Extraction des évolutions récurrentes dans un unique graphe dynamique attribué

Zhi Cheng, Frédéric Flouvat, Nazha Selmaoui-Folcher

PPME - Université de la Nouvelle Calédonie, BP R4, 98851, Nouméa, Nouvelle Calédonie
prénom.nom@univ-nc.nc,
http://pages.univ-nc.nc/~ nom

Résumé. Un grand nombre d'applications nécessitent d'analyser un unique graphe attribué évoluant dans le temps. Cette tâche est particulièrement complexe car la structure du graphe et les attributs associés à chacun de ses nœuds ne sont pas figés. Dans ce travail, nous nous focalisons sur la découverte de motifs récurrents dans un tel graphe. Ces motifs, des séquences de sous-graphes connexes, représentent les évolutions récurrentes de sous-ensembles de nœuds et de leurs attributs. Différentes contraintes ont été définies (e.g. fréquence, volume, connectivité, non redondance, continuité) et un algorithme original a été proposé. Les expérimentations réalisées sur des jeux de données synthétiques et réelles démontrent l'intérêt de l'approche proposée et son passage à l'échelle.

1 Introduction

Les graphes sont de plus en plus utilisés pour représenter des données (ex. spatio-temporelles) et modéliser des phénomènes complexes. Un grand nombre d'algorithmes de fouille de graphes ont été développés (Aggarwal et Wang (2010); Cook et Holder (2006)) et utilisés dans différents domaines d'application tels que la télédétection, les réseaux sociaux, les réseaux biologiques et la bioinformatique (Berlingerio et al. (2011); Prakash et al. (2014); Sanhes et al. (2013)). Récemment, plusieurs algorithmes ont été proposés pour analyser des évolutions de graphes à travers le temps (Ahmed et Karypis (2015); Araujo et al. (2016); Berlingerio et al. (2009); Borgwardt et al. (2006); Desmier et al. (2012); Inokuchi et Washio (2012); Ozaki et Ohkawa (2009); Robardet (2009)). Par exemple, Ahmed et Karypis (2015) ont exploité des co-évolutions relationnelles fréquentes dans un graphe dynamique labélisé, i.e. ensemble de nœuds dont les liens évoluent de façon similaire. Araujo et al. (2016) ont adopté une approche incrémentale d'analyse des tenseurs pour découvrir des communautés périodiques dans un large réseau (graphe non étiqueté). Berlingerio et al. (2009) ont introduit de nouveaux motifs de sous-graphes (absolute-time) pour extraire des règles de graphes d'évolution. Borgwardt et al. (2006) recherchent des sous-graphes dans des graphes dynamiques étiquetés en insérant et supprimant des arêtes dans le temps. Inokuchi et Washio (2012) ont proposé une méthode pour extraire des motifs fréquents et pertinents dans une base de séquences de graphes étiquetés. Ozaki et Ohkawa (2009) ont proposé une méthode pour découvrir des séquences de sous-graphes corrélés dans une séquence de graphes étiquetés. Robardet (2009) développe une

méthode de découverte des évolutions de quasi-cliques qui sont légèrement modifiées à des temps consécutifs. Dans la littérature, peu de travaux s'attaquent au problème de l'extraction dans un graphe dynamique attribué, i.e. un graphe où les attributs, les arêtes et les nœuds évoluant dans le temps. Fouiller ce type de graphe est une tâche complexe. Desmier et al. (2012) étudient l'extraction de co-évolution cohésives dans un graphe dynamique attribué. Ces motifs représentent un ensemble de sommets avec les mêmes valeurs d'attributs et les mêmes voisins pendant un certain lapse de temps (les nœuds et les valeurs d'attributs restent figés). Ces auteurs ont étendu leurs travaux dans Desmier et al. (2013) en intégrant des contraintes sur la topologie et les valeurs d'attributs du graphe. Notre travail se focalise sur la recherche de motifs plus généraux, décrivant des évolutions récurrentes dans un graphe dynamique attribué (section 2). Ces motifs représentent des séquences de sous-graphes vérifiant des contraintes topologiques, des contraintes de fréquence et de non redondance. Nous décrivons une stratégie originale, appelée *RPminer* basée sur des intersections de sous-graphes et une extension progressive des motifs au cours du temps (section 3). Les expérimentations réalisées sur des jeux de données synthétiques et réelles démontrent le passage à l'échelle de cet algorithme incrémental et l'intérêt des motifs extraits (section 4).

