Granularité des motifs de co-variations dans des graphes attribués dynamiques

Elise Desmier[*,**], Marc Plantevit[*,***], Jean-François Boulicaut[*,**]

[*]Université de Lyon, CNRS
[**]INSA-Lyon, LIRIS, UMR5205, F-69621 Villeurbanne, France
[***]Université Lyon 1, LIRIS, UMR5205, F-69622 Villeurbanne, France

Résumé. Découvrir des connaissances dans des graphes qui sont dynamiques et dont les sommets sont attribués est de plus en plus étudié, par exemple dans le contexte de l'analyse d'interactions sociales. Il est souvent possible d'expliciter des hiérarchies sur les attributs permettant de formaliser des connaissances a priori sur les descriptions des sommets. Nous proposons d'étendre des techniques de fouille sous contraintes récemment proposées pour l'analyse de graphes attribués dynamiques lorsque l'on exploite de telles hiérarchies et donc le potentiel de généralisation/spécialisation qu'elles permettent. Nous décrivons un algorithme qui calcule des motifs de co-évolution multi-niveaux, c'est-à-dire des ensembles de sommets qui satisfont une contrainte topologique et qui évoluent de la même façon selon un ensemble de tendances et de pas de temps. Nos expérimentations montrent que l'utilisation d'une hiérarchie permet d'extraire des collections de motifs plus concises sans perdre d'information.

1 Introduction

Les graphes sont un puissant outil de représentation pour la découverte de connaissances dans de nombreux contextes. Ainsi, nous pouvons nous intéresser à des graphes qui décrivent des entités (nœuds) mises en relation (arêtes) : souvent, ces entités peuvent être décrites au moyen d'attributs et les relations ou descriptions des attributs peuvent évoluer au cours du temps. Nous parlerons alors de graphe attribué et dynamique (Jin et al. (2007); Boden et al. (2012)). Développer de nouvelles méthodes pour la fouille de tels graphes est important, ne serait-ce que pour le potentiel applicatif des analyses d'interactions sociales. L'explicitation puis l'exploitation de hiérarchies déclarant certaines relations entre attributs a été très étudié, notamment dans le contexte de la découverte de motifs et de règles d'association multi-dimensionnelles (Srikant et Agrawal (1996); Han et Fu (1999); Chen et al. (2009); Plantevit et al. (2010)). Nous pensons que de telles hiérarchies, souvent faciles à expliciter, permettraient d'ajouter de la connaissance du domaine sur des graphes attribués dynamiques pour améliorer la pertinence des fouilles réalisées. Nous décidons d'étendre la proposition présentée dans Desmier et al. (2013) pour la découverte de motifs de co-évolution dans des graphes attribués dynamiques : chaque motif découvert va correspondre à trois ensembles qui sont (a) un ensemble de nœuds, (b) un ensemble de pas de temps, et (c) un ensemble d'attributs tel

qu'une tendance d'évolution (croissance, décroissance) est associée à chaque attribut. Prenons par exemple un graphe représentant un ensemble d'aéroports pour les noeuds, reliés deux à deux par une arête s'il existe au moins un vol direct entre les deux d'aéroports et dont les attributs représentent le nombre de vols au départ et à l'arrivée de chaque aéroport. Un motif pourrait par exemple être un groupe de 9 aéroports qui ont vu leur nombre de vols au départ et à l'arrivée diminuer en décembre car il y a eu de fortes chutes de neige. Cependant ce type de motif peut entrainer beaucoup de redondances dûes à la nécessité d'avoir un respect strict des tendances sur les attributs. Supposons par exemple que 11 aéroports ont vu leur nombre de vols au départ diminuer, 10 ont vu leur nombre de vols à l'arrivée diminuer tandis que 9 d'entre eux ont eu les deux à la fois : dans une approche comme celle de Desmier et al. (2013), trois motifs seront alors extraits. Afin d'améliorer la pertinence des motifs de co-évolution calculés, nous introduisons ici l'utilisation d'une hiérarchie sur les attributs. Nous supposons que la hiérarchie est fournie par l'analyste et nous étudions la découverte de motifs multi-niveaux, c'est à dire pouvant contenir des éléments appartenant à plusieurs niveaux de la hiérarchie. Nous proposons une mesure de pureté pour vérifier que la tendance est respectée par un certain pourcentage des attributs fils. Cette mesure permet d'accepter qu'un ensemble de nœuds et de pas de temps ne respecte pas strictement la tendance des attributs bien qu'il apporte une information suffisamment pertinente. Dans notre exemple, considérons une hiérarchie qui expliquerait que l'attribut « Nombre de vols » est le parent du nombre d'arrivée et du nombre de départs. Le motif extrait présenterait alors un groupe de 12 aéroports qui ont vu leur nombre de vols diminuer en décembre, en incluant ceux qui ont vu leur nombre de vols diminuer soit au départ soit à l'arrivée. Cette multiplicité des niveaux des attributs permet d'obtenir des motifs plus concis et donc d'une certaine façon plus robustes au bruit ou aux erreurs : des motifs plus généraux sont découverts dont les nœuds ne respectent pas une co-évolution stricte. Il devient donc possible d'intégrer au motif des entités ou des pas de temps pour lesquels la co-évolution n'est en partie pas respectée parce qu'ils ont eu temporairement un comportement différent.

Nos contributions sont les suivantes. Nous définissons un nouveau problème de fouille de données : la découverte de motifs hiérarchiques de co-évolution dans des graphes attribués dynamiques. Nous définissons ce type de motif comme une séquence de graphes connexes et dont les attributs co-évoluent, et nous introduisons des mesures qui permettent d'évaluer leur pureté. Ensuite, nous proposons un algorithme qui calcule l'ensemble des motifs qui satisfont une combinaison de contraintes spécifiée par l'analyste. Une formalisation du problème et des mesures est faite dans la Section 2 et l'algorithme est décrit dans la Section 3. Les résultats des expérimentations quantitatives et qualitatives sont présentés dans la Section 4. Un état de l'art est proposé dans la Section 5 avant une brève conclusion en Section 6.

