alors que les aspects visualisation de ces règles et interaction avec l'utilisateur-expert sont très peu représentés. De manière générale, le nombre de règles générées croît de manière exponentielle avec la taille des données. En situation réelle, un expert n'a ni le temps, ni les capacités cognitives de traiter ces flots d'information. Pour l'aider à y faire face, différents travaux proposés dans la littérature tournent autour de deux axes complémentaires : la réduction du nombre de règles d'association extraites et le développement d'outils de visualisation interactive. Dans ce papier, nous focalisons notre intérêt sur les méthodes de visualisation.

Un état de l'art des différentes techniques de visualisation de règles d'association est décrit dans Couturier et Mephu-Nguifo (2007). La limitation commune qui en ressort est que lorsque le nombre de règles est élévé, l'interaction avec l'utilisateur devient difficile. Partant

de ce constat, nous proposons ici une nouvelle approche effectuant un regroupement de règles d'association, et particulièrement de règles génériques (Bastide et al. (2000)), en exploitant une représentation en oeil de poisson couplée à une représentation textuelle, 2D ou 3D pour la visualisation de toutes les règles. La représentation en oeil de poisson, dont une première approche est développée dans Couturier et al. (2006), va permettre à l'utilisateur de se focaliser sur un sous-ensemble de règles. Notre approche intègre donc une phase d'extraction de bases génériques et une phase de visualisation en oeil de poisson de ces règles génériques. Ces différentes phases peuvent bien évidement être itérées, et permettent un couplage fort des phases de fouille et de post-traitement. Cette approche est implémenté via le prototype, CBVAR (Clustered-based Visualizer of Association Rules) permettant de gérer simultanément l'extraction et la visualisation de règles dans le but de pouvoir traiter des quantités plus importantes.

Le reste du papier est organisé comme suit : dans la seconde section, nous présentons le prototype que nous avons développé pour la visualisation de clusters de règles. La troisième section est consacrée à la présentation des résultats de nos expérimentations. Enfin, dans la dernière section, nous donnons nos conclusions et perspectives.

2 Le prototype CBVAR (Clustered-based Visualizer of Association Rules)

Dans ce papier, nous attaquons le problème du nombre de règles à visualiser sur deux fronts différents. Tout d'abord, nous travaillons sur des bases génériques de règles d'association, constituant un ensemble générateur de toutes les règles d'association, tout en étant de taille très compacte. Grâce à ce choix, nous réduisons en amont du processus le nombre de règles à traiter. Ensuite, l'organisation de l'ensemble de règles à visualiser se fait grâce à un ensemble de clusters. Ceci permet de diminuer la charge cognitive et de mettre en place un dispositif de visualisation interactif et coopératif. En aval du processus, nous optimisons l'espace écran pour visualiser notre ensemble de règles. En nous basant sur ces deux points, nous proposons le prototype CBVAR implémenté en JAVA et qui fonctionne actuellement sous l'environnement UNIX[1]. Sa principale caractéristique est qu'il intègre un module d'extraction de règles génériques et un module de visualisation qui sont présentés ci-après.

2.1 Extraction de règles génériques

Afin d'obtenir nos clusters, nous utilisons différents outils existants qui sont regroupés au sein d'un script shell. L'extraction des règles d'association nécessite en entrée un fichier texte (dat, txt, etc.). Chaque ligne de ce fichier contient la liste des items composant un objet. Les itemsets fermés fréquents et leurs générateurs minimaux associés, ainsi que les règles génériques sont stockés dans un fichier ($.txt$) grâce à l'algorithme PRINCE (Hamrouni et al. (2005)). Ce dernier génère également un fichier XML ($.xml$) contenant les informations relatives au support minimum (*minsup*) et à la confiance minimum (*minconf*). Ce fichier respecte le standard PMML (Predictive Model Markup Language) dans le but de ne pas limiter la portée de notre prototype sur des applications utilisant des DTD spécifiques.