2 Notations et définitions

2.1 Graphe dynamique attribué

La base de données en entrée est constituée d'un unique graphe dynamique attribué $\mathcal{G} =< G_{t_1}, G_{t_2}, ..., G_{t_{max}} >$ représentant l'évolution d'un graphe sur un ensemble de temps $\mathcal{T} = \{t_1, t_2, ...t_{max}\}$. L'ensemble des nœuds de $\mathcal{G}$ est noté $\mathcal{V}$. Un nœud de $\mathcal{G}$ est étiqueté par un ensemble d'attributs A (numériques ou catégoriels). Chaque attribut $a \in A$ est associé à un domaine de valeurs $\mathbb{D}_a$. Pour chaque temps $t \in T$, le graphe $\mathcal{G}$ est un graphe attribué non-orienté noté $G_t = (V_t, E_t, \lambda_t)$ où $V_t \subseteq \mathcal{V}$ est l'ensemble des nœuds au temps t , $E_t \subseteq V_t \times V_t$ est l'ensemble des arêtes au temps t, et $\lambda_t : V_t \rightarrow 2^{A\mathbb{D}}$ est la fonction associant à chaque sommet de V_t un ensemble de valeurs $A\mathbb{D} = \bigcup_{a \in A}(a \times \mathbb{D}_a)$. Dans la suite de l'article, nous prendrons $\mathbb{D}_a = \{+, -, 0\}$ afin de simplifier les exemples ($\mathcal{G}$ représente alors un graphe de tendances). La figure 1 présente un exemple de graphe dynamique attribué qui n'est pas nécessairement connexe à un temps donné.
Un graphe $G' = (V', E', \lambda')$ est un sous-graphe attribué d'un graphe $G = (V, E, \lambda)$, noté $G' \sqsubseteq G$, si et seulement si (1) $V' \subseteq V$, (2) $E' \subseteq E$, et (3) $\forall v' \in V' : \lambda'(v') \subseteq \lambda(v')$. Le graphe G' est un sous-graphe attribué connexe de G, noté $G' \sqsubseteq_{conn} G$, si et seulement si $G' \sqsubseteq G$ et, pour tout $u', v' \in V'$, il existe un chemin entre u' et v' dans G'.

2.2 Une nouveau domaine de motifs et ses contraintes

2.2.1 Evolutions récurrentes de nœuds

Soit (V, λ) un sous-ensemble de nœuds attribués de $\mathcal{G}$ avec $V \subseteq \mathcal{V}$ et $\lambda : V \rightarrow 2^{A\mathbb{D}}$. (V, λ) peut être vu comme un graphe attribué sans arêtes. La définition d'un sous-graphe attribué présentée dans la section précédente peut être facilement étendue à un ensemble des nœuds attribués. Nous obtenons ainsi $(V', \lambda') \sqsubseteq (V, \lambda)$ ssi $V' \subseteq V$ et $\forall v' \in V' : \lambda'(v') \subseteq \lambda(v')$. Pour faciliter la lecture des exemples, l'ensemble des nœuds attribués (V, λ) sera noté $(v_1$:

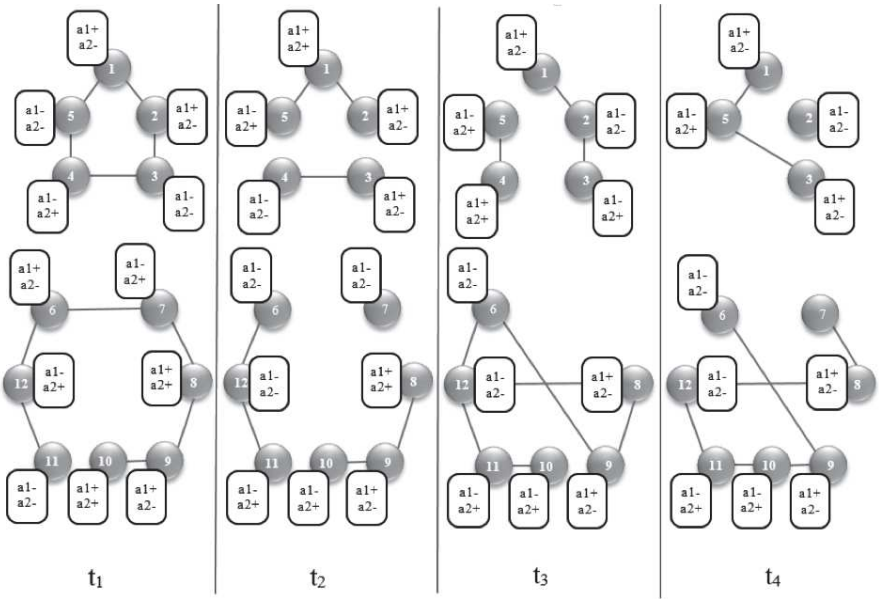

FIG. 1 – *Un exemple de graphe dynamique attribué $\mathcal{G}$*

$\lambda(v_1) \mid v_2 : \lambda(v_2) \mid ...), \forall v_1, v_2... \in V$. Dans la figure 1, $(1 : a_1 + a_2- \mid 2 : a_1 + a_2- \mid 3 : a_1 - a_2- \mid 4 : a_1 - a_2+ \mid 5 : a_1 - a_2-)$ est un ensemble de nœuds attribués de t_1.
L'évolution d'un sous-ensemble de nœuds de $\mathcal{G}$ à un temps $t \in \mathcal{T}$ est une séquence $S =< (V'_1, \lambda'_1), ..., (V'_k, \lambda'_k) >$ telle que $\forall i \in \{1, 2, ..., k\}$, $\exists E'_i \subseteq E_{t+i-1}$, $(V'_i, E'_i, \lambda'_i) \sqsubseteq G_{t+i-1}$. Par exemple, dans la figure 1, $\langle (1 : a_1 + a_2- \mid 2 : a_1 + a_2- \mid 3 : a_1 - a_2- \mid 4 : a_1 - a_2+ \mid 5 : a_1 - a_2-)\ (1 : a_1 + a_2+ \mid 2 : a_1 + a_2- \mid 5 : a_1 - a_2+)\rangle$ est une évolution débutant au temps t_1.
Soit $T_P = \{t_{i_1}, t_{i_2}, ..., t_{i_m}\}$ un ensemble de temps associés à l'évolution $S_P =< (V'_1, \lambda'_1), (V'_2, \lambda'_2), ..., (V'_k, \lambda'_k) >$. Une évolution récurrente d'un sous-ensemble de nœuds de $\mathcal{G}$ à T_P, selon la séquence S_P, est notée $P = (S_P, T_P)$. Dans ce cas, la taille de P est k. Dans la figure 1, $\big(\langle (1 : a_1+ \mid 2 : a_1 + a_2- \mid 5 : a_1-)(1 : a_1+ \mid 2 : a_2-)\rangle, \{t_1, t_2\}\big)$ est un exemple d'évolution récurrente débutant aux temps t_1 et t_2.
Une relation de spécialisation/généralisation peut être définie sur ce domaine de motifs. Soit $P1 = (< (V'_1, \lambda'_1), (V'_2, \lambda'_2), ..., (V'_k, \lambda'_k) >, T_{P1})$ et $P2 = (< (V''_1, \lambda''_1), (V''_2, \lambda''_2), ..., (V''_l, \lambda''_l) >, T_{P2})$, deux motifs représentant des évolutions récurrentes de $\mathcal{G}$. $P1$ est une évolution récurrente plus générale (resp. plus spécifique) que $P2$, noté $P1 \preceq P2$, s'il existe $j \in \{0, ..., l - k\}$, tel que $\forall i \in \{1, ..., k\}$, $(V'_i, \lambda'_i) \sqsubseteq (V''_{i+j}, \lambda''_{i+j})$. Par exemple dans la figure 1, $(\langle 1 : a_1+ \mid 2 : a_1 + a_2- \mid 5 : a_1-)(1 : a_1+ \mid 2 : a_2-)\rangle, \{t_1, t_2\})$ est un évolution récurrente plus spécifique que $\big(\langle (1 : a_1+ \mid 2 : a_1 + a_2-)(1 : a_1+)\rangle, \{t_1, t_2\}\big)$.