2 Motifs Hiérarchiques de Co-évolution

Un graphe attribué dynamique $\mathcal{G}$ se définit comme une séquence de graphes attribués G_t sur T pas de temps, i.e., $G = \{G_t | t = 1..T\}$ avec $G_t = (\mathcal{V}, E_t, A_t)$. $\mathcal{V}$ est un ensemble de nœuds, E_t est un ensemble d'arêtes qui dépendent du temps et A_t est un vecteur de valeurs pour les attributs de $\mathcal{A}$ au temps t. $\mathcal{A}$ est un ensemble d'attributs commun à tous les nœuds et à tous les temps. Une hiérarchie $\mathcal{H}$ sur $\mathcal{A}$ est un arbre dont le nœud $\mathcal{A}ll$ est la racine et les attributs de $\mathcal{A}$ les feuilles. Les arcs représentent une relation is_a. La relation de spécialisation (resp. généralisation) correspond à un parcours de haut en bas (resp. de bas en haut), c'est à dire de la

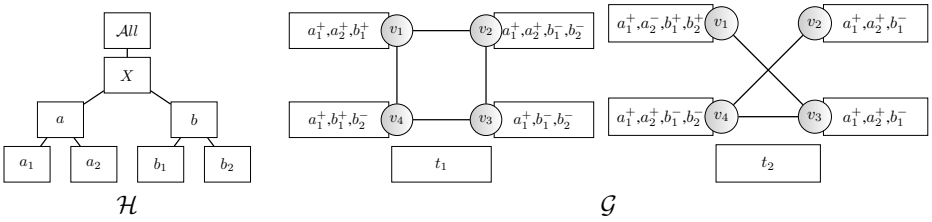


FIG. 1 – *Exemple jouet.*

racine aux feuilles (resp. des feuilles à la racine). Cette hiérarchie représente une connaissance a priori de l'analyste. La fonction $parent(x)$ retourne les nœuds parents directs du nœud x, $children(x)$ les nœuds fils directs, $up(x)$ l'ensemble des ancêtres de x, x inclus, $down(x)$ l'ensemble de ses descendants, x inclus. La fonction $leaf(x)$ retourne les descendants de x qui sont des feuilles de la hiérarchie, i.e., $leaf(x) = down(x) \cap \mathcal{A}$. Le domaine de la hiérarchie $dom(\mathcal{H})$ contient tous les nœuds sauf le nœud racine.

Un motif hiérarchique de co-évolution extrait à partir d'un graphe attribué dynamique et d'une hiérarchie est un ensemble de nœuds, de pas de temps et d'attributs. Ses nœuds respectent la même tendance sur les attributs à chaque pas de temps et ils sont connectés dans le graphe par un chemin de taille maximale définie. Formellement, un motif hiérarchique de co-évolution $M = (V, T, \Omega)$ est une séquence de graphes $G_t[V]$ dont les arêtes sont induites de $\mathcal{G}$ par V et T où $V \subseteq \mathcal{V}$, $T \subseteq \mathcal{T}$ et $\Omega = A \times \{+, -\}$ tel que $A \subseteq (dom(\mathcal{H}))$. Par abus de langage, nous utiliserons indifféremment Ω ou A pour désigner les attributs du motif. Un attribut associé à une tendance est noté $a^s \in \Omega$ tel que $s \in \{+, -\}$ et $a \in A$. On dit que $\overline{s}$ est le symétrique de s (i.e., si $s = +$ alors $\overline{s} = -$ et inversement). Par définition, ce motif M doit respecter une contrainte de co-évolution et une contrainte de diamètre. La première vérifie que la tendance associée à chaque attribut du motif est respectée pour au moins un de ses attributs fils par tous les nœuds à tous les temps. La deuxième vérifie que la longueur du plus court chemin entre chaque paire de nœuds à chaque pas de temps est inférieure à un seuil donné.

coevolution(M) Soit $\delta_{condition}$ la fonction de Kronecker telle que $\delta_{a^s(v,t)} = 1$ si le nœud v au pas de temps t respecte la tendance s pour l'attribut a. Un motif $M = (V, T, \Omega)$ respecte la contrainte de co-évolution si $\forall a^s \in \Omega$

— $\sum_{t \in T} \sum_{v \in V} \delta_{a^s(v,t)} \geq \sum_{t \in T} \sum_{v \in V} \delta_{a^{\overline{s}}(v,t)}$
— $\forall v \in V, \forall t \in T, \exists a' \in leaf(a)$ t.q. $\delta_{a'^s(v,t)} = 1$

diameter(M) Soit un paramètre k défini par l'utilisateur, et $pcc_G(v, w)$ la longueur du plus court chemin entre deux nœuds du graphe, le motif $M = (V, T, \Omega)$ respecte la contrainte de diamètre si $\forall t \in T,\ \max_{v,w \in V} pcc_{G_t[V]}(v, w) \leq k$.

Pour éviter de produire certains motifs, nous utilisons également des contraintes de taille et de volume. Etant donné un motif $M = (V, T, A)$, **sizeMinV(M)** impose $|V| \geq min_V$, **sizeMinT(M)** impose $|T| \geq min_T$, **sizeMinA(M)** impose $|leaf(A)| \geq min_A$. Finalement, si $volume(M) = \sum_{t \in T} \sum_{a \in leaf(A)} \sum_{v \in V} \delta_{a^s(v,t)}$, la contrainte **volumeMin(M)** impose $volume(M) \geq \vartheta$. Notons que ces définitions sont adaptées de celles décrites dans Desmier et al. (2013) pour prendre en compte l'existence des hiérarchies et donc la possibilité qu'un motif contienne des attributs appartenant à plusieurs niveaux.