[1] Disponible à l'adresse suivante http ://www.cril.univ-artois.fr/∼couturier/cbvar/

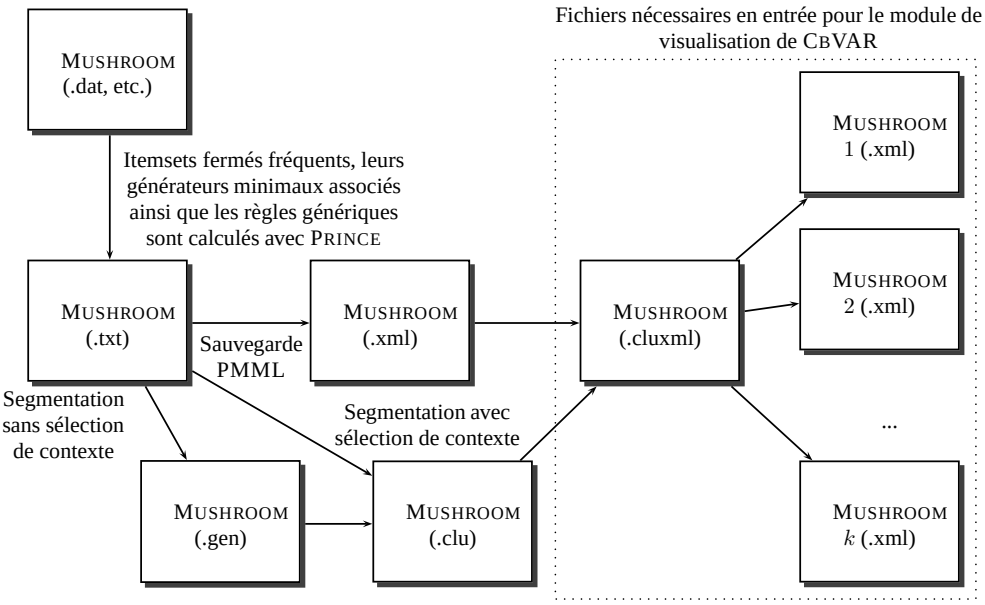

FIG. 1 – *Génération de clusters : Le cas de la base* MUSHROOM.

En outre, un fichier ($.gen$) contenant le méta-contexte associé est généré par PRINCE. Cette étape est très coûteuse en temps de calcul. Pour cette raison, notre prototype peut réutiliser un méta-contexte existant spécifié par l'utilisateur. S'il n'en sélectionne aucun, il sera regénéré. Durant la dernière étape de l'extraction, un fichier ($.clu$) est généré contenant pour chaque règle, un cluster associé. Le nombre de cluster k est spécifié par l'utilisateur. Grâce à ce fichier et au fichier XML correspondant, k fichiers ($.xml$), correspondant aux k clusters sont générés avec en plus, un fichier ($.cluxml$) de méta-cluster listant ces k fichiers XML. Le paquetage contenant ces fichiers constitue le point d'entrée du module de visualisation de CBVAR. Un exemple est présenté dans la figure 1. Chacun des outils utilisés est un paramètre possible pour la génération des clusters. Il est tout à fait possible de les changer en modifiant le script shell associé. Ce script est exécuté par CBVAR en utilisant différents paramètres fixés par l'utilisateur : le support, la confiance, le méta-contexte (si nécessaire) et le nombre de clusters.

2.2 Visualisation de règles d'association : L'approche Fisheye View

Plusieurs représentations connues en Interface Homme Machine (IHM), ont été proposées pour représenter des informations abondantes (Couturier et al. (2006)). Parmi ces représentations, la déformation en œil de poisson (Fisheyes view (FEV)) présentée dans Furnas (1986) nous paraît la plus adaptée et une première approche appliquée à la visualisation de règles d'association a d'ailleurs été proposée dans Couturier et al. (2006). Dans ce travail, la représentation exploite une matrice 2D et une FEV pour visualiser des règles d'association tel que chaque règle constitue un point d'intérêt. Ce principe est réutilisé dans notre approche à la différence que notre point d'intérêt sera un cluster (*cf.* Figure 2).

