Contribution à l'atelier Bases de Données Inductives BDI'07 organisé dans le cadre de la plate-forme AFIA 2007, Grenoble, juillet 2007.

Extraction d'hyper-rectangles fermés sous contraintes

L. Cerf[1], J. Besson[2], C. Robardet[1] et J-F. Boulicaut[1]

[1] *LIRIS / CNRS UMR 5205, INSA de Lyon, 69621 Villeurbanne, France*
[2] *INRA*

L'extraction de motifs locaux sous contraintes est une tâche centrale pour la découverte de connaissances dans les bases de données. De nombreux travaux ont porté sur la définition de nouveaux types de motifs pertinents dans des relations binaires, ainsi que sur le développement d'algorithmes efficaces pour les extraire. Parmi les motifs ensemblistes, les itemsets fermés fréquents ou concepts formels sont apparus comme étant particulièrement bien adaptés dans ce type de données, permettant de capturer les associations localement fortes de la relation binaire. Nous proposons dans cet article de généraliser ce type de motifs pour des relations d'ordre supérieur, développement rendu nécessaire notamment par l'intérêt croissant pour l'étude de données dynamiques. Nous proposons un algorithme efficace permettant de calculer ces motifs grâce à un résumé des données qui permet de réduire les calculs.

Mots clefs : Fouille de données, bases de données, motifs locaux ensemblistes, contraintes, n dimnsions.

1 Introduction

L'étude des relations binaires a suscité ces dernières années un grand intérêt de par notamment le développement d'algorithmes d'extraction de motifs locaux sous contraintes qui permettent d'identifier des régularités localement fortes dans ce type de données. De nombreux domaines d'application produisent à l'heure actuelle des données relationnelle d'ordre supérieur, notamment les applications visant à étudier la dynamique de systèmes, e.g. l'étude de la dynamique des gènes dans des situations biologiques particulières. Nous nous proposons dans cet article d'étendre le cadre formel d'extraction de motifs fermés sous contraintes aux cas de données relationnelles d'ordre n. Des travaux récents s'inscrivent dans ce cadre. Dans [1], les auteurs définissent la notion de *cube fermé fréquent* pour les données ternaires. Un tel cube est formé de trois sous-ensembles, un pour chaque dimension, tels que toutes les associations portées par les éléments de ces ensembles appartiennent à la relation. De plus ces ensembles sont fermés c'est-à-dire aucun élément ne peut être ajouté à l'un des trois ensembles sans rendre la propriété précédente fausse. Les auteurs proposent deux algorithmes. Le premier est assez naïf et consiste à générer tous les sous-ensembles possibles de la plus petite dimension puis à calculer l'ensemble des concepts formels sur la projection des données sur chaque sous-ensemble. Par post-traitement, la contrainte de fermeture est vérifié sur chaque motif extrait. Dans leur second algorithme, ils proposent d'utiliser un ensemble de *cutters*, des ensembles d'éléments non associés par la relation, pour découper l'espace de recherche. Quatre contraintes, assez coûteuses à vérifier, sont alors utilisées pour assurer la justesse de l'extraction.

Le travail présenté ici est une généralisation à l'ordre n de ce problème. Nous proposons un cadre théorique ainsi qu'un nouvel algorithme basé sur l'utilisation de représentations partielles des données, des résumés. Cet ensemble permet de résumer l'espace de recherche de telle sorte à pouvoir vérifier efficacement la contrainte de fermeture des motifs recherchés et de propager efficacement les contraintes.