2.2.2 Mesures d'intérêt et contraintes

Nous proposons dans cette sous-section plusieurs mesures et contraintes permettant à l'utilisateur de filtrer des motifs d'intérêt en sortie de l'extraction.

Connectivité. Les nœuds d'un graphe représentent souvent des individus/objets, et les arêtes représentent des relations entre ces individus/objets. L'intégration lors de l'extraction d'une contrainte de connexité entre les nœuds permet de désigner des évolutions potentiellement corrélées. $P = (\ <\ (V'_1, \lambda'_1),\ (V'_2, \lambda'_2), ..., (V'_k, \lambda'_k)\ >, T_P\ )$ est une évolution de nœuds connexes dans $\mathcal{G}$ si pour $\forall t \in T_P$, $\forall i \in \{1, 2, ..., k\}$, $\exists E'_i \subseteq E_{t+i-1}$, $(V'_i, E'_i, \lambda'_i) \sqsubseteq_{conn} G_{t+i-1}$. Par exemple, $\big(\langle(1 : a_1+ \mid 2 : a_1 + a_2- \mid 5 : a_1-)(1 : a_1+ \mid 2 : a_2-)\rangle, \{t_1, t_2\}\big)$ est une évolution de nœuds connexes (figure 1).

Non redondance. Il est bien connu que le nombre de motifs générés est souvent très grand et que certains de ces motifs sont redondants. Si deux motifs $P1 = (S_{P1}, T_{P1})$ et $P2 = (S_{P2}, T_{P2})$ sont tels que $P1 \preceq P2$ et $T_{P1} = T_{P2}$, alors il n'est pas nécessaire de conserver $P1$. En effet, la séquence de nœuds attribués de $P1$ se retrouve dans $P2$ et ses éléments apparaissent exactement aux mêmes temps. Cette contrainte se rapproche de la notion de fermé utilisée pour d'autres domaines de motifs (p.ex. itemset, séquence, arbre). Plus formellement, soit Sol l'ensemble des motifs solutions. Soient $P1 = (S_{P1}, T_{P1})$ et $P2 = (S_{P2}, T_{P2})$ deux évolutions récurrentes. Si $P1 \in Sol$ alors $\nexists P2 \in Sol$ tel que $P1 \prec P2$ et $T_{P1} = T_{P2}$. Par exemple, $\big(\langle(1 : a_1+ \mid 2 : a_1 + a_2-)(1 : a_1+ \mid 2 : a_2-)\rangle, \{t_1, t_2\}\big)$ et $\big(\langle(1 : a_1+ \mid 2 : a_1 + a_2- \mid 5 : a_1-)(1 : a_1+ \mid 2 : a_2-)\rangle, \{t_1, t_2\}\big)$ sont deux évolutions redondantes.

Fréquence. La contrainte de fréquence minimale vise à filtrer les motifs apparaissant plus d'un certain nombre de fois. Elle est couramment utilisée lorsque la base de données est une collection de transactions. Toutefois, sa définition dans le cadre d'un graphe unique est généralement plus complexe (Fiedler et Borgelt, 2007; Bringmann et Nijssen, 2008), principalement à cause de la présence d'occurrences entrelacées. De par la nature des motifs, elle reste simple à calculer dans notre cas. La fréquence d'un motif est tout simplement le nombre de temps à partir desquels débute l'évolution étudiée. Elle représente le nombre de récurrences de l'évolution. Soit $P = (S_P, T_P)$ un motif. La fréquence de P est $sup(P) = |T_P|$. Ainsi, P est une évolution récurrente fréquente (encore appelée évolution récurrente) ssi $sup(P) \geq minsup$, où $minsup$ est un seuil défini par l'utilisateur. Par exemple, dans la figure 1, la fréquence de $\big(\langle(6 : a_2- \mid 11 : a_1- \mid 12 : a_1-)(11 : a_1 - a_2+ \mid 12 : a_1 - a_2-)\rangle, \{t_1, t_2, t_3\}\big)$ est 3, car cette évolution commence à t_1, t_2 et t_3.

