Une approche logique pour la fouille de règles d'association

Abdelhamid Boudane*, Said Jabbour*
Lakhdar Sais*, Yakoub Salhi*

*CRIL-CNRS, Université d' Artois F-62307 Lens Cedex France
{boudane,jabbour,sais,salhi}@cril.fr

Résumé. La découverte de règles d'association à partir de données transactionnelles est une tâche largement étudiée en fouille de données. Les algorithmes proposés dans ce cadre partagent la même méthodologie en deux étapes à savoir l'énumération des itemsets fréquents suivie par l'étape de génération de règles. Dans cet article, nous proposons une nouvelle approche basée sur la satisfiabilité propositionnelle pour extraire les règles d'association en une seule étape. Pour montrer la flexibilité et la déclarativité de notre approche, nous considérons également deux autres variantes, à savoir la fouille de règles d'association fermées et la fouille de règles indirectes. Les expérimentation sur plusieurs jeux de données montrent que notre approche offre de meilleures performances comparée à des approches spécialisées.

1 Introduction

L'extraction des règles d'association est l'une des tâches fondamentales de la fouille de données. Elle vise à découvrir des relations intéressantes cachées dans de grandes bases de données. Ces relations entre les items (ensembles d'attributs) sont présentées sous forme d'implications, appelées règles d'association. Depuis la première application, largement connue sous la dénomination de panier de la ménagère (Agrawal et Srikant, 1994), plusieurs nouveaux domaines d'application ont été identifiés comme la bioinformatique, le diagnostic médical, la détection d'intrusion, la fouille du web et l'analyse des données scientifiques.

L'extraction de règles d'association a connu de nombreux développements théoriques et algorithmiques. Parmi ces algorithmes, Apriori (Agrawal et Srikant, 1994) et FP-Growth (Han et al., 2004) sont les plus connus. Tous ces algorithmes partagent une même méthodologie à deux étapes. La première est consacrée à la recherche de tous les itemsets fréquents, et la seconde consiste à générer les règles avec une grande confiance en combinant ces itemsets fréquents. On peut citer aussi l'approche logique GUHA proposée par Hájek et al. (Hájek et al., 2010) il y a plus d'une trentaine d'année.

Comme souligné dans (Raedt et al., 2011), les contraintes font souvent partie de la spécification de plusieurs problèmes de fouille de données. Cette observation a conduit à un nouveau domaine de recherche actif et multidisciplinaire initié dans (Raedt et al., 2008), et permettant une fertilisation croisée entre la fouille de données et l'intelligence artificielle (IA). Deux modèles de représentation et de résolution d'IA ont été utilisés pour modéliser et résoudre plusieur problèmes de fouille de données, à savoir la programmation par contrainte (PPC) et la

satisfiabilité propositionnelle (SAT). Parmi ces problèmes, nous pouvons citer l'extraction de motifs (Raedt et al., 2011; Jabbour et al., 2015a) et le clustering (Davidson et al., 2010). Suivant cette tendance de recherche, nous proposons dans cet article une nouvelle approche basée sur la satisfiabilité propositionnelle pour fouiller les règles d'association en une seule étape. Dans notre deuxième contribution, nous considérons deux variantes bien connues, à savoir la fouille de règles d'association fermées (Taouil et al., 2000), et la fouille de règles d'association indirectes (Tan et al., 2000). Notre objectif est de montrer la flexibilité et la déclarativités de notre approche.

2 Préliminaires

2.1 Logique propositionnelle et problème SAT

Soit $Prop$ un ensemble dénombrable de variables propositionnelles. Nous utilisons les lettres p, q, r, etc. pour noter les éléments de Prop. L'ensemble de formules propositionnelles est défini par induction à partir de Prop, les deux constantes $\bot$ (resp. $\top$) qui désignent les états $faux$ (resp. $vrai$) et en utilisant les connecteurs logiques usuels $\neg$, $\wedge$, $\vee$, $\rightarrow$, et $\leftrightarrow$. On utilise $\mathcal{P}(A)$ pour désigner l'ensemble des variables propositionnelles apparaissant dans la formule A. Une interprétation booléenne $\mathcal{I}$ d'une formule A est définie comme étant une fonction de $\mathcal{P}(A)$ vers $\{0,1\}$ (0 correspond à $faux$ et 1 à $vrai$). Un *modèle* d'une formule A est une interprétation $\mathcal{I}$ qui satisfait A, c-à-d $\mathcal{I}(A) = 1$. Une formule A est satisfiable s'il existe un modèle de A. Une formule sous forme normale conjonctive (CNF) est une conjonction ($\wedge$) de clauses, où une *clause* est une disjonction ($\vee$) de littéraux. Un *littéral* est une variable propositionnelle (p) ou sa négation ($\neg p$). Le problème *SAT* consiste à décider si une formule CNF donnée admet un modèle ou non.