Par définition, la co-évolution requiert que la tendance soit vraie pour au moins l'un des attributs fils, mais si elle n'est vérifiée que pour une faible proportion des attributs fils, l'information apportée par le motif n'est pas satisfaisante. Il faut donc savoir si le motif apporte une information intéressante et s'il est préférable de spécialiser l'attribut ou de le conserver en l'état. Considérons la Figure 1, soit le motif $M = \langle\{v_1v_2v_3v_4\}\{t_2\}\{a^+\}\rangle$ et deux motifs fils $M_1 = \langle\{v_1v_2v_3v_4\}\{t_2\}\{a_1^+\}\rangle$ et $M_2 = \langle\{v_2v_3v_4\}\{t_2\}\{a_2^+\}\rangle$, le motif M est plus concis tout en apportant quasiment la même information. À l'inverse, soit un motif $M = \langle\{v_1v_2v_3v_4\}\{t_1\}\{a^+\}\rangle$ et un motif fils $M_1 = \langle\{v_1v_2v_3v_4\}\{t_1\}\{a_1^+\}\rangle$ et considérant que $M_2 = \langle\{v_1v_2\}\{t_1\}\{a_2^+\}\rangle$ n'est pas un motif valide par rapport à la contrainte de volume, le motif M_1 est plus précis que son parent M.

Soit une mesure de pureté du motif $M = (V, T, \Omega)$, $purity(M)$ est le nombre de triplets $(v, t, a^s), v \in V, t \in T, a^s \in \Omega$ valides par rapport au nombre de triplets possibles :

$$purity(M) = \frac{volume(M)}{|V| \times |T| \times |leaf(A)|}$$

Pour savoir si l'attribut résume bien l'information sur ses fils dans le motif, introduisons la contrainte **pureMin(M)** qui impose que la pureté du motif soit supérieure à un seuil $\psi \in [0, 1]$ fixé par l'analyste. Il faut également savoir si la spécialisation de l'attribut apporte un gain de pureté tel qu'il compense la perte de généralité. Pour celà, nous définissons la contrainte **gainMin(M)** qui impose que la pureté du motif fils divisée par la pureté maximale de ses motifs parents soit supérieure à un seuil $\gamma > 1$ fixé par l'analyste. Formellement, le gain de pureté apporté par un motif $M = (V, T, \Omega)$ par rapport à ses motifs parents $M_i = (V, T, \Omega_i)$, c'est à dire les motifs tels que $\Omega_i = (\Omega \setminus a^s) \cup parent(a)^s$ avec $a \in \Omega$ se définit ainsi :

$$gain(M) = \frac{purity(M)}{max_{M_i \in parent(M)}(purity(M_i))}$$

Nous pouvons maintenant dire que la contrainte **pureMin(M)** impose que $purity(M) > \psi$ et que la contrainte **gainMin(M)** impose que $gain(M) > \gamma$.

Un motif qui respecte **gainMin(M)** est un motif qui apporte une information plus pure que tous ses motifs parents. S'il ne la respecte pas, ses motifs parents ont donc une pureté équivalente et apportent au moins autant d'information. Notons que si un motif fils n'apporte pas une gain de pureté suffisant, la pureté de ses descendants pourrait être bien supérieure à celle du motif parent (la pureté est toujours égale à 1 pour les attributs feuilles de la hiérarchie). Cependant la spécialisation engendrerait beaucoup de motifs potentiels sans réel apport d'information.

Un motif hiérarchique de co-évolution M tel qu'il est « inclus » dans un motif M' n'apporte aucune information supplémentaire, un motif ne doit être conservé que s'il est maximal. Un motif hiérarchique de co-évolution $M = (V, T, \Omega)$ respecte **maximal(M)**, i.e., s'il n'existe pas de motif hiérarchique de co-évolution $M' = (V', T', \Omega')$ tel que $M \subseteq M'$ au sens où $V \subseteq V'$, $T \subseteq T'$ et $\Omega \subseteq \Omega'$.

Définition du problème : Étant donné un graphe attribué dynamique et une hiérarchie, notre problème consiste à extraire l'ensemble des motifs hiérarchiques de co-évolution tels qu'ils respectent les contraintes de co-évolution et de diamètre et potentiellement une conjonction de contraintes supplémentaires dont les seuils sont choisis par l'analyste.

3 Algorithme

L'algorithme 1 présente notre proposition. L'énumération peut être représentée par un arbre. Chaque nœud de l'arbre contient 2 tri-sets (collections de trois ensembles), P contient les éléments présents dans le motif en construction et C contient les éléments encore à énumérer. Au début de l'algorithme $P = \varnothing$ et $C = \langle\{\mathcal{V}\}\{\mathcal{T}\}\{children(\mathcal{A}ll) \times \{-,+\}\}\rangle$. À chaque étape soit un élément est énuméré (nœud, temps ou attribut) soit un attribut a de P est spécialisé et un attribut de C est énuméré en conservant a non spécialisé. Au début de l'algorithme et afin de pouvoir mieux exploiter les propriétés d'élagage, un nœud est énuméré puis un temps et un attribut. A chaque étape, les éléments de C sont supprimés s'ils ne peuvent pas être ajoutés à P en donnant un motif valide, i.e., s'ils ne peuvent pas respecter les différentes contraintes. Si le nouveau motif P ne respecte pas les contraintes, l'énumération est stoppée. Cependant la majorité des contraintes présentées dans la Section 2 n'exhibe pas de propriété de monotonie et nous avons défini des bornes et des contraintes relaxées pour élaguer l'espace de recherche.