Nous couplons les approches par matrice 2D et 3D dans notre prototype. La représentation 2D va permettre d'obtenir une vue globale des règles, dans laquelle chaque case va représen-

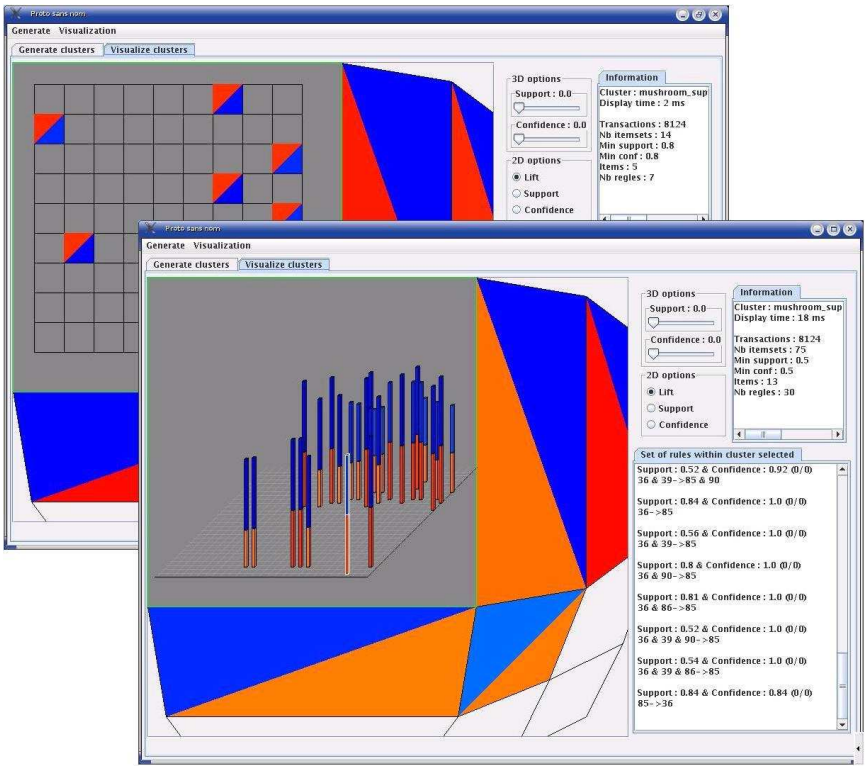

FIG. 2 – *Affichage des clusters de règles d'association avec l'outil* CBVAR.

ter une règle de chaque cluster. Cette règle est sélectionnée en fonction des mesures d'intérêt comme le *lift* (par défaut), le *support* ou la *confiance*. Pour obtenir le détail d'un cluster, l'utilisateur va activer la FEV sur l'une des cases. Pour le représenter au niveau du centre d'intérêt de la FEV, une représentation 3D en projection cabinet, gérant le problème d'occlusions (Couturier et Mephu-Nguifo (2007)), est utilisée pour les contextes épars alors qu'une représentation 2D l'est pour les contextes denses. Un contexte dense est caractérisé par la présence d'au moins une règle exacte, puisque au moins un générateur minimal est différent de sa fermeture associée, ce qui n'est pas le cas pour les contextes épars.