2.2 Règles d'association

Soit Ω un ensemble non vide fini de symboles, appelés *items*. Nous utilisons les lettres a, b, c, etc. pour noter les éléments de Ω. Un *itemset* I sur Ω est défini comme étant un sous ensemble de Ω, c-à-d, $I \subseteq \Omega$. Nous utilisons 2^{Ω} pour désigner l'ensemble des itemsets sur Ω et nous utilisons les lettres majuscules I, J, K, etc. pour noter les éléments de 2^{Ω}.

Une *transaction* est une paire ordonnée (i, I) où i est un nombre naturel appelé *identifiant* et I est un itemset, c-à-d $(i, I) \in \mathbb{N} \times 2^{\Omega}$.

Étant donné une base de données transactionnelles $\mathcal{D}$ et un itemset I, la couverture de I dans $\mathcal{D}$, notée $\mathcal{C}(I, \mathcal{D})$, est définie comme suit : $\{i \in \mathbb{N} \mid (i, J) \in \mathcal{D}\ et\ I \subseteq J\}$. Le support de I dans $\mathcal{D}$, notée $\mathcal{S}(I, \mathcal{D})$, correspond à la cardinalité de $\mathcal{C}(I, \mathcal{D})$, c-à-d $\mathcal{S}(I, \mathcal{D}) = |\mathcal{C}(I, \mathcal{D})|$.

Un itemset $I \subseteq \Omega$ tel que $\mathcal{S}(I, \mathcal{D}) \geq 1$ est un *itemset fermé* si, pour tous les itemsets J avec $I \subset J$, $\mathcal{S}(J, \mathcal{D}) < \mathcal{S}(I, \mathcal{D})$.

Une *règle d'association* est un motif de la forme $X \rightarrow Y$ où X (appelé antécédent) et Y (appelé conséquence) sont deux itemsets disjoints. Le *support d'une règle d'association* $X \rightarrow Y$ dans $\mathcal{D}$ est : $\mathcal{S}(X \rightarrow Y, \mathcal{D}) = \frac{\mathcal{S}(X \cup Y, \mathcal{D})}{|\mathcal{D}|}$. La *confiance* de $X \rightarrow Y$ dans $\mathcal{D}$ est définie comme étant $\mathcal{C}onf(X \rightarrow Y, \mathcal{D}) = \frac{\mathcal{S}(X \cup Y, \mathcal{D})}{\mathcal{S}(X, \mathcal{D})}$. Elle donne une estimation de la probabilité conditionnelle de Y sachant X. Étant donné une base de données transactionnelle $\mathcal{D}$, un minimum sup-

port α et un minimum de confiance β, le problème de fouille de règles d'association consiste plus précisément à calculer l'ensemble suivant : $\mathcal{MAR}(\mathcal{D}, \alpha, \beta) = \{X \rightarrow Y \mid X, Y \subseteq \Omega \wedge \mathcal{S}(X \rightarrow Y, \mathcal{D}) \geq \alpha \wedge \mathcal{C}onf(X \rightarrow Y, \mathcal{D}) \geq \beta\}$

3 SAT et fouille de règles d'association

Dans cette section, nous décrivons un encodage SAT pour le problème de fouille de règles d'association. Pour représenter les deux itemsets de chaque règle candidate $X \rightarrow Y$, on associe deux variables propositionnelles x_a et y_a à chaque item a. Ensuite, pour représenter la couverture de X et $X \cup Y$, on associe à chaque identifiant d'une transaction $i \in \{1 \ldots m\}$ deux variables propositionnelles p_i et q_i. Les deux clauses de la première formule exprime que X

$$(\bigvee_{a \in \Omega} x_a) \wedge (\bigvee_{a \in \Omega} y_a) \quad (1) \qquad \bigwedge_{i \in 1..m} (\neg q_i \leftrightarrow \neg p_i \vee (\bigvee_{a \in \Omega \setminus I_i} y_a)) \quad (4)$$

$$\bigwedge_{a \in \Omega} (\neg x_a \vee \neg y_a) \quad (2) \qquad \sum_{i \in 1..m} q_i \geq m \times \alpha \quad (5)$$