Lorsqu'un élément est énuméré, il est supprimé de C et ajouté à P, si l'élément est un attribut a^s, son symétrique $a^{\overline{s}}$ est également supprimé de C. Lors de l'étape de spécialisation d'un attribut a de P, tous les motifs fils qui apportent un gain de pureté sont énumérés, i.e., $\forall a_i \in children(a)$ t.q. **gainMin(**P_i**)** est respecté, le motif P_i est énuméré et les a_j^s et $a_j^{\overline{s}}$ t.q. $a_j \in children(a)$ avec $j > i$ sont ajoutés à C. Tous les motifs contenant $a \in P.\Omega$ non spécialisé et $b \in C.\Omega$ sont également énumérés. L'étape de spécialisation est présentée dans la Fig. 2, les deux nœuds fils de gauche représentent la spécialisation, tandis que les deux de droite représentent l'énumération avec l'attribut non spécialisé. Lors d'une étape d'énumération sans spécialisation, il ne faut considérer que les deux nœuds fils de droite comme exemple.

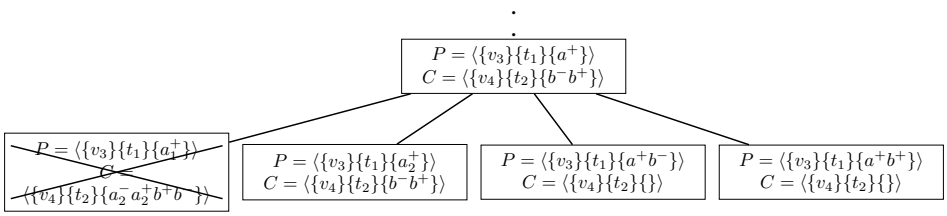


FIG. 2 – *Exemple d'une étape de spécialisation. Les deux nœuds de gauche présentent la spécialisation de l'attribut, les nœuds de droite présente l'énumération d'un autre attribut.*

Soit $C = (V, T, \Omega)$ et $P = (V', T', \Omega')$. À chaque étape C est élagué en fonction de la contrainte de co-évolution, ne sont conservés que les $v \in V$ t.q. **coevolution(**v, T', Ω'**)**, les $t \in T$ t.q. **coevolution(**V', t, Ω'**)** et les $a^s \in \Omega$ t.q. **coevolution(**V', T', a^s**)**. De plus, si **coevolution(**P**)** n'est pas respectée, aucun motif valide ne pourra être énuméré par la suite.

Le diamètre n'est pas monotone, l'ajout d'un nœud à un ensemble de nœuds peut diminuer ou augmenter le diamètre du sous-graphe induit. Il faut donc uniquement vérifier que le graphe induit par l'ensemble de nœuds de P peut respecter la contrainte de diamètre en ajoutant tout ou partie de l'ensemble de nœuds de C. Considérant la Fig. 1 et un diamètre maximal $k = 2$, le motif $\langle\{v_1v_4\}\{t_2\}\{\}\rangle$ ne respecte pas **diameter**, pourtant si le nœud v_3 est ajouté, le motif résultant $\langle\{v_1v_3v_4\}\{t_2\}\{\}\rangle$ respecte la contrainte. À l'inverse si le nœud v_2 est ajouté à ce dernier, le motif $\langle\{v_1v_2v_3v_4\}\{t_2\}\{\}\rangle$ ne respecte plus **diameter** puisque

Algorithme 1 : Enumeration

```
Input : P = ∅,C = (V, T, children(All)),attr
1  begin
2      if ¬C.empty then
3          if coevolution(P ∪ C.V) ∧ lightdiameter(P ∪ C.V) ∧ BSvol(P ∪ C) > ϑ
           ∧ BSvol(P∪C) / (|P.V|×|P.T|×|P.A|) > ψ then
4              hasSon ← false
5              if ¬(attr = ∅) then
6                  ch ← children(attr)
7                  for i in 1..|ch| do
8                      if gainMin(P.V ∪ C.V, P.T ∪ C.T, P.A \ attr ∪ ch[i]) then
9                          Enumeration((P \ attr) ∪ ch[i], C ∪ ch[i + 1..|ch|], ch[i])
10                         hasSon ← true
11             if hasSon then
12                 for i in 1..|C.A| do
13                     Enumeration(P ∪ C.A[i], C \ C.A[1..i], i)
14             else
15                 E ← ElementTypeToEnumerate(P, C)
16                 for i in 1..|C.E| do
17                     if E = A then
18                         attr ← C.E[i]
19                     Enumeration(P ∪ C.E[i], C \ C.E[1..i], attr)
20                 Enumeration(P, C \ C.E,∅)
21     else if
       coevolution(P)∧diameter(P)∧volumeMin(P)∧pureMin(P)∧maximal(P)
       then
22         output (P)
```

$pcc_{G_t[v_1v_2v_3v_4]}(v_1, v_2) = 3$. Nous proposons la contrainte relaxée **lightdiameter**(v, w, V, T) qui impose que $\forall t \in T$, $pcc_{G_t[V]}(v, w) \leq k$. Soit $P = (V', T', \Omega')$ et $C = (V, T, \Omega)$, si $\exists v, w \in V'$ t.q. **lightdiameter**$(v, w, V \cup V', T')$ n'est pas respectée, aucun motif valide ne pourra être énuméré. Ne sont conservés dans C que les $v \in V$ t.q. $\forall w \in V'$, **lightdiameter**$(v, w, V \cup V', T')$ et les $t \in T$ t.q. $\forall v, w \in V'$, **lightdiameter**$(v, w, V \cup V', t)$.