3 Expérimentations

Afin de démontrer la valeur de notre approche, nous avons mis en œuvre des expérimentations sur deux jeux de données[2], à savoir MUSHROOM et T10I4D100K. Le premier est composé de **8 124** transactions pour **119** items, et il s'agit d'un jeu dense. Le second est composé de **100 000** transactions pour **1 000** items, et il s'agit d'un jeu épars. Un ensemble très

[2]http ://fimi.cs.helsinki.fi/data

important de règles est généré à partir de ces jeux de données, et plus particulièrement quand les valeurs de *minsup* et *minconf* sont basses. Les expérimentations ont été réalisées sous Linux avec un PC muni d'un processeur Pentium IV 3 Ghz exploitant 1 GB de RAM . Les métriques *minsup* et *minconf* sont variables. Par conséquence, le nombre de règles génériques extraites (noté *# règles génériques*) augmentent tant que les valeurs des métriques diminuent : de $\mathbf{537}$ (avec *minsup* = $\mathbf{0{,}4}$ et *minconf* = $\mathbf{0{,}4}$) jusqu'à $\mathbf{7\ 057}$ (avec *minsup* = $\mathbf{0{,}2}$ et *minconf* = $\mathbf{0{,}2}$) pour les données MUSHROOM, et de $\mathbf{594}$ (avec *minsup* = $\mathbf{0{,}006}$ et *minconf* = $\mathbf{0{,}006}$) jusqu'à $\mathbf{3\ 623}$ (avec *minsup* = $\mathbf{0{,}004}$ et *minconf* = $\mathbf{0{,}004}$) pour les données T10I4D100K.

Le nombre de clusters est défini en fonction du nombre x de règles d'association que peut contenir un cluster. Nos résultats sont présentés dans la table 1. Tous les fichiers XML sont initialement illisibles dans un même espace écran à cause du nombre important de règles. Notre approche permet de visualiser ces jeux de données dans un même espace écran, avec approximativement les même temps d'affichage pour un nombre de clusters égal à ($\frac{\#\ regles\ generiques}{100}$) pour des données denses ou éparses. Nos tests exploitant différentes valeurs de x (*i.e.*, x = $\mathbf{25}$ and x = $\mathbf{50}$) requièrent en général plus de temps à l'affichage. Cependant, pour les clusters contenant moins de $\mathbf{100}$ règles, il est possible de réduire le temps relatif à l'affichage d'un cluster puisqu'il est le seul à être affiché en détail tout en contenant moins de règles.

Jeu de données	*minsup*	*minconf*	# règles	# règles génériques	Temps d'affichage (ms)			
					Sans clusters	Avec ($\frac{\#\ regles\ generiques}{x}$) clusters		
						x = $\mathbf{100}$	x = $\mathbf{50}$	x = $\mathbf{25}$
MUSHROOM	$\mathbf{0{,}4}$	$\mathbf{0{,}4}$	**7 020**	**537**	828	**375**	375	594
MUSHROOM	$\mathbf{0{,}3}$	$\mathbf{0{,}3}$	**94 894**	**2 184**	2 250	**1 782**	3 250	5 750
MUSHROOM	$\mathbf{0{,}2}$	$\mathbf{0{,}2}$	**19 197 504**	**7 057**	-	**17 953**	-	-
T10I4D100K	$\mathbf{0{,}006}$	$\mathbf{0{,}006}$	**928**	**594**	1 453	**516**	813	1 594
T10I4D100K	$\mathbf{0{,}005}$	$\mathbf{0{,}005}$	**2 216**	**1 231**	2 475	**1 375**	2 329	4 125
T10I4D100K	$\mathbf{0{,}004}$	$\mathbf{0{,}004}$	**8 090**	**3 623**	-	**6 078**	11 750	27 000

TAB. 1 – *Évolution des temps d'affichage par rapport à la variation du nombre de clusters.*