$$\bigwedge_{i \in 1..m} (\neg p_i \leftrightarrow \bigvee_{a \in \Omega \setminus I_i} x_a) \quad (3) \qquad \frac{\sum_{i \in 1..m} q_i}{\sum_{i \in 1..m} p_i} \geq \beta \quad (6)$$

et Y ne sont pas vides. La deuxième formule propositionnelle permet d'exprimer la contrainte $X \cap Y = \emptyset$, en imposant que x_a et y_a ne peuvent pas être vrais en même temps. Pour obtenir la couverture de l'itemset X (resp. $X \cup Y$), on utilise la formule propositionnelle 3 (resp. 4). Dans cette formule, p_i (resp. q_i) est $faux$ ssi X (resp. $X \cup Y$) contient un item qui n'appartient pas à la transaction i. les deux formules 5 et 6 expriment que le support et la confiance de la règle candidate doivent être supérieurs ou égaux au seuils spécifiés α et β.

$\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ désigne la conjonction de formules (1), (2), (3), (4), (5) et (6).

3.1 Règles d'association fermées et indirectes

Dans cette section, notre objectif est de montrer la flexibilité de notre approche declarative, en considérant d'autres variantes de règles d'association.

Définition 1 (Règle d'association fermée) *Une règle d'association $r : X \rightarrow Y$ est une règle fermée ssi $r : X \rightarrow Y$ est une règle d'association valide et $X \cup Y$ est un itemset fermé.*

Intuitivement, nous obtenons les règles d'association fermées en maximisant l'union de l'antécédent et la conséquence, sans diminuer ni le support ni la confiance. L'encodage SAT du problème de fouille de règles d'association fermées, noté $\mathcal{E}_{CAR}(\mathcal{D}, \alpha, \beta)$, peut être obtenu simplement en ajoutant à l'encodage décrit précédemment ($\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$) la formule suivante :

$$\bigwedge_{a \in \Omega} (\bigwedge_{i=1}^{m} (q_i \rightarrow a \in I_i) \wedge \neg x_a \rightarrow y_a) \quad (7)$$

La formule (7) signifie que si on a $\mathcal{C}(X \cup Y, \mathcal{D}) = \mathcal{C}(X \cup Y \cup \{a\}, \mathcal{D})$ alors $a \in Y$.

Définition 2 *Soit $\mathcal{D}$ une base de données transactionnelles. Deux items a_0 et b_0 sont indirectement liés par l'intermédiaire d'un itemset M, appelé médiateur, en respectant un maximum de support λ, un minimum de support α et un seuil de dépendance médiateur β ssi les conditions suivantes sont satisfaites :*

- $\mathcal{S}(\{a_0\} \rightarrow \{b_0\}, \mathcal{D}) \leq \lambda$ *(condition de support sur la pair d'items)* [1].
- *Il existe un itemset non vide M tel que :*
  1. $\mathcal{S}(\{a_0\} \rightarrow M, \mathcal{D}) \geq \alpha$ *et* $\mathcal{S}(\{b_0\} \rightarrow M, \mathcal{D}) \geq \alpha$ *(Condition de Support sur M)*
  2. $\mathcal{D}ep(\{a_0\}, M, \mathcal{D}) \geq \beta$ *et* $\mathcal{D}ep(\{b_0\}, M, \mathcal{D}) \geq \beta$ *où* $\mathcal{D}ep(P, Q, \mathcal{D})$ *est une mesure de dépendance entre P et Q.*

En d'autres termes, dans la fouille des règles indirectes, nous cherchons les paires d'items qui sont infréquentes (rares) mais qui sont impliquées séparément dans des règles d'association intéressantes avec la même conséquence.
En utilisant la confiance comme mesure de dépendance et un seuil minimum de confiance au lieu du seuil de dépendance médiateur, les deux conditions (support du médiateur et dépendance) dans la définition 2 peuvent être réécrites comme suit : $\{a_0\} \rightarrow M$ et $\{b_0\} \rightarrow M$ sont deux règles d'association valides en respectant les seuils α et β.

On utilise pour chaque item les variables propositionnelles $x_c^{a_0}$ et $x_c^{b_0}$ pour représenter a_0 et b_0 respectivement. On utilise le même ensemble de variables y_a pour chaque item a comme dans $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ afin de capturer les éléments du médiateur. De plus, on introduit des variables de la forme $p_i^{a_0}$ (resp. $p_i^{b_0}$) pour exprimer la couverture de l'item a_0 (resp. b_0) de la même façon que dans $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ et des variables de la forme $q_i^{a_0}$ et $q_i^{b_0}$ pour exprimer la couverture de $M \cup \{a_0\}$ et $M \cup \{b_0\}$ respectivement. Enfin, on introduit des variables de la forme r_i pour capturer la couverture de $\{a_0, b_0\}$.

