Nouvelle approche de fouille de graphes AC-réduits fréquents

Brahim Douar*,** Michel Liquière*, Chiraz Latiri**, Yahya Slimani**

*Equipe COCONUT, LIRMM, 161 rue Ada 34095 - Montpellier, France
{douar,liquiere}@lirmm.fr
**URPAH, Département informatique, Faculté des Sciences de Tunis, Tunisie
chiraz.latiri@gnet.tn, yahya.slimani@fst.rnu.tn

Résumé. La fouille de graphes est devenue une piste de recherche intéressante et un défi réel en matière de fouille de données. Parmi les différentes familles de motifs de graphes, les graphes fréquents permettent une caractérisation intéressante des groupes de graphes, ainsi qu'une discrimination des différents graphes lors de la classification ou de la segmentation. A cause de la NP-complétude du test d'isomorphisme de sous-graphes et de l'immensité de l'espace de recherche, les algorithmes de fouille de graphes sont exponentiels en temps d'exécution et/ou occupation mémoire. Dans cet article, nous étudions un nouvel opérateur de projection polynomial nommé AC-projection basé sur une propriété clé du domaine de la programmation par contraintes, à savoir l'arc consistance. Cet opérateur est censé remplacer l'utilisation de l'isomorphisme de sous-graphes en établissant un biais sur la projection. Cette étude est suivie d'une évaluation expérimentale du pouvoir discriminant des patterns AC-réduits découverts.

1 Introduction

Avec la croissance importante du besoin d'analyser une grande masse de données structurées tels que les composés chimiques, les structures de protéines ou même les documents XML, pour n'en citer que quelques-uns, la fouille de graphes est devenue une problématique de recherche intéressante et un défi réel en matière de fouille de données. En effet, la découverte de sous-graphes fréquents est un réel challenge vu leur nombre exponentiel. A ceci s'ajoute la NP-complétude du problème d'isomorphisme d'un sous-graphe général (Garey et Johnson, 1979). Dans cet article, nous introduisons la notion de biais de projection afin de proposer un opérateur similaire à l'isomorphisme de sous-graphes mais ayant une complexité polynomiale et des contraintes relaxées. Nous utilisons ensuite cet opérateur dans un processus de fouille de graphes.

2 Fondements et travaux connexes

Nous présentons dans ce qui suit les fondements mathématiques du domaine de la fouille de graphes. Cette présentation est accompagnée d'une brève citation des travaux connexes et

particulièrement d'une approche en largeur sur laquelle se base notre algorithme de fouille de graphes.

Définition 2.1 *(Graphe étiqueté) Un graphe étiqueté peut être représenté par un quadruplet,* $G = (S, A, E, e)$*, avec*

- S *est un ensemble de sommets,*
- $A \subseteq S \times S$ *est un ensemble d'arêtes,*
- E *est un ensemble d'étiquettes,*
- $e : S \cup A \rightarrow E$, e *est une fonction qui affecte les étiquettes aux sommets et aux arêtes du graphe.*

Définition 2.2 *(Isomorphisme, Isomorphisme de sous-graphes) Soient deux graphes* G_1 *et* G_2*, un isomorphisme est une fonction bijective* $f : S(G_1) \rightarrow S(G_2)$*, telle que*

- $\forall x \in S(G_1), e(x) = e(f(x))$*, et*
- $\forall (x, y) \in A(G_1), (f(x), f(y)) \in A(G_2)$ *et* $e(x, y) = e(f(x), f(y))$.

Un isomorphisme de sous-graphes de G_1 *vers* G_2 *est un isomorphisme de* G_1 *vers un sous-graphe de* G_2*.*

Définition 2.3 *(Fouille de graphes) Soit un ensemble de graphes, GS=*$\{G_i | \ i = 0.....n\}$*, et un support minimal (minSup), on pose*

$$\varsigma(g, G) = \begin{cases} 1 & \textit{s'il existe une projection de } g \textit{ vers } G \\ 0 & \textit{sinon.} \end{cases}$$

$$\sigma\,(g, GS) = \sum_{G_i \,\in\, GS} \varsigma(g, G_i)$$

$\sigma\,(g, GS)$ *représente la fréquence d'apparition de* g *dans* GS*, c.a.d, le support de* g *dans* GS*. Le processus de fouille de graphes consiste à trouver tout graphe* g *tel que* $\sigma\,(g, GS)$ *est supérieur ou égal à minSup.*

