L'extraction de règles de dépendance bien définies entre ensembles de variables multivaluées

Frédéric Pennerath[*,**]

[*]UMI 2958 GeorgiaTech - CNRS
[**]Supélec
2 rue Édouard Belin, 57070 Metz

Résumé. Cet article étudie la faisabilité et l'intérêt de l'extraction de règles de dépendance entre ensembles de variables multivaluées en comparaison du problème bien connu de l'extraction des règles d'association fréquentes. Une règle de dépendance correspond à une dépendance fonctionnelle approximative caractérisée principalement par l'entropie conditionnelle associée. L'article montre comment établir une analogie formelle entre les deux familles de règles et comment adapter à l'aide de cette analogie l'algorithme « Eclat » afin d'extraire d'un jeu de données les règles de dépendance dites bien définies. Une étude expérimentale conclut sur les forces et inconvénients des règles de dépendance bien définies vis-à-vis des règles d'association fréquentes.

1 Introduction

Dans de nombreux problèmes de fouille de motifs, les données à analyser forment une collection d'échantillons décrits par différents attributs multivalués appelés *variables* dans ce qui suit. Un tel jeu de données est illustré par l'exemple simplifié de la figure 1(a). Les motifs

	V_1	V_2	V_3	V_4
D_1	a	1	oui	1
D_2	c	2	oui	1
D_3	b	1	non	1
D_4	c	2	non	1

(a) Avec variables multivaluées

	V_1 = a	V_1 = b	V_1 = c	V_2 = 1	V_2 = 2	V_3 = yes	V_3 = no	V_4 = 1
D_1	×			×		×		×
D_2			×		×	×		×
D_3		×		×			×	×
D_4			×		×		×	×

(b) Avec items ou attributs monovalués

FIG. 1 – *Interprétation mono ou multivaluée d'un même jeu de données.*

les plus souvent recherchés dans ces données sont des *itemsets* : un itemset est un ensemble d'attributs monovalués appelés *items* représentant chacun une des valeurs possibles d'une des variables multivaluées. Un jeu de données défini sur un ensemble d'attributs multivalués peut être représenté de façon équivalente par des items comme illustré sur la figure 1(b). Un itemset *couvre* une donnée si tous les items de l'itemset sont aussi des items de la donnée. La *fréquence* d'un itemset est la proportion des données couvertes par celui-ci. Une question consiste alors

à se demander quelles sont les paires d'itemsets disjoints H et C telles que C apparaît suffisamment fréquemment dans les données couvertes par H ? Cette question a été formalisée et résolue par le problème de l'*extraction des règles d'association fréquentes* [1]. Indépendamment de cette direction de recherche, des méthodes ont été proposées pour fouiller directement des données multivaluées à l'aide des *dépendances fonctionnelles approximatives* (voir par exemple [3, 6]) ou d'extensions de l'ACF (voir par exemple [5]). Une dépendance fonctionnelle (DF) $X \rightarrow Y$ entre deux ensembles disjoints de variables X et Y traduit le fait que pour le jeu de données considéré, les valeurs des variables de Y peuvent être déduites des valeurs des variables de X. Une DF est approximative (DFA) si les valeurs de Y peuvent être déterminées à partir de celles de X avec un risque d'erreur faible en de ça d'un certain seuil (exprimé en nombre d'exceptions à la règle, nombre de valeurs indéterminées, etc). L'article propose un rapprochement entre les DFA et les règles d'association fréquentes, comme cela a déjà été fait notamment dans [6]. L'originalité de ce rapprochement est de reposer sur une analogie formelle simple et élégante mettant en bijection les concepts des motifs fréquents avec ceux associés aux ensembles de variables multivalués caractérisés par des mesures entropiques. Les DFA sont ainsi caractérisées principalement par l'entropie conditionnelle et sont alors appelées dans ce qui suit des *règles de dépendance* entre ensembles de variables multivaluées pour souligner l'analogie formelle au cœur de cet article, mettant en correspondance règles de dépendance et règles d'association. L'article introduit ainsi un cadre théorique similaire à celui des règles d'association pour fouiller les ensembles de variables multivaluées à l'aide des règles de dépendance dites bien définies puis propose H-Eclat, une adaptation de l'algorithme Eclat [7] pour extraire rapidement ces règles. La section 2 commence ainsi par souligner l'analogie formelle entre les deux formes de règles. La section 3 décrit les principes de H-Eclat puis présente les résultats expérimentaux obtenus avant de conclure à la section 4.