L'espace de recherche est également élagué grâce à la notion de volume : l'énumération est arrêtée si le volume du motif ne peut pas être supérieur à ϑ. Contrairement à Desmier et al. (2013), la fonction de volume n'est pas monotone. Par exemple, si $M \subseteq M'$, $volume(M)$ peut-être supérieur ou inférieur à $volume(M')$: dans la Fig 1, $volume(\langle\{v_1v_2v_3v_4\}\{t_2\}\{X^+\}\rangle) = 9$ alors que $volume(\langle\{v_2v_3v_4\}\{t_2\}\{a^+b^-\}\rangle) = 10$ et $volume(\langle\{v_1v_2v_3v_4\}\{t_2\}\{b^+\}\rangle) = 2$. Soit $BSvol(M)$ une borne supérieure de $volume(M)$: $BSvol(M = (V, T, A)) = \sum_{t\in T} \sum_{a\in leaf(A)} \max\left(\sum_{v\in V} \delta_{a^+(v,t)}, \sum_{v\in V} \delta_{a^-(v,t)}\right)$

Soit $P = (V, T, \Omega)$ et $C = (V', T', \Omega')$, si $BSvol(V \cup V', T \cup T', \Omega \cup \Omega') > \vartheta$ l'énumération continue, sinon plus aucun motif valide ne peut être énuméré (preuve disponible). Finalement, la contrainte de pureté, qui n'est pas monotone puisqu'elle dépend de la fonction $volume$, donne également lieu à élagage. Si $\frac{BSvol(V \cup V', T \cup T', \Omega \cup \Omega')}{|V| \times |T| \times |\Omega|} > \psi$ l'énumération continue, sinon aucun motif valide ne pourra être énuméré.

Lorsque C est vide, $P = (V, T, \Omega)$ est un motif final potentiel. S'il vérifie **sizeMinV(**P**)**, **sizeMinT(**P**)**, **sizeMinA(**P**)**, **volumeMin(**P**)**, **pureMin(**P**)** et **maximal(**P**)** alors le motif est valide. Il est maximal s'il n'existe pas un ensemble de nœuds $v \notin V$ t.q. le motif $(V \cup v, T, \Omega)$ respecte les contraintes, il n'existe pas $t \notin T$ t.q. le motif $(V, T \cup T, \Omega)$ respecte les contraintes et il n'existe pas d'attribut $a^s \notin up(\Omega)$ et $a^s \notin down(\Omega)$ t.q. $(V, T, \Omega \cup a^s)$ respecte les contraintes. L'algorithme extrait donc tous les motifs hiérarchiques de co-évolution qui respectent les contraintes considérant les seuils min_V,min_T, min_A, $\mathcal{V}$, ψ et γ.

4 Expérimentations

Graphe attribué dynamique		$\lvert V\rvert$	$\lvert T\rvert$	$\lvert A\rvert$	Densité	Hiérarchie
DBLP		2145	10	43	1.3×10^{-3}	Fig. 3(a)
US Flights	Septembre 2001	220	30	6	5.7×10^{-2}	Fig. 3(b)
	Katrina	280	8	8	5×10^{-2}	

Nous présentons des résultats expérimentaux obtenus sur des jeux de données réels pour illustrer l'intérêt de notre approche. Toutes les expérimentations ont été réalisées sur une ferme de calcul. Chaque machine est équipée de 2 processeurs à 2,5GHz et 16GB de RAM et utilise la distribution « Scientific Linux ». L'algorithme est implémenté en C++. L'objectif des expérimentations est de répondre aux questions suivantes : Les motifs sont ils pertinents ? Est-il possible de les extraire en un temps acceptable ? Les nouvelles mesures qui ont trait à la pureté permettent-elle d'obtenir des motifs plus faciles à interpréter ? **« DBLP »** est un graphe de co-auteurs créé à partir de la base DBLP [1]. Il y a 43 attributs décrivant le nombre de publications dans une sélection de conférences et journaux dans les domaines des bases de données et de l'extraction de connaissance. Chacun des 2145 nœuds représente un auteur qui a publié au moins 10 fois dans l'une des conférences / journaux entre janvier 1990 et décembre 2012. Ces 22 ans sont divisés en 10 périodes de 5 années consécutives, chaque période possède 3 années en commun avec la période précédente et la période suivante pour garder une consistance dans les données [2]. Chaque arête du graphe relie deux auteurs s'ils ont co-publié au moins une fois ensemble sur la période. Les jeux de données **« US Flights »** ont été créés à partir de la base RITA « On-Time Performance » [3]. Elle contient des informations sur les vols des principaux transporteurs aux États-Unis entre 1990 et 2012. Nous avons dérivé deux jeux où les nœuds représentent des aéroports qui sont connectés par une arête s'il y a eu au moins un vol entre eux sur la période. Les attributs sont le nombre de vols au départ, à l'arrivée, annulés et déviés, les retards au départ et à l'arrivée, le temps au sol avant le décollage et après l'atterrissage. Le jeu « Septembre 2001 » a été créé en agrégeant les données sur 30 périodes qui sont chaque jour du mois de septembre 2001. Le jeu « Katrina » a été créé en agrégeant les données sur 8 périodes qui sont les 3 semaines avant l'ouragan Katrina, les 2 semaines durant l'ouragan et les 3 semaines après.

1. http ://dblp.uni-trier.de/
2. [1990-1994][1992-1996][1994-1998]...[2008-2012]
3. http ://www.transtats.bts.gov

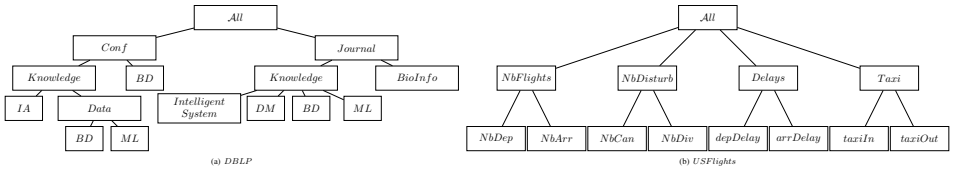
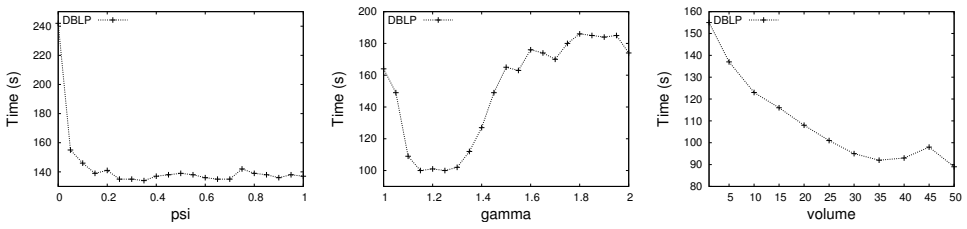