Volume. Le volume est une autre mesure correspondante au nombre de nœuds du sous-graphe considéré. Elle représente par exemple la taille d'une communauté dans un réseau social (en supposant que les nœuds soient les individus et les arêtes les liens d'amitié). Soit $vol(P) = min_{\forall i \in \{1...k\}}(|V'_i|)$, le volume du motif $P = (\ <\ (V'_1, \lambda'_1), ...,\ (V'_k, \lambda'_k)\ >, T_P\ )$. P est une évolution récurrente suffisamment volumineuse ssi $vol(P) \geq minvol$, où $minvol$ est un seuil défini par l'utilisateur. Par exemple, $\big(\langle(1 : a_1+ \mid 2 : a_1 + a_2- \mid 5 : a_1-)(1 : a_1+ \mid 2 : a_2-)\rangle, \{t_1, t_2\}\big)$ a un volume de 2.

Continuité. Par défaut, une évolution peut représenter des nœuds très différents à chaque étape. Autrement dit, si $P = (\ <\ (V'_1, \lambda'_1), ...,\ (V'_k, \lambda'_k)\ >, T_P\ )$, il est possible d'avoir $\bigcap_{\forall i \in 1...k} V'_i = \emptyset$. L'interprétation de telles évolutions peut être difficile par les utilisateurs car il n'y a pas a priori de lien direct entre les individus/objets observés (les nœuds). Nous proposons donc une nouvelle contrainte visant à cibler les motifs décrivant des évolutions autour d'un noyau commun d'individus. Ainsi, il est possible de suivre l'évolution dans le

temps d'un certain nombre de nœuds, tout en considérant aussi les nœuds voisins (directement ou indirectement). Soit $com(P) = |\bigcap_{\forall i \in 1...k} V'_i|$, le nombre de nœuds apparaissant à toutes les étapes de $P = (< (V'_1, \lambda'_1), ..., (V'_k, \lambda'_k) >, T_P)$. P est une évolution récurrente continue dans le temps ssi $com(P) \geq mincom$, où $mincom$ est un seuil défini par l'utilisateur. Par exemple, le motif $\big(\langle (1 : a_1+ \mid 2 : a_1 + a_2- \mid 5 : a_1-)(1 : a_1+ \mid 2 : a_2-)\rangle, \{t_1, t_2\}\big)$ a deux nœud communs à t_1 et à t_2, i.e. $comp(P) = 2$.

2.2.3 Problème étudié

Etant donné un graphe dynamique attribué $\mathcal{G}$, le problème est d'énumérer l'ensemble des évolutions récurrentes dans $\mathcal{G}$, noté Sol, tel que $\forall P \in Sol$, (1) les nœuds de P sont connectés à chaque temps, (2) P est non redondant dans Sol, (3) P est fréquent (i.e. $sup(P) \geq minsup$), (4) P est suffisamment volumineux (i.e. $vol(P) \geq minvol$) et (5) P est centré sur un noyau suffisamment grand (i.e. $com(P) \geq mincom$), où $minsup$, $minvol$ et $mincom$ sont des seuils définis par l'utilisateur.

3 L'extraction des évolutions sous-contraintes

Dans cette section, nous présentons les propriétés et les stratégies développées pour extraire les évolutions récurrentes satisfaisant les contraintes présentées dans la section précédente. Nous introduisons également l'algorithme permettant d'extraire efficacement ces motifs. Contrairement à un certain nombre d'algorithmes d'extraction de motifs, notre approche ne s'appuie pas sur une stratégie générer-tester. Elle ne suit pas un parcours de l'espace de recherche en largeur ou en profondeur. Elle ne fait pas non plus un parcours basé sur des projections successives des données (comme dans *PrefixSpan*). Elle utilise des intersections successives des composantes connexes entre chaque temps. Au fur et à mesure de ces intersections et du parcours des temps, les motifs sont progressivement étendus. On obtient ainsi à chaque itération (à chaque temps) un ensemble de solutions de taille différente. L'avantage de cette approche est d'éviter la génération d'un grand nombre de motifs ne vérifiant pas les contraintes et de traiter de manière incrémentale le graphe dynamique attribué, i.e. la séquence de graphes. Dans la sous-section suivante, nous allons introduire la notion d'intersection entre des graphes attribués sur laquelle repose notre stratégie d'énumération.