2 Les ensembles de variables bien définis

Étant donné un ensemble fini $\mathbb{V} = \{V_1, \ldots, V_m\}$ de m variables multivaluées à domaine fini, un *jeu de données* défini sur $\mathbb{V}$ est un ensemble de n données $\mathcal{D} = \{D_i\}_{1 \leq i \leq n}$, où chaque donnée D_i est un vecteur à m composantes associant une valeur $D_i(V)$ à chaque variable V de $\mathbb{V}$ comme illustré sur la figure 1(a). Les éléments de $\mathbb{V}$ sont appelés des *variables atomiques*. Un *ensemble de variables* est défini comme un sous-ensemble de variables atomiques. Les valeurs d'un ensemble de variables $\mathcal{V} \subseteq \mathbb{V}$ sont les combinaisons de valeurs des variables de $\mathcal{V}$ dans les données. Formellement, si on suppose les variables atomiques ordonnées selon un ordre d'indexation arbitraire (i.e. V_i précède V_j si $i < j$) et si le *domaine* $\mathbb{D}_V$ d'une variable V désigne l'ensemble des valeurs que peut prendre V, alors l'ensemble, ou *domaine* $\mathbb{D}_\mathcal{V}$, des valeurs de l'ensemble de variables $\mathcal{V} = \{V_{i_1} \ldots V_{i_k}\}$ tel que $i_1 < \cdots < i_k$ est défini comme le produit cartésien $\mathbb{D}_{V_{i_1}} \times \cdots \times \mathbb{D}_{V_{i_k}}$. Une valeur de $\mathcal{V}$ est donc un tuple élément de $\mathbb{D}_\mathcal{V}$. Dans l'exemple de la figure 1(a), (b, oui) est une des six valeurs de $\mathbb{D}_{\{V_1, V_3\}} = \{a, b, c\} \times \{oui, non\}$. Un tuple $t = (v_1, \ldots, v_k)$ valeur de $\mathcal{V} = \{V_{i_1} \ldots V_{i_k}\}$ *couvre* une donnée D si et seulement si $D(V_{i_j}) = v_j$ pour tout entier j variant de 1 à k. La *couverture* $C_\mathcal{D}(t)$ d'un tuple t dans les données $\mathcal{D}$ est l'ensemble des données couvertes par V. Dans l'exemple le tuple $(c, 2, 1)$ de $\{V_1, V_2, V_4\}$ a pour couverture $\{D_2, D_4\}$. Les valeurs d'un ensemble de variables $\mathcal{V}$ définissent une partition des données : la *partition de données* $\mathcal{P}_\mathcal{D}(\mathcal{V})$ de l'ensemble de variables $\mathcal{V}$ est définie comme l'union des couvertures