Afin d'interagir en temps réel, il est nécessaire de charger toutes les règles en mémoire. Cependant, nous pouvons observer qu'avec des valeurs de *minsup* (resp. *minconf*) égales à $\mathbf{0{,}2}$ (resp. $\mathbf{0{,}2}$) pour les données MUSHROOM et avec des valeurs de *minsup* (resp. *minconf*) égales à $\mathbf{0{,}004}$ (resp. $\mathbf{0{,}004}$) pour les données T10I4D100K, il n'est pas toujours possible de le faire avec les visualisations classiques sans clustering. Notre approche permet de visualiser le premier jeu de données en $\mathbf{17\ 953}$ ms avec $\frac{\#\ regles\ generiques}{100}$ clusters. Cet ensemble représente $\mathbf{7\ 057}$ règles qui sont incluses dans le même espace écran (*cf.* Tableau 1). Le second jeu de données est affiché avec notre approche pour les différentes valeurs de x utilisées pour nos tests. Dans ce cas, l'ensemble de règles d'association contient jusqu'à $\mathbf{3\ 623}$ règles (*cf.* Tableau 1). Dans ce contexte, nous n'avons pas utilisé de valeur de *minsup* élevée puisque le nombre de règles n'aurait pas été assez important pour justifier l'intérêt de notre approche. Une approche classique aurait elle aussi permis de les traiter.

4 Conclusion et perspectives

Dans ce papier, nous avons présenté une méthode de visualisation de grands ensembles de règles d'association. Ainsi, nous avons proposé d'utiliser le clustering sur un ensemble de

règles génériques pour réduire la charge cognitive de l'utilisateur. Nous avons implémenté le prototype CBVAR correspondant. Les expérimentations que nous avons menées ont mis en exergue la possibilité de visualiser rapidement dans un même espace écran quelques milliers de règles en quelques secondes et que notre prototype est performant sur des quantités importantes de données. Ainsi, grâce à l'hybridation d'une représentation 2D et d'une FEV, notre approche permet d'obtenir simultanément une vue globale et détaillée de nos règles.

Pour la suite, nous souhaitons principalement focaliser nos efforts sur l'étape de clustering qui est centrale dans notre approche. Ainsi, la piste des k-hiérarchies faibles sera considérée. En outre, comme la famille des regroupements produite par une hiérarchie faible constitue une famille de Moore, alors la structure du treillis pourrait devenir une structure de visualisation privilégiée. Du point de vue IHM, nous souhaitons réaliser une évaluation utilisateur.

Remerciements

Ce travail est soutenu par le projet Franco-Tunisien CMCU 05G1412.

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 DOOD'00*, pp. 972–986, London, United Kingdom.

Couturier, O. et E. Mephu-Nguifo (2007). Visualisation de règles d'association en 3D par réduction des occlusions. *Revue I3 (A paraître)*.

Couturier, O., J. Rouillard, et V. Chevrin (2006). Une approche hybride pour une meilleure visualisation de grands ensembles de règles d'association. Dans *Actes d'ERGOIA'06*, Biarritz, France.

Furnas, G. (1986). Generalized fisheye views. In *Proceedings of CHI'86*, pp. 16–23, Boston, USA.

Hamrouni, T., S. Ben Yahia, et Y. Slimani (2005). PRINCE : An algorithm for generating rule bases without closure computations. In A. M. Tjoa et J. Trujillo (Eds.), *Proceedings of DaWaK 2005*, pp. 346–355, Copenhagen, Denmark.

Summary

One of the most important points in Data Mining is to propose efficient and easy-to-use graphical tools to users. These tools must be able to generate explicit knowledge and to restitute it. Even though considered as a key step in the Knowledge Data Discovery (KDD) process, the association rule visualization received much less attention than that paid to the extraction step, however. Standing at the crossroads of Data mining (DM) and Human-Machine Interaction (HCI), we present an integrated framework covering the last two steps of the KDD process, *i.e.*, patterns' extraction and visualization. At first, the extraction process yields compact and informative set of association rules better known as "generic bases". Acting as "irreducible" nuclei of association rules, they permit to drastically reduce the number of handled association rules. Second, the visualization process is based on a clustering of these generic rules, making it possible to scalably handle large quantities of association rules.