Les deux règles $\{a_0\} \rightarrow M$ et $\{b_0\} \rightarrow M$ sont capturées comme suit :

$$\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta) \wedge (\sum_{a \in \Omega} x_a^{a_0} = 1) \ , \ \mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta) \wedge (\sum_{a \in \Omega} x_a^{b_0} = 1) \tag{8}$$

Nous décrivons maintenant la formule qui permet de capturer la couverture de $\{a_0, b_0\}$ et celle exprimant que $\{a_0\} \rightarrow \{b_0\}$ est infréquente en respectant le seuil maximum de support λ :

$$r_i \leftrightarrow (p_i^{a_0} \wedge p_i^{b_0}) \ , \ \sum_{i=1}^{m} r_i \leq m \times \lambda \tag{9}$$

Dans la définition 2, les items a_0 et b_0 sont interchangeables menant à des règles d'association indirectes symétriques. Pour éviter l'énumération de ces règles redondantes, nous cassons les symétries entre a_0 et b_0 en ajoutant les contraintes $a_0 < b_0$ sur l'ensemble des items Ω exprimées comme suit :

$$\bigwedge_{a, a' \in \Omega, a' \leq a} \neg x_a^{a_0} \vee \neg x_{a'}^{b_0} \tag{10}$$

On utilise $\mathcal{E}_{IR}(\mathcal{D}, \lambda, \alpha, \beta)$ pour désigner l'encodage du problème de fouille de règles d'association indirectes $(8) \wedge (9) \wedge (10)$.

Finalement, les encodages des différents types de règles d'associations sont corrects et complets. La preuve découle naturellement des différentes contraintes associées.

1. on peut aussi écrire $\frac{\mathcal{S}(\{a_0, b_0\}, \mathcal{D})}{|\mathcal{D}|} \leq \lambda$

3.2 Résultats

Dans cette section nous allons présenter quelques résultats obtenus en comparant l'approche basée sur SAT à des approches spécialisées. Nous considérons les trois tâches de fouille de règles d'association, appelées : pures, fermées, et indirectes. Nous indiquons par SFAR_R avec $R \in \{Pure, Ferm, Ind\}$, notre solveur basé sur l'approche SAT pour fouiller les règles d'association (R) correspondantes (pour plus de détails, voir (Jabbour et al., 2015b)). Les contraintes (5) et (6) sont gérées dynamiquement dans le solveur en associant à chacune un propagateur comme dans (Jabbour et al., 2015a). Pour les règles *pures* et *fermées*, nous comparons notre approche à l'algorithme *ZART* implémenté dans *Coron* [2]. Par contre, pour les règles *indirectes*, nous comparons notre solveur à l'algorithme *INDIRECT* implémenté dans *SPMF* [3] (Fournier-Viger et al., 2014). Nous avons testé sur 16 bases de données transactionnelles [4]. Pour chaque base, nous avons généré 400 (resp. 250) configurations pour la fouille des règles Pures et Fermées (resp. indirectes), en variant les seuils de support et de confiance.

	SFAR_Pure		ZART_Pure		SFAR_Ferm		ZART_Ferm		SFAR_Ind		SPMF_Ind	
données	#S	moy. t(s)	#S	moy. t(s)	#S	moy. t(s)	#S	moy. t(s)	#S	moy. t(s)	#S	moy. t(s)
Audiology	20	855.00	20	855.01	20	855.00	20	855.01	124	453.74	61	680.45
Zoo-1	400	19.12	400	6.37	400	0.52	400	11.28	250	0.15	250	9.12
Tic-tac-toe	400	0.09	400	0.24	400	0.09	400	0.23	250	0.09	250	0.20
Anneal	101	709.50	101	678.41	147	604.09	103	679.31	171	309.69	55	702.04
Australian-credit	245	370.17	264	321.62	268	323.29	226	403.72	232	121.06	156	339.56
German-credit	306	246.88	322	192.52	329	198.02	304	238.79	244	49.07	210	154.49
Heart-cleveland	284	286.38	301	252.27	304	251.05	262	340.15	235	64.97	203	300.48
Hepatitis	305	241.41	304	228.00	324	206.02	266	312.26	245	32.98	205	187.92
Hypothyroid	85	732.12	121	665.41	107	686.95	64	761.59	163	336.40	81	621.29
Kr-vs-kp	172	552.92	203	487.73	192	523.66	146	590.89	204	206.47	114	499.33
Lymph	336	181.64	338	170.37	387	63.22	291	281.35	250	6.10	211	170.19
Mushroom	366	109.12	387	46.00	400	30.32	390	42.84	250	8.89	250	29.62
Primary-tumor	400	3.68	400	1.17	400	2.03	400	18.82	250	0.15	250	2.63
Soybean	400	2.90	400	1.50	400	0.17	400	7.94	250	0.05	250	0.76
Splice-1	380	53.44	400	3.52	380	54.04	400	3.25	250	61.73	250	0.50
Vote	380	66.74	400	1.46	400	32.40	398	30.22	250	0.84	250	1.48
Total	4560	279.76	**4741**	**247.29**	**4838**	**242.24**	4470	286.10	**3618**	**103.27**	3046	231.25