Les algorithmes de fouille de graphes sont généralement basés sur cette définition et traitent avec le cas particulier où l'opérateur de projection est l'isomorphisme de sous-graphes. Ces algorithmes se basent sur deux paradigmes de parcours de l'espace de recherche à savoir, le parcours en largeur et le parcours en profondeur. Leur but étant de trouver les sous-graphes connectés ayant un nombre suffisant d'arêtes dans un seul graphe éparse non dirigé. La plupart de ces algorithmes utilisent des méthodes différentes pour énumérer et générer les motifs candidats. Une comparaison quantitative intéressante des approches de fouille de graphes les plus cités est donnée dans (Wörlein et al., 2005).

L'approche de fouille de graphes que nous présentons plus loin dans cet article se base sur une approche en largeur intensivement citée dans la littérature. Il s'agit de l'approche de fouille de graphe inspiré d'Apriori (Agrawal et Skirant, 1994) nommée FSG et décrite dans (Kuramochi et Karypis, 2001, 2004).

3 AC-projection

L'AC-projection, initialement introduite dans (Liquiere, 2007), propose un opérateur de projection basé sur l'algorithme de l'arc consistance issue du domaine de la programmation par contraintes (Bessiere, 2006). Cette méthode de projection possède des propriétés intéressantes, à savoir la polynomialité, la validation locale, la parallélisation, l'interprétation structurelle, etc.

Définition 3.1 *(Mappage) Soient deux graphes étiquetés G_1 et G_2. Nous appelons mappage de G_1 dans G_2 la correspondance $\mathcal{I} : S(G_1) \rightarrow 2^{S(G_2)} | \forall x \in S(G_1), \forall y \in \mathcal{I}(x), e(x) = e(y)$.*

Définition 3.2 *(AC-compatible $\curvearrowright$) Soit un graphe G, $S_1 \subseteq S(G), S_2 \subseteq S(G)$*
S_1 est AC-compatible avec S_2 ssi
– $\forall x_k \in S_1, \exists y_p \in S_2 | (x_k, y_p) \in A(G)$
– $\forall y_q \in S_2, \exists x_m \in S_1 | (x_m, y_q) \in A(G)$.
On note $S_1 \curvearrowright S_2$

Définition 3.3 *(Consistance d'un arc) Soient deux graphes G_1 et G_2. Un mappage $\mathcal{I} : S(G_1) \rightarrow 2^{S(G_2)}$ est consistant avec un arc $(x, y) \in A(G_1)$, ssi $\mathcal{I}(x) \curvearrowright \mathcal{I}(y)$.*

Définition 3.4 *(AC-projection $\rightharpoondown$) Soient deux graphes G_1 et G_2. Un mappage $\mathcal{I}$ de G_1 dans G_2 est une AC-projection ssi $\mathcal{I}$ est consistant avec tous les arcs $e \in A(G_1)$. On note $G_1 \rightharpoondown G_2$. Nous notons que l'AC-projection est calculable en un temps polynomial avec une complexité en $O(a \times s^2)$ avec « a » désignant le nombre d'arêtes de G_1 et « s » le nombre maximal de sommets de même étiquette de G_2.*

Définition 3.5 *(AC-équivalence $\leftrightharpoons$)*
Deux graphes G_1 et G_2 sont AC-équivalents ssi $G_1 \rightharpoondown G_2$ et $G_2 \rightharpoondown G_1$.
On note $G_1 \leftrightharpoons G_2$.

Nous avons alors une relation d'équivalence entre graphes en utilisant l'AC-projection. Ceci fait alors apparaitre des classes d'équivalence entre graphes. Chaque classe d'équivalence est représenté par un élément minimal que nous nommons "graphe AC-réduit".

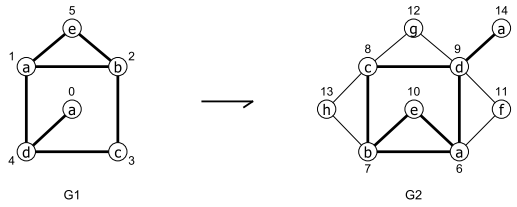

FIG. 1 – *Un exemple d'AC-projection ($G_1 \rightharpoondown G_2$)*

4 FGMAC : Nouvelle approche de fouille de graphes

Dans cette section, nous présentons FGMAC [1], une version modifiée de FSG (Kuramochi et Karypis, 2001) basée sur l'AC-projection. En effet, dans cette version nous avons changé la fonction dédiée pour le calcul du support d'un pattern, l'AC-projection est désormais utilisée à la place de l'isomorphisme de sous-graphes pour vérifier si un graphe candidat apparait dans une transaction donnée ou pas.