non vides $C_{\mathcal{D}}(t)$ pour les différentes valeurs t de $\mathbb{D}_{\mathcal{V}}$. La partition de l'ensemble de variables $\mathcal{V} = \{V_1, V_2\}$ dans les données de la figure 1(a) est $\{C_{\mathcal{V}}((a,1)), C_{\mathcal{V}}((c,2)), C_{\mathcal{V}}((b,1))\}$, soit $\{\{D_1\}, \{D_2, D_4\}, \{D_3\}\}$. Un ensemble de variables $\mathcal{V}$ contiendra donc d'autant plus d'information qu'il permettra de partitionner les données, c'est-à-dire de distinguer les données à partir des valeurs que peut prendre $\mathcal{V}$. Ainsi la variable V_4 contient peu d'information puisque sa partition $\mathcal{P}_{\mathcal{D}}(\mathcal{V}) = \{\{D_1, D_2, D_3, D_4\}\}$ ne permet pas de distinguer les données. Cette quantité d'information peut se mesurer par l'entropie associée à une distribution, qui, dans le cas présent, correspond à la distribution $P_{\mathcal{V}}$ des valeurs de l'ensemble de variables $\mathcal{V}$ dans le jeu de données obtenue par le tirage aléatoire équiprobable d'une donnée quelconque de $\mathcal{D}$. L'entropie notée $H(\mathcal{V})$ de cette distribution $P_{\mathcal{V}}$ est égale à $H(\mathcal{V}) = -\sum_{v \in \mathbb{D}_{\mathcal{V}}} P_{\mathcal{V}}(v) \cdot \log_2(P_{\mathcal{V}}(v))$. L'entropie d'un ensemble de variables $\mathcal{V}$ est toujours positive ou nulle. Elle est nulle seulement si sa valeur est parfaitement déterminée (i.e. il existe un tuple $v \in \mathbb{D}_{\mathcal{V}}$ tel que $P_{\mathcal{V}}(v) = 1$). Un tel ensemble est dit *parfaitement défini*. Par extension, l'entropie de l'ensemble de variables vide est fixée à zéro : $H(\emptyset) = 0$. Par ailleurs l'entropie ne peut être supérieure à $\log_2(|\mathcal{D}|)$ où $|\mathcal{D}|$ désigne le nombre de données. Cette propriété permet de définir une *entropie relative* $H^r(\mathcal{V})$ d'un ensemble de variables $\mathcal{V}$ comprise dans l'intervalle unité, égale au rapport de l'entropie « absolue » $H(\mathcal{V})$ sur $\log_2(|\mathcal{D}|)$, de la même manière que la *fréquence relative* est le rapport du support d'un itemset sur la taille du jeu de données. Dans la suite, le terme entropie peut faire référence à l'entropie relative ou absolue.

L'entropie est une fonction croissante dans l'ensemble [1] $(2^{\mathbb{V}}, \subseteq)$ des ensembles de variables ordonnés par la relation d'inclusion : $\forall \mathcal{V}_1, \forall \mathcal{V}_2, \mathcal{V}_1 \subseteq \mathcal{V}_2 \Rightarrow H(\mathcal{V}_1) \leq H(\mathcal{V}_2)$. Du fait de cette dernière propriété et pour toute valeur seuil $H_{max} \geq 0$, le prédicat $(H(\mathcal{V}) \leq H_{max})$ d'un ensemble de variables $\mathcal{V}$ est anti monotone dans l'ordre $(2^{\mathbb{V}}, \subseteq)$ des ensembles de variables ordonnés par la relation d'inclusion. Un ensemble de variables vérifiant ce prédicat est dit *bien défini* relativement à H_{max} par analogie avec le caractère fréquent $(\sigma(\mathcal{I}) \geq \sigma_{min})$ d'un itemset $\mathcal{I}$. Par conséquent la *recherche des ensembles de variables bien définis* relativement à la valeur seuil H_{max}, qui consiste à calculer l'entropie de tous les ensembles de variables bien définis, constitue un problème analogue au problème de la recherche des itemsets fréquents. Ce problème a déjà été abordé dans le contexte plus spécifique des itemsets [2, 4] où les itemsets bien définis sont appelés des *motifs de faible entropie* ou *low entropy patterns*. Contrairement à l'entropie, l'entropie conditionnelle tient compte d'une connaissance a priori : étant donnés deux ensembles de variables disjoints $\mathcal{V}_1$ et $\mathcal{V}_2$, l'entropie conditionnelle $H(\mathcal{V}_1|\mathcal{V}_2)$ est définie comme $H(\mathcal{V}_1|\mathcal{V}_2) = H(\mathcal{V}_1 \cup \mathcal{V}_2) - H(\mathcal{V}_2)$. Sa valeur est positive ou nulle. Elle est nulle lorsque la valeur de $\mathcal{V}_1$ se déduit de la valeur de $\mathcal{V}_2$. L'entropie conditionnelle $H(\mathcal{V}_c|\mathcal{V}_h)$ permet de déterminer quelle est l'incertitude résiduelle que présente un ensemble de variables « conclusion » V_c lorsque un ensemble de variables « hypothèse » V_h est connu. Cette entropie exprime donc un lien de dépendance de V_h vers V_c, autrement dit est une mesure associée à une règle $\mathcal{V}_h \rightarrow \mathcal{V}_c$ que l'on nomme *règle de dépendance* dans la suite. Pour cette raison, on notera $H^c(\mathcal{V}_h \rightarrow \mathcal{V}_c)$ l'entropie conditionnelle $H(\mathcal{V}_c|\mathcal{V}_h)$ associée à la règle $\mathcal{V}_h \rightarrow \mathcal{V}_c$. L'entropie conditionnelle de $\mathcal{V}_h \rightarrow \mathcal{V}_c$ est une fonction décroissante de $\mathcal{V}_h$ et croissante de $\mathcal{V}_c$ dans l'ordre des ensembles de variables ordonnés par l'inclusion, à l'inverse de la confiance $\mathrm{conf}(\mathcal{I}_h \rightarrow \mathcal{I}_c) = \frac{\sigma(\mathcal{I}_c \cup \mathcal{I}_h)}{\sigma(\mathcal{I}_h)}$ d'une règle d'association $\mathcal{I}_h \rightarrow \mathcal{I}_c$ qui est une fonction croissante de $\mathcal{I}_h$ et décroissante de $\mathcal{I}_c$. De ce parallèle s'ensuit le problème analogue de l'extraction des règles d'association fréquentes[1] dans le domaine de la fouille d'ensembles de variables : étant