TAB. 1 – *Règles Pures, Fermées, et Indirectes :* SFAR *vs* ZART *et* SFAR *vs* SPMF

D'après les résultats présentés dans la Table 1 (où #S représente le nombre de configurations résolues sous une limite de temps égale à 900s), nous constatons que l'approche basée sur SAT donne des performances meilleures que les approches spécialisées pour la fouille des règles d'association fermées et indirectes. Par contre, pour les règles d'association pures, les approches spécialisées sont meilleures.

4 Conclusion et perspectives

Dans ce travail, nous avons développé une nouvelle approche efficace et déclarative pour fouiller les règles d'association. Cette méthode, basée sur un encodage SAT, permet d'extraire les règles d'association en une seule étape. Comme deuxième contribution nous avons montré

2. Coron : http ://coron.loria.fr/site/system.php
3. SPMF : http ://www.philippe-fournier-viger.com/spmf/
4. https ://dtai.cs.kuleuven.be/CP4IM/datasets/

que cette approche est flexible en modélisant facilement d'autres types de règles d'association tels que les règles fermées et les règles indirectes. Les expérimentations ont montré que cette approche est plus efficace pour l'extraction des règles fermées et indirectes. Comme perspectives, nous envisageons d'étendre cette approche en modélisant d'autres types de règles d'association tels que les règles minimales non redondantes et les règles exceptionnelles.

Références

Agrawal, R. et R. Srikant (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of VLDB'94*, pp. 487–499.

Davidson, I., S. S. Ravi, et L. Shamis (2010). A SAT-based framework for efficient constrained clustering. In *Proceedings of SDM'10*, pp. 94–105.

Fournier-Viger, P., A. Gomaric, T. Gueniche, A. Soltani, C.-W. Wu, et V. S. Tseng (2014). Spmf : a java open-source pattern mining library. *The Journal of Machine Learning Research 15*(1), 3389–3393.

Hájek, P., M. Holena, et J. Rauch (2010). The guha method and its meaning for data mining. *Journal of Computer and System Sciences 76*(1), 34 – 48.

Han, J., J. Pei, Y. Yin, et R. Mao (2004). Mining frequent patterns without candidate generation : A frequent-pattern tree approach. *Data Mining and Knowledge Discovery 8*(1), 53–87.

Jabbour, S., L. Sais, et Y. Salhi (2015a). Decomposition based SAT encodings for itemset mining problems. In *Proceedings of PAKDD'15*, pp. 662–674.

Jabbour, S., L. Sais, et Y. Salhi (2015b). On SAT models enumeration in itemset mining. *CoRR abs/1506.02561*.

Raedt, L. D., T. Guns, et S. Nijssen (2008). Constraint programming for itemset mining. In *Proceedings of SIGKDD'08*, pp. 204–212.

Raedt, L. D., S. Nijssen, B. O'Sullivan, et P. V. Hentenryck (2011). Constraint programming meets machine learning and data mining. *Dagstuhl Reports 1*(5), 61–83.

Tan, P.-N., V. Kumar, et J. Srivastava (2000). Indirect association : Mining higher order dependencies in data. In *Proceedings of PKDD'00*, pp. 632–637.

Taouil, R., N. Pasquier, Y. Bastide, et L. Lakhal (2000). Mining bases for association rules using closed sets. In *Proceedings of ICDE'00*, pp. 307.

Summary

All the algorithms that mine association rules, share the same two steps methodology: frequent itemsets enumeration followed by effective association rules generation step. In this paper, we propose a new propositional satisfiability based approach to mine association rules in a single step. To highlight the flexibility of our proposed framework, we also address two other variants, namely the closed and indirect association rules mining tasks. Experiments on many datasets show that on both closed and indirect association rules mining tasks, our declarative approach achieves better performance than the state-of-the-art specialized techniques.