L'algorithme FGMAC commence par énumérer tous les graphes fréquents ayant une ou deux arêtes. Ensuite, en se basant sur ces deux ensembles, un processus itératif est amorcé. Lors de chaque itération, il commence par générer les graphes candidats ayant une taille supérieure d'une seule arête par rapport aux fréquents de l'itération précédente (Algorithme 1, ligne 5). Il calcule après la fréquence de chaque graphe moyennant l'AC-projection comme opérateur de projection et non pas l'isomorphisme de sous-graphes. L'algorithme élague les sous-graphes qui ne satisfont pas la contrainte du support minimal (Algorithme 1, lignes 6-11).

La particularité de l'algorithme FGMAC est de ne retourner que les graphes AC-réduits fréquents (Algorithme 1, ligne 11) qui ne représente qu'un sous-ensemble des graphes fréquents à l'isomorphisme près.

Algorithme 1: FGMAC

Entrées : Un dataset de graphes D, Support minimal σ
Sorties : L'ensemble des graphes AC-réduits fréquents F

1 $F^1 \leftarrow$trouver tous les graphes fréquents à 1 arête dans D ;
2 $F^2 \leftarrow$trouver tous les graphes fréquents à 2 arêtes dans D ;
3 $k \leftarrow 3$;
4 **tant que** $F^{k-1} \neq \emptyset$ **faire**
5 $\quad C^k \leftarrow$ `fsg-gen` (F^{k-1})
6 $\quad$ **pour chaque** *candidat* $g^k \in C^k$ **faire**
7 $\quad\quad g^k.count \leftarrow 0$;
8 $\quad\quad$ **pour chaque** *transaction* $t \in D$ **faire**
9 $\quad\quad\quad$ **si** $g^k \rightarrowtail t$ **alors**
10 $\quad\quad\quad\quad g^k.count \leftarrow g^k.count + 1$;
11 $\quad F^k \leftarrow \{$`AC-reduce`$(g^k \in C^k) | g^k.count \geq \sigma|D|\}$;
12 $\quad k \leftarrow k + 1$;
13 **retourner** F ;

5 Étude expérimentale

En vu de montrer l'intêret de recourir à l'AC-projection pour la fouille de graphes et souligner son apport, nous présenterons dans ce qui suit une évaluation expérimentale de l'approche. Nous tenons à préciser que l'ensemble des graphes AC-réduits fréquents trouvés par FGMAC

1. « **F**requent sub**G**raph **M**ining with **A**rc **C**onsistency »

n'est pas exhaustif par rapport aux motifs à l'isomorphisme près. Nous présentons alors dans ce qui suit une évaluation qualitative des motifs AC-réduits qui consiste à mesurer le pouvoir discriminant de ces motifs dans un processus de classification supervisée de graphes. Pour nos évaluations expérimentales nous avons sélectionné cinq bases de graphes largement citées dans la littérature à savoir PTC-FM, PTC-FR, PTC-MM, PTC-MR et HIA (Smalter et al., 2008).

Nous commençons notre démarche expérimentale par une propositionnalisation des motifs issues de la fouille de graphes. Chaque transaction du dataset est représentée par un vecteur binaire avec une taille égale au nombre de motifs fréquents. Chaque motif est associé à une position spécifique du vecteur, et si la transaction en question contient le motif alors le bit à la position correspondante est défini à un, sinon il sera alors défini à zéro. En connaissant la classe de chaque transaction, nous lançons alors un processus de classification supervisée par arbre de décision (C4.5 (Quinlan, 1993)) et nous effectuons une validation croisée en considérant le pcc [2] comme métrique d'évaluation du pouvoir discriminant des motifs.