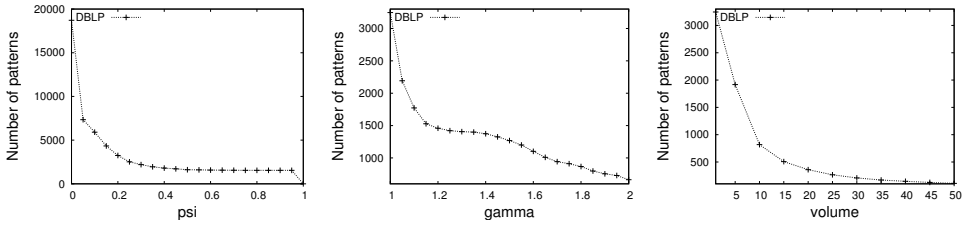
FIG. 3 – *Hiérarchie des jeux de données.*

« DBLP » : Étude d'un réseau de co-auteurs Pour évaluer l'effet des contraintes sur le déroulement de l'algorithme, nous étudions le temps d'exécution et le nombre de motifs extraits en fonction des seuils ψ, γ et ϑ. Considérant ψ, l'augmentation du seuil de pureté a un effet visible sur le nombre de motifs et sur le temps d'exécution. Ils décroissent fortement entre 0 et 0,2, plus de 75% des motifs ont une pureté inférieure à 0,2. Ceci est dû au fait que la majorité des triplets (v, t, a^-) et (v, t, a^+) sont invalides dans le jeu de données car beaucoup d'auteurs ne publient jamais dans certains journaux ou conférences. Considérant le seuil de gain, le nombre de motifs est également très dépendant de γ. Par contre, le comportement en terme de temps d'exécution n'est pas monotone. Ce temps diminue fortement puis ré-augmente. Ceci s'explique par le fait que la hiérarchie possède beaucoup de niveaux. Lorsque γ est bas, les attributs sont spécialisés et il y a de nombreuses énumérations possibles. Avec une valeur moyenne pour γ, les attributs sont conservés à des niveaux moyens de la hiérarchie, ce qui permet de bien élaguer le reste des éléments puisque les auteurs ont des comportements similaires sur ces ensembles de conférences ou journaux. À l'inverse, si γ est élevé, les attributs ne sont que très peu spécialisés et le reste des éléments est peu élagué. Enfin, avec le seuil de volume ϑ, le nombre de motifs est logiquement impacté mais on note que le temps de calcul l'est également bien que la contrainte ne soit pas monotone. La hiérarchie a un effet important sur le résultat de l'extraction. Pour le montrer nous avons créé cinq hiérarchies pour le jeu de données DBLP en enlevant à chaque fois un niveau de profondeur à celle proposée dans la Fig. 3(a). Les résultats de l'expérimentation sont présentés dans la Fig. 5. Si la hiérarchie est trop générale elle implique un grand nombre de motifs et un long temps d'exécution, mais si elle est plus précise le nombre de motifs et le temps d'exécution redeviennent acceptables.

Comme « DBLP » a beaucoup d'attributs à 0, il n'est pas pertinent de mettre un seuil de pureté trop fort. Nos seuils ont été fixés à $\psi = 0,35$, $\gamma = 1,1$ et $\vartheta = 20$, $min_V = 2$, $min_T = 2$, $min_A = 2$. Sur les 5 motifs obtenus, deux groupes se dégagent. Un premier concerne des chercheurs en fouille de données (Jiawei Han, Jian Pei, Ke Wang, Guozhu Dong, Philip S. Yu, Jiong Yang, Ming Li, Charu C. Aggarwal, Christos Faloutsos, Jianyong Wang, Zhi-Hua Zhou). Les motifs contiennent tous les même attributs « Journal - Data Mining » et « IEEE-TKDE ». Le nombre de publications augmente dans les journaux de « Data Mining » et « IEEE-TKDE » entre 1998 et 2008 dans les deux premiers motifs tandis que le nombre de publications augmente dans les journaux de « Data Mining » et diminue dans le journal « IEEE-TKDE » entre 2004 et 2012 dans le troisième motif. Tous ces motifs ont une pureté comprise entre 0,4 et 0,44, ce qui est une pureté relativement importante considérant le jeu de données. Ces motifs reflètent que le journal « IEEE-TKDE », qui est classé en tant que « Journal - Base de Données » dans la hiérarchie, a une forte attractivité en « Fouille de Données ». Un second groupe de motifs implique deux motifs contenant les même conférences « VLDB » et « ICDE » et un auteur commun



(a) Temps d'exécution (s)



(b) Nombre de motifs

FIG. 4 – *Nombre de motifs et temps d'exécution sur le jeu de données DBLP lors de la variation des seuils ψ , γ et ϑ (resp. égaux à $0,6$, 1 et 1 lorsqu'ils sont fixes).*

Philip S. Yu. Le premier motif qui contient 6 auteurs montre une augmentation du nombre de publications dans ces conférences entre 1998 et 2006, le second qui contient 15 auteurs montre une diminution entre 2004 et 2012. Si ce comportement semble logique pour la conférence « VLDB » étant donné la nouvelle politique de « VLDB endowment » (processus de relecture via PVLDB), il est intéressant de noter que le comportement est identique sur la conférence « ICDE ». Ces motifs peuvent montrer qu'il y a eu une évolution dans les collaborations entre les auteurs et que ce changement a pu impliquer une évolution de la politique de publication.