3.1 L'intersection de graphes attribués

Lien entre intersection et fréquence Soient $i, j \in \mathcal{T}$. L'intersection entre deux graphes attribués $G_i = (V_i, E_i, \lambda_i)$ et $G_j = (V_j, E_j, \lambda_j)$, notée $G_i \sqcap G_j$, est un graphe attribué $G' = (V', E', \lambda')$ tel que $V' = V_i \cap V_j$, $E' = E_i \cap E_j$, $\forall v \in V$, $\lambda'(v) = \lambda_i(v) \cap \lambda_j(v)$. G' est un graphe constitué des nœuds, des arêtes et des valeurs d'attributs communs aux graphes G_i et G_j. On remarque que tout sous-graphe de G' apparaît aux temps i et j. Ils apparaissent donc au moins deux fois dans $\mathcal{G}$. La figure 2 montre que le sous-graphe c apparaît au moins 2 fois dans les données (à t_1 et t_3).
Cette définition peut être généralisée à l'intersection de k graphes, avec $k \in \{2, 3, ...|\mathcal{T}|\}$. Soit $T^k \subseteq \mathcal{T}$ un sous-ensemble de temps tel que $|T^k| = k$. L'intersection des graphes de $\mathcal{G}$ sur les k temps de T^k, notée $\sqcap_{i \in T^k} G_i$, est un graphe $G = (V, E, \lambda)$ avec $V = \cap_{i \in T^k} V_i$, $E = \cap_{i \in T^k} E_i$,

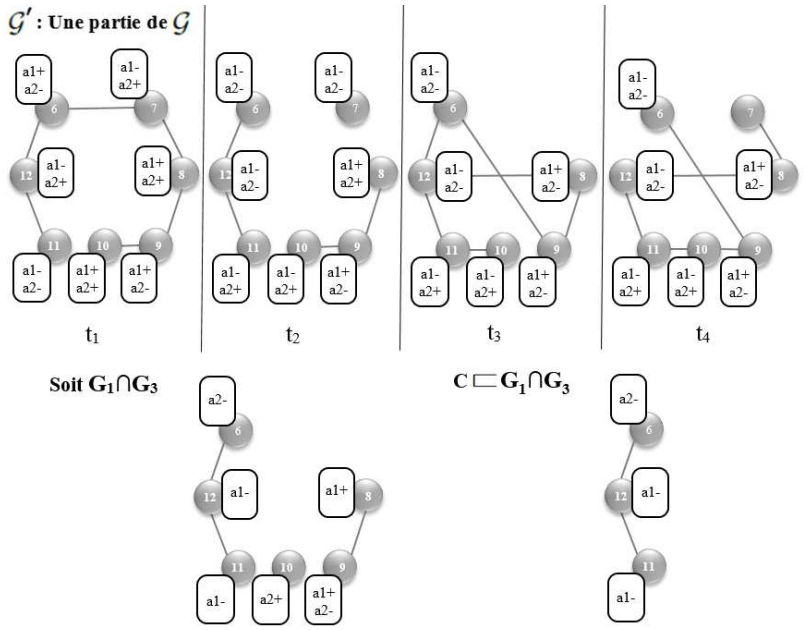

FIG. 2 – *Exemple d'intersection de graphes*

$\forall v \in V, \lambda(v) = \bigcap_{i \in T^k} \lambda_i(v)$. La fréquence minimale d'apparition dans $\mathcal{G}$ de tout sous-ensemble de nœuds attribués de $\sqcap_{i \in T^k} G_i$ est k. Ainsi, tout motif construit à partir de l'intersection d'un nombre $minsup$ de graphes de $\mathcal{G}$ respectera la contrainte de fréquence minimale.