1. La notation $2^{\mathcal{E}}$ désigne l'ensemble des parties de l'ensemble $\mathcal{E}$.

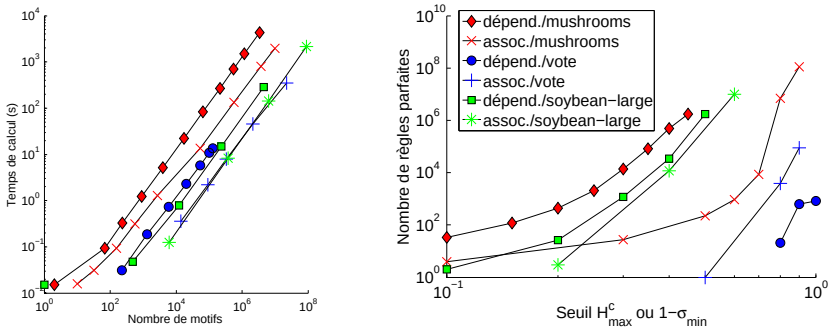
donnés un jeu de données et deux seuils d'entropie H_{max} et H^c_{max}, le problème de l'*extraction des règles de dépendance bien définies* consiste à calculer l'entropie conditionnelle des *règles bien définies* $\mathcal{V}_h \rightarrow \mathcal{V}_c$ telles que l'ensemble de variables $\mathcal{V}_h \cup \mathcal{V}_c$ est bien défini relativement à H_{max} et que $H^c(\mathcal{V}_h \rightarrow \mathcal{V}_c) \leq H^c_{max}$. Le cas $H^c_{max} = 0$ correspond à l'extraction des règles de dépendance dites *parfaites* dont l'entropie conditionnelle est nulle, par analogie avec les règles d'association parfaites de confiance égale à un.