« US Flights » : Étude des vols domestiques aux États-Unis Pour le jeu « Septembre 2001 », le but était d'extraire des motifs traces des évènements qui ont affecté une très grande partie des aéroports. Les paramètres ont été fixés à $min_V = 140$, $\psi = 0,9$ et $\gamma = 1$ puis le seuil de gain a été augmenté à $\gamma = 1,2$ pour voir l'effet de l'accentuation de la contrainte sur la qualité des motifs. Tous les résultats sont obtenus en moins de 5 secondes. Dans la première extraction, 9 motifs sont extraits et 7 dans la seconde. Pour la première extraction, plusieurs motifs représentent une diminution de « NbDep,NbArr » entre le dimanche et le lundi. Dans la seconde, ces même motifs se retrouvent dans un ensemble de motifs où le nombre de vols « NbFlights » est extrait. En particulier, trois motifs de la première extraction ont rapport à une même date et ont 95% d'aéroports en commun ($\langle\{145$ airports$\}\{$2001-09-21$\}\{NbArr^-\}\rangle$, $\langle\{147$ airports$\}\{$2001-09-21$\}\{NbDep^-\}\rangle$ et $\langle\{141$ airports$\}\{$2001-09-21$\}\{NbArr^-NbDep^-\}\rangle$). Dans la seconde extraction, ils se retrouvent dans un seul motif contenant l'attribut parent « NbFlights » et tous les aéroports ($\langle\{151$ airports$\}\{$2001-09-21$\}\{NbFlights^-\}\rangle$). Dans les deux extractions, des motifs traduisent une augmentation du nombre d'annulation les 11 et 12 septembre et une diminution le 14 septembre. Ici les motifs sont identiques car l'attribut « NbCan »

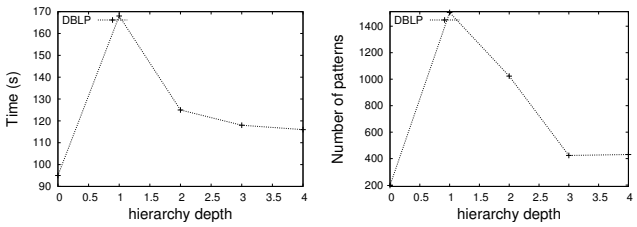


FIG. 5 – *Nombre de motifs et temps d'exécution sur données DBLP avec variation de la profondeur de la hiérarchie ($\psi = 0,2$, $\gamma = 1$ et $\vartheta = 50$).*

apporte beaucoup plus d'information que son parent « NbDisturb » puisqu'il y a peu ou pas d'augmentation du nombre de vols déviés. L'ajout de la hiérarchie et des contraintes de pureté permet donc d'obtenir un ensemble de motifs plus concis et pourtant très purs ($> 0,9$). Pour le jeu de données « Katrina », il y a eu deux extractions, en faisant varier le seuil de volume. Les paramètres sont fixés à $\vartheta = 100$, $\psi = 0,9$ et $\gamma = 1$ puis $\gamma = 1,2$. 24 motifs sont extraits dans la première expérimentation et 16 dans la seconde. Chaque expérimentation a demandé moins de 85 secondes. Les motifs des deux extractions sont similaires et se recoupent, ils présentent des ensembles d'aéroports qui ont une augmentation ou une diminution de leurs vols, de leurs délais et de leurs annulations à différentes périodes de l'ouragan. Dans la première extraction les motifs contiennent principalement les six premiers attributs du jeu de données ainsi que l'attribut parent « Delays », tandis que dans la seconde extraction, ils contiennent les attributs « NbFlights », « Delays » et « NbCan ». Dans les deux cas, les motifs contiennent les même temps, i.e., principalement les 6 premières semaines du jeu de données. Un motif en particulier est intéressant : lors de la première extraction il possède 65 aéroports avec « NbDep et « NbArr » qui diminuent ; dans la deuxième extraction il contient 83 aéroports incluant les 65 premiers avec « NbFlights » qui diminue. Les motifs extraits en augmentant γ contiennent principalement des attributs parents pour les délais et les vols, pourtant les motifs ont tous une pureté supérieure à $0,9$. Ils apportent donc une information plus concise tout en conservant une précision tout à fait acceptable. De plus, l'arrêt à un niveau supérieur de la hiérarchie permet d'augmenter le nombre d'aéroports concernés qui pouvaient ne pas apparaitre dans la première extraction à cause d'une erreur dans les données ou plus simplement parce-qu'ils ne respectaient pas une tendance à un temps du motif alors qu'ils respectaient toutes les autres évolutions.

5 État de l'art

La fouille de graphes dynamiques est très étudiée. Lahiri et Berger-Wolf (2010) recherchent des sous-graphes similaires apparaissant périodiquement. Inokuchi et Washio (2010) proposent l'extraction de sous-séquences de sous-graphes induits fréquents tels qu'un graphe est un sous-graphe d'un autre s'il existe une fonction sur les nœuds, arcs, étiquettes et graphes de la séquence. Prado et al. (2013a) proposent un algorithme de fouille de sous-graphes planaires fréquents à partir d'une base de données de graphes planaires. Ces motifs peuvent être utilisés comme base pour l'extraction de motifs spatio-temporels. Les graphes attribués ont également