2 Contribution

Soit n ensembles $D^1, \cdots, D^n$ et $\mathcal{R}$ une relation n-aire sur ces ensembles, i.e. un sous-ensemble du produit Cartesien $D^1 \times \cdots \times D^n$. Un Hyper-rectangle fermé $\langle X^1, \cdots, X^n \rangle$ est alors un ensemble ordonné d'ensembles tel que

	x_1^1	x_2^1	x_3^1	x_1^1	x_2^1	x_3^1	x_1^1	x_2^1	x_3^1
x_1^2	×	×	×	×	×	×	×	×	
x_2^2	×	×		×			×	×	
x_3^2		×				×	×		×
x_4^2			×	×		×	×	×	×
	x_1^3			x_2^3			x_3^3		

FIG. 1: Un exemple de données tri-dimensionnelle $\mathcal{R}_{ex}$

– $X^j \subseteq D^j$.
– les éléments de chaque dimension sont complètement connectés aux éléments des autres dimensions : $\forall j \in \{1 \cdots n\}, \forall x^j \in X^j, (x^1, \cdots, x^n) \in \mathcal{R}$.
– il est fermé : $\forall j \in \{1 \cdots n\}, \forall x^j \in D^j \setminus X^j, (X^1, \cdots, X^j \cup \{x^j\}, \cdots, X^n)$ n'est pas complètement connecté.

Example 1. *La Figure 1 présente un exemple de données booléennes tri-dimensionnelles.*

$\langle (x_1^1, x_2^1), (x_1^2, x_2^2), (x_1^3, x_3^3) \rangle$ *et* $\langle (x_3^1), (x_4^2), (x_1^3, x_2^3, x_3^3) \rangle$ *sont deux exemples de HRF dans* $\mathcal{R}_{ex}$.

L'algorithme que nous proposons calcule l'ensemble des HRF qui satisfont des contraintes de taille minimale sur chacune des dimensions, i.e. $\forall i = 1 \cdots n, |X^i| \geq MinTaille_i$ où $MinTaille_i$ est un paramètre défini par l'utilisateur. Il procède par réduction récursive de l'espace de recherche à la fois par énumération et par propagation d'éléments. L'énumération d'un élément $x^j \in D^j$ consiste à découper l'espace de recherche en deux : un espace possédant tous les HRF contenant x^j, et un autre possédant ceux ne contenant pas x^j. Comme pour le calcul des concepts formels, la contrainte de fermeture doit être vérifiée dans certains cas. Afin de réduire la quantité de données stockées à chaque énumération et de réduire le coût de la vérification de la fermeture des motifs, nous proposons d'utiliser des projections partielles des données. Au lieu de considérer le jeu de données complet, on ne s'intéresse qu'à sa projection sur chacune des dimensions en ne conservant que les associations pouvant appartenir aux HRF calculés sur l'espace de recherche considéré. Ce résumé des données, calculé très efficacement à l'aide de l'opérateur **et** bit à bit, permet de propager autant que si l'on considérait le jeu de données dans son intégralité. En effet, les éléments qui ne sont pas associés avec toutes les combinaisons d'éléments déjà énumérés sur les autres dimensions, n'ont pas besoin d'être énumérés, car on est sûr qu'ils ne peuvent appartenir à aucun HRF. Cette information est alors utilisée pour restreindre le nombre d'énumérations. De plus cette structure de données permet aisément de vérifier la contrainte de taille minimale.

Les expérimentations montrent que l'algorithme proposé est plus efficace en trois dimensions que l'algorithme proposé dans [1], et lorsqu'aucune dimension n'est plus petite que les autres, il est également plus performant que [2]. Des applications montrent l'intérêt de pouvoir extraire de tels motifs en dimension supérieure.

Références

[1] Liping Ji, Kian-Lee Tan, and Anthony K. H. Tung. Mining frequent closed cubes in 3d datasets. In *VLDB'2006 : Proceedings of the 32nd international conference on Very large data bases*, pages 811–822. VLDB Endowment, 2006.

[2] Robert Jaschke, Andreas Hotho, Christoph Schmitz, Bernhard Ganter, and Gerd Stumme. Trias–an algorithm for mining iceberg tri-lattices. In *ICDM '06 : Proceedings of the Sixth International Conference on Data Mining*, pages 907–911, Washington, DC, USA, 2006. IEEE Computer Society.