3 L'extraction des règles de dépendance bien définies

Le problème de l'extraction des règles de dépendance bien définies peut être résolu en deux étapes selon le même principe déjà employé pour extraire les règles d'association fréquentes à partir des motifs fréquents [1] : les ensembles de variables bien définis sont d'abord fouillés dans les données avant d'être utilisés dans une seconde étape pour construire les règles de dépendance bien définies. Cette dernière étape peut être réalisée en apportant des modifications mineures à l'algorithme d'extraction des règles d'association fréquente [1]. La principale difficulté réside donc dans le calcul de l'entropie des ensembles de variables bien définis, tâche équivalente au calcul de la fréquence des itemsets fréquents. Une possibilité est d'adapter les principes de l'algorithme Apriori [1] comme cela a déjà été fait notamment dans [3, 2, 4] . Toutefois cette solution n'est pas privilégiée car l'approche par niveau d'Apriori est connue pour souffrir d'une consommation mémoire importante et d'un niveau de performance relativement faible en pratique. La solution proposée s'inspire de l'algorithme Eclat de recherche des itemsets fréquents [7]. L'idée principale d'Eclat est d'utiliser un codage dit « vertical » du jeu de données et la propriété selon laquelle la couverture d'un itemset $\mathcal{I}$ est l'intersection des couvertures de tous les items $i \in \mathcal{I}$. H-Eclat repose ainsi sur une propriété analogue à celle utilisée par Eclat selon laquelle la partition $\mathcal{P}_\mathcal{D}(\mathcal{V}_1 \cup \mathcal{V}_2)$ de l'union de deux ensembles de variables $\mathcal{V}_1$ et $\mathcal{V}_2$ est l'intersection $\mathcal{P}_\mathcal{D}(\mathcal{V}_1) \cap \mathcal{P}_\mathcal{D}(\mathcal{V}_2)$ des deux partitions de $\mathcal{V}_1$ et $\mathcal{V}_2$ [2]. Les principales différences avec Eclat sont que les itemsets sont remplacés par des ensembles de variables, que la contrainte anti-monotone $(\sigma(\mathcal{I}) \geq \sigma_{min})$ devient $(H(\mathcal{V}) \leq H_{max})$ et surtout que, les couvertures des itemsets sont remplacées par des *listes de parties*, c'est-à-dire des listes de couvertures représentant des partitions de données et dont l'intersection peut se calculer selon une complexité linéaire par rapport au nombre de données.

Les tests réalisés ont visé d'une part à estimer les performances de l'algorithme H-Eclat et de l'extraction des règles de dépendance bien définies vis-à-vis de celles de l'algorithme Eclat et de l'extraction des règles d'association fréquentes et d'autre part, à estimer qualitativement l'utilité des règles de dépendance bien définies par rapport aux règles d'association fréquentes. La figure 2(a) représente le temps de calcul exprimé en fonction du nombre de motifs découverts sur les jeux de données `mushrooms`, `soybeans` et `vote` pour différentes valeurs seuil H_{max} et σ_{min}. Les temps de calcul moyens par motif croissent linéairement avec le nombre de motifs découverts comme cela était prévisible. H-Eclat est plus lent que Eclat car l'intersection de partitions est une opération plus complexe que l'intersection de deux ensembles. Le rapport des performances dépend du jeu de données mais pas des valeurs seuil. Cette relative lenteur est toutefois compensée par le nombre d'ensembles de variables à rechercher bien inférieur à celui des itemsets correspondants : par exemple, H-Eclat extrait en 10 secondes tous les 2^{17}=131072

2. L'intersection de deux partitions est la plus grand partition incluse dans chacune des deux partitions, une partition P_1 étant incluse dans une partition P_2 si toutes les parties de P_1 sont incluses dans une des parties de P_2.



(a) Temps de calcul de H-Eclat et Eclat fonction du nombre de motifs extraits

(b) Nombre de règles de dépendance bien définies et de règles d'association fréquentes