Lien entre intersection et non redondance L'intersection de graphes permet aussi d'avoir d'autres propriétés. Etudions plus particulièrement les composantes connexes issues de l'intersection de plusieurs graphes. Nous noterons $\mathbb{C}_{i \sqcap j}$ l'ensemble des composantes connexes du graphe obtenu après intersection des graphes de $\mathcal{G}$ aux temps i et j, i.e. $G_i \sqcap G_j$. Plus formellement, $\mathbb{C}_{i \sqcap j} = \{(V, E, \lambda) \mid (V, E, \lambda) \sqsubseteq_{conn} G_i \sqcap G_j$ et $\nexists (V', E', \lambda'), (V, E, \lambda) \sqsubset (V', E', \lambda')$ tel que $(V', E', \lambda') \sqsubseteq_{conn} G_i \sqcap G_j\}$.
Soient deux composantes connexes c et c' obtenues après intersection des graphes aux temps $\{i, j\}$ et $\{k, l\}$, i.e. $c \in \mathbb{C}_{i \sqcap j}$ et $c' \in \mathbb{C}_{k \sqcap l}$, $\forall i, j, k, l \in \mathcal{T}$. Soit $T_c = \{t \in \mathcal{T} \mid c \sqsubseteq G_t\}$ (resp. $T_{c'}$) l'ensemble des temps de $\mathcal{T}$ où la composante connexe c (resp. c') apparaît. Il est impossible d'avoir $c \sqsubset c'$ (ou l'inverse) et $\boldsymbol{T_c = T_{c'}}$. En effet, cela impliquerait que c' apparaitrait aux temps $\{k, l\}$ mais aussi $\{i, j\}$. On aurait donc $c' \sqsubseteq G_i \sqcap G_j$, ce qui n'est pas possible car c est une composante connexe de $G_i \sqcap G_j$ (elle est donc maximale). Dans la figure 2, la composante connexe $c_1 = (6 : a_2- \mid 11 : a_1- \mid 12 : a_1-)$ est dans G_1, G_2 et G_3. Elle apparaît donc dans $G_1 \sqcap G_2$ et $G_1 \sqcap G_3$. Il n'existe pas de sur-ensemble de nœuds attribués l'incluant et apparaissant aux même temps. A noter que le sous-ensemble $c_2 = (11 : a_1- \mid 12 : a_1-)$ est obtenu en faisant $G_1 \sqcap G_4$, mais il apparaît à des temps différents (t_1, t_2, t_3 et t_4). Pour conclure, si $c = (V, E, \lambda)$, alors le motif $(< (V, \lambda) >, T_c)$ vérifie la contrainte de connexité

(c est une composante connexe), mais aussi la contrainte de non redondance (dans l'ensemble des solutions de taille 1). Autrement dit, ce motif sera soit un motif solution, soit une partie d'un motif solution. Cette propriété peut être généralisée à tous les ensembles $T, T' \subseteq \mathcal{T}$. Notons $\mathbb{C}_{\sqcap T}$ (resp. $\mathbb{C}_{\sqcap T'}$), l'ensemble des composantes connexes obtenues après intersection des graphes aux temps T (resp. T'), i.e $\underset{i \in T}{\sqcap} G_i$. Si $c \in \mathbb{C}_{\sqcap T}$, alors $\nexists c' \in \mathbb{C}_{\sqcap \mathbb{T}'}$ tel que $c \sqsubset c'$ et $T_c = T_{c'}$. Les motifs de taille 1 associés à ces composantes connexes vérifient les contraintes de connectivité et de non redondance. La réciproque est vraie. Tout motif solution de taille 1, ou tout sous-motif de taille 1 d'un motif solution, peut être dérivé des composantes connexes obtenues après intersection entre des graphes de $\mathcal{G}$. Ces intersections permettent donc d'obtenir les "briques de base" servant à construire l'ensemble des motifs solutions. L'avantage de ces intersections est d'éviter de faire un grand nombre de tests d'inclusion lors de l'extraction (pour vérifier les contraintes de support et de non redondance). Le nombre d'intersections est $2^{|\mathcal{T}|}$. Il ne dépend que du nombre de temps dans $\mathcal{G}$ alors que le nombre de tests d'inclusion dépend du nombre de motifs générés qui est beaucoup plus important.