été étudiés. Moser et al. (2009) ont proposé une méthode pour trouver des sous-graphes homogènes denses, c'est à dire qui partagent un grand nombre d'attributs. Silva et al. (2012) proposent l'extraction de paires de sous-graphes et d'ensembles d'attributs booléens tels que les attributs sont fortement corrélés avec les sous-graphes. Mougel et al. (2012) recherchent des collections de k-cliques percolées homogènes, c'est à dire des ensembles de cliques qui se chevauchent et partagent un ensemble d'attributs. Prado et al. (2013b) proposent une méthode pour trouver des régularités sur les descripteurs des attributs dans des graphes attribués. Pour cela, ils utilisent les attributs associés aux nœuds ainsi que des propriétés topologiques calculées pour chaque noeud. Récemment des études ont également été faites sur les graphes attribués dynamiques. Boden et al. (2012) proposent d'extraire des clusters dans des graphes attribués puis d'associer les clusters similaires à des temps consécutifs. Jin et al. (2007) étudient les graphes dynamiques dont les nœuds sont associés à un poids. Ils extraient des groupes de nœuds connectés dont le poids suit une évolution similaire croissant ou décroissant sur des temps consécutifs. Desmier et al. (2013) extraient des ensembles de nœuds similaires ayant une même tendance dans le temps sur leurs attributs et une évolution différente du reste du graphe. Aucun de ces travaux n'intègre de connaissance a priori via des hiérarchies.
L'utilisation de connaissances utilisateur dans le processus d'extraction est très étudiée en fouille de données. Cela peut se faire en pré-traitement des données, en processus itératif lors de l'extraction ou encore pendant l'extraction. Nous présentons ici des méthodes utilisant une hiérarchie comme connaissance à priori. Srikant et Agrawal (1996) proposent l'extraction de « séquences étendues ». Ils utilisent une taxonomie sur les objets pour extraire des séquences possédant plusieurs niveaux de hiérarchie. Han et Fu (1999) proposent une méthode d'extraction de règles d'association multi-niveaux, pour laquelle ils utilisent une taxonomie déjà existante. Ils développent une extension de l'algorithme Apriori et procèdent en deux étape, une extraction des motifs d'un même niveau puis un spécialisation en profondeur des objets. Chen et al. (2009) ajoutent une information de structure à un outil OLAP. Ils introduisent différentes dimensions et mesures pour aggréger les graphes par rapport à une hiérarchie sur les dimensions afin d'obtenir des informations. Plantevit et al. (2010) définissent le problème de l'extraction de séquences multi-dimensionnelles et multi-niveaux et proposent un algorithme pour les extraire. Aucun des travaux mentionnés ne porte sur les graphes attribués dynamiques.

6 Conclusion

Nous nous sommes intéressés à l'extraction de motifs hiérarchiques de co-évolution dans des graphes attribués dynamiques, c'est à dire un sous-graphe connexe induit par un ensemble de nœuds de temps et d'attributs appartenant à une hiérarchie et associés à une tendance respectée par les nœuds. Nous avons proposé plusieurs contraintes qui permettent d'obtenir des motifs concis et sans perte d'information et nous avons présenté un algorithme complet exploitant l'ensemble des contraintes. Les résultats des expérimentations sur trois jeux de données réels montrent que l'utilisation de la hiérarchie permet d'obtenir un nombre restreint de motifs. Deux perspectives semblent intéressantes, premièrement, la possibilité d'avoir une hiérarchie à héritage multiple ; deuxièmement créer une hiérarchie sur les temps et les nœuds du graphe.

Remerciements Les auteurs remercient l'ANR pour le financement de ce travail à travers le projet FOSTER (ANR-2010-COSI-012-02), ainsi que le Centre de Calcul du CNRS/IN2P3.

Références

Boden, B., S. Günnemann, et T. Seidl (2012). Tracing clusters in evolving graphs with node attributes. In *CIKM*, pp. 2331–2334.

Chen, C., X. Yan, F. Zhu, J. Han, et P. S. Yu (2009). Graph olap : a multi-dimensional framework for graph data analysis. *KAIS 21*(1), 41–63.

Desmier, E., M. Plantevit, C. Robardet, et J.-F. Boulicaut (2013). Trend mining in dynamic attributed graphs. In *ECML-PKDD*, pp. 654–669.

Han, J. et Y. Fu (1999). Mining multiple-level association rules in large databases. *IEEE TKDE 11*(5), 798–804.

Inokuchi, A. et T. Washio (2010). Mining frequent graph sequence patterns induced by vertices. In *SDM*, pp. 466–477.

Jin, R., S. McCallen, et E. Almaas (2007). Trend Motif : A Graph Mining Approach for Analysis of Dynamic Complex Networks. In *ICDM*, pp. 541–546. IEEE.

Lahiri, M. et T. Berger-Wolf (2010). Periodic subgraph mining in dynamic networks. *KAIS 24*(3), 467–497.

Moser, F., R. Colak, A. Rafiey, et M. Ester (2009). Mining cohesive patterns from graphs with feature vectors. In *SDM*, pp. 593–604.

Mougel, P.-N., C. Rigotti, et O. Gandrillon (2012). Finding collections of k-clique percolated components in attributed graphs. In *PAKDD*, pp. 181–192.

Plantevit, M., A. Laurent, D. Laurent, M. Teisseire, et Y. W. Choong (2010). Mining multidimensional and multilevel sequential patterns. *TKDD 4*(1).

Prado, A., B. Jeudy, É. Fromont, et F. Diot (2013a). Mining spatiotemporal patterns in dynamic plane graphs. *IDA Journal 17*(1), 71–92.

Prado, A., M. Plantevit, C. Robardet, et J.-F. Boulicaut (2013b). Mining graph topological patterns : Finding covariations among vertex descriptors. *IEEE TKDE 25*(9), 2090–2104.

Silva, A., W. M. Jr., et M. J. Zaki (2012). Mining attribute-structure correlated patterns in large attributed graphs. *PVLDB 5*(5), 466–477.

Srikant, R. et R. Agrawal (1996). Mining sequential patterns : Generalizations and performance improvements. In *EDBT*, pp. 3–17.

Summary

Mining dynamic and attributed graphs is a timely challenge, for example in the context of social interactions analysis. It is often possible to associate a hierarchy on the attributes of graphs to formalize prior knowledge. E.g., studying scientific scientific collaboration networks, conferences and journals in which researchers publish can be grouped w.r.t. types/topics. We propose to extend recent constraint-based mining method by exploiting such hierarchies on attributes and their subsumption ability. We define an algorithm that enumerate all multi-level co-evolution patterns, i.e., set of vertices that satisfy a topologic constraint and which have the same evolution on a set of trends and timestamps. Experiments show that hierarchies enable to return more concise collections of patterns without information loss.