FIG. 2 – *Temps de calcul et nombres de règles*

ensembles possibles de variables pour le jeu `vote` (pour $H^r_{max} = 1$) quand Eclat trouve pour le même temps de calcul 500K d'itemsets fréquents, soit seulement $2,5\%$ des 22 millions d'itemsets ayant une fréquence non nulle. Le même phénomène est observé entre règles de dépendance bien définies et règles d'association fréquentes. Ainsi selon la figure 2(b), `vote` contient au total seulement 835 règles de dépendance parfaites ($H^c_{max} = 0$ et $H^r_{max} = 1$) à comparer à un peu moins d'un million de règles d'association parfaites ($c_{min} = 1$ et $\sigma_{min} = 0$). Contrairement à la recherche des motifs bien définis ou fréquents, les temps de calcul moyens pour extraire règles de dépendance ou d'association sont approximativement les mêmes car contrairement à la recherche des motifs bien définis ou fréquents, l'algorithme utilisé est le même. Du point de vue de l'extraction de connaissances, les règles de dépendance permettent d'exprimer une information plus synthétique mais parfois aussi moins précise que celles des règles d'association. Par exemple, la règle de dépendance parfaite extraite du jeu de données `mushrooms` « `forme du pied, taille des lamelles, changement de couleur, couleur des spores` → `comestibilité` » permet de prédire si un champignon est comestible ou non. Cette seule règle équivaut à la donnée de 55 règles d'association parfaites ayant pour hypothèse les 55 jeux de valeurs que prennent dans les données les quatre variables composant l'hypothèse de la règle de dépendance. D'un autre côté, les règles d'association permettent une analyse à un niveau de granularité plus fin. Ainsi la règle d'association parfaite `forme de tige effilée, pas d'odeur` → `comestible` permet d'identifier un cas particulier de champignon comestible quand la règle de dépendance `forme de tige, odeur` → `comestibilité` n'est pas parfaite. Règles de dépendance et règles d'association apportent deux points de vues utiles et complémentaires.

4 Conclusions

Cet article compare et oppose les règles de dépendance entre ensembles de variables multivaluées aux règles d'association entre itemsets. Les premières présentent ainsi l'avantage sur les secondes d'être moins nombreuses et plus synthétiques avec toutefois l'inconvénient d'être

moins précises. L'article suggère aussi en perspective que l'analogie formelle entre règles de dépendance et règles d'association fréquentes qui a permis d'adapter Eclat, soit davantage exploitée pour adapter les nombreuses techniques développées dans le cadre des itemsets fréquents au cas de la fouille des ensembles de variables multivaluées.

Références

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases, (VLDB'94), Santiago de Chile, Chile*, pages 478–499. Morgan Kaufmann, 1994.

[2] H. Heikinheimo, J. K. Seppänen, E. Hinkkanen, H. Mannila, and T. Mielikäinen. Finding low-entropy sets and trees from binary data. In P. Berkhin, R. Caruana, and X. Wu, editors, *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA*, pages 350–359. ACM, 2007.

[3] Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen. Tane : An efficient algorithm for discovering functional and approximate dependencies. *Comput. J.*, 42(2) :100–111, 1999.

[4] M. Mampaey. Mining non-redundant information-theoretic dependencies between itemsets. In *Proceedings of the 12th International Conference on Data Warehousing and Knowledge Discovery DAWAK 2010, Bilbao, Spain*, volume 6263 of *Lecture Notes in Computer Science*, pages 130–141. Springer, 2010.

[5] R. Medina and L. Nourine. A unified hierarchy for functional dependencies, conditional functional dependencies and association rules. In S. Ferré and S. Rudolph, editors, *Formal Concept Analysis, 7th International Conference, ICFCA 2009, Darmstadt, Germany, May 21-24, 2009, Proceedings*, volume 5548 of *Lecture Notes in Computer Science*, pages 98–113. Springer, 2009.

[6] D. Sánchez, J. M. Serrano, I. Blanco, M. J. Martín-Bautista, and M.-A. Vila. Using association rules to mine for strong approximate dependencies. *Data Min. Knowl. Discov.*, 16 :313–348, June 2008.

[7] M. J. Zaki. Scalable algorithms for association mining. *IEEE T. Knowl. Data. En.*, 12(3) :372–390, 2000.

Summary

This article studies the feasibility and interest for extracting dependance rules between sets of multivalued variables comparatively to those of frequent association rules. A dependency rule is an approximate functional dependency mainly characterized by conditional entropy. The article shows how to establish a formal analogy between both types of rules and how to adapt thanks to this analogy the algorithm "Eclat" in order to extract from a dataset said definite dependance rules. An experimental study concludes on pros & cons of dependance rules compared to association rules.