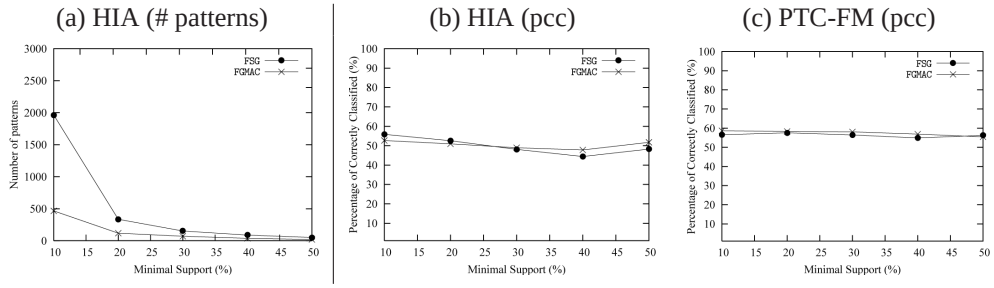

FIG. 2 – *(a) Comparaison des nombres de patterns générés par FGMAC et de FSG, (b,c) Comparaison du pcc associé aux datasets PTC-FM et HIA*

La courbe (a) de la figure 2 montre que le nombre de patterns fréquents découverts par FGMAC est inférieur aux patterns à l'isomorphisme près découverts par FSG pour tous les supports. Les deux courbes b et c montrent un pcc comparable pour FGMAC et FSG ce qui prouve qu'expérimentalement les patterns approximés trouvés par FGMAC sont aussi expressifs que l'ensemble exhaustif des sous-graphes à l'isomorphisme près.

6 Conclusion

Dans cet article nous avons introduit la notion de biais de projection pour la fouille de graphes. Nous avons utilisé l'AC-projection, un opérateur polynomial, à la place de l'isomorphisme de sous-graphes dans un processus de fouille de graphes. Dans un deuxième temps nous comptons proposer une approche en profondeur de fouille de graphes basé sur l'AC-projection et assurant une meilleure mise à l'échelle pour les grands graphes.

2. Pourcentage de classifications correctes

Références

Agrawal, R. et R. Skirant (1994). Fast algorithms for mining association rules. In *proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile*, pp. 478–499.

Bessiere, C. (2006). Constraint propagation. In F. Rossi, P. van Beek, et T. Walsh (Eds.), *Handbook of Constraint Programming*, Chapter 3. Amsterdam : Elsevier.

Garey, M. R. et D. S. Johnson (1979). *Computers and Intractability : A Guide to the Theory of NP-Completeness*. New York, NY, USA : W. H. Freeman & Co.

Kuramochi, M. et G. Karypis (2001). Frequent subgraph discovery. In N. Cercone, T. Y. Lin, et X. Wu (Eds.), *International Conference on Data Mining*, pp. 313–320. IEEE Computer Society.

Kuramochi, M. et G. Karypis (2004). An efficient algorithm for discovering frequent subgraphs. *IEEE Transactions on Knowledge and Data Engineering 16*, 1038–1051.

Liquiere, M. (2007). Arc consistency projection : A new generalization relation for graphs. In U. Priss, S. Polovina, et R. Hill (Eds.), *ICCS*, Volume 4604 of *LNCS*, pp. 333–346. Springer.

Quinlan, J. R. (1993). *C4.5 : Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)* (1 ed.). Morgan Kaufmann.

Smalter, A. M., J. Huan, et G. H. Lushington (2008). Chemical compound classification with automatically mined structure patterns. In A. Brazma, S. Miyano, et T. Akutsu (Eds.), *APBC*, Volume 6 of *Advances in Bioinformatics and Computational Biology*, pp. 39–48. Imperial College Press.

Wörlein, M., T. Meinl, I. Fischer, et M. Philippsen (2005). A quantitative comparison of the subgraph miners mofa, gspan, ffsm, and gaston. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Volume 3721 of *LNCS*, pp. 392–403. Springer.

Summary

Graph mining is an attractive track and a real challenge in the data mining field. Among the various kinds of graph patterns, frequent subgraphs seem to be relevant in characterizing graphsets, discriminating different groups of sets, and classifying and clustering graphs. Because of the NP-Completeness of subgraph isomorphism test as well as the huge search space, fragment miners are exponential in runtime and/or memory consumption. In this paper we study a new polynomial projection operator named AC-Projection based on a key technique of constraint programming namely Arc Consistency (AC). This is intended to replace the use of the exponential subgraph isomorphism. We study the relevance of frequent AC-reduced graph patterns on classification.