3.2 La construction des motifs de taille 1

Les motifs de taille 1, i.e. du type $(< (V, \lambda) >, T)$, construits précédemment à partir des composantes connexes extraites des intersections de $\mathcal{G}$, vérifient directement les contraintes de fréquence, de connectivité et de non redondance. Pour que ces motifs soient des solutions, il suffit donc qu'ils vérifient les contraintes de volume et de continuité. Or, ces contraintes sont simples et peu coûteuses à calculer car elles s'appuient uniquement sur la structure du motif. L'ensemble des motifs de taille 1 vérifiant toutes les contraintes peut donc être défini de la manière suivante : $\{(< (V, \lambda) >, T) \mid T \subseteq \mathcal{T}, |T| \geq minsup, |V| \geq minvol$, et $\exists c = (V, E, \lambda)$ tel que $c \in \mathbb{C}_{\sqcap T}\}$. Un prétraitement des graphes avant intersections permet de réduire les tests de connectivités. Les intersections ne sont pas alors réalisées sur les graphes initiaux mais sur leurs composants connexes. Pour résumer, les composantes connexes sont d'abord identifiées dans les graphes de $\mathcal{G}$, et l'extraction des motifs est ensuite effectuée sur les intersections de ces composantes connexes.

3.3 L'extension des motifs

Les motifs de taille 1 extraits dans les intersections peuvent ensuite être combinés, en fonction des temps où ils apparaissent, afin de générer les autres motifs. Cette extension peut se faire de manière incrémentale en traitant les temps les uns après les autres.
La figure 3 illustre cette construction incrémentale à partir des temps t_1 et t_2. Elle représente l'extension en parallèle des motifs apparaissant à t_1 et t_2 (donc de deux de leurs occurrences donc). A noter que la contrainte de fréquence étant directement liée au nombre de temps "intersectés", nous pouvons en déduire que la fréquence minimale dans cet exemple est 2. $\mathbb{C}_i$ et $\mathbb{C}_j$ sont les ensembles de composantes connexes de G_i et G_j. Les rectangles représentent les motifs de taille 1 obtenus après intersection de $\mathbb{C}_i$ et $\mathbb{C}_j$, et extraction des composantes connexes. Supposons par exemple qu'il existe un motif solution $P = (< (V'_1, \lambda'_1), ...(V'_n, \lambda'_n) >, \{t_1, t_2, t_3\})$. Les traitements réalisés pour $\{t_1, t_2\}$ permettent d'obtenir deux occurrences de la séquence $< (V'_1, \lambda'_1) >$ (celle en t_1 et t_2). L'occurrence au temps t_1 ne peut être étendue que par un ensemble de nœuds attribués de t_2 (aucun "gap" de temps n'étant autorisé), à condition

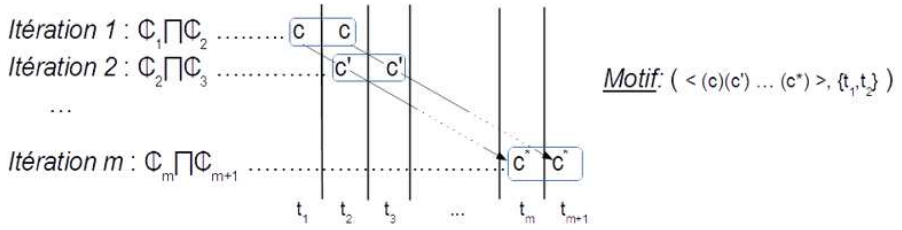

FIG. 3 – *Extension en parallèle des motifs à partir de* $\{t_1, t_2\}$

que ces deux ensembles aient un nombre suffisamment important de nœuds en communs (contrainte de continuité). De même, celle au temps t_2 ne peut être étendue que par les motifs de taille 1 trouvés à t_3. En étudiant ainsi $\{t_2, t_3\}$, on étend $< (V'_1, \lambda'_1) >$ avec (V'_2, λ'_2), et on obtient $< (V'_1, \lambda'_1), (V'_2, \lambda'_2) >$. Ce processus continue jusqu'à ce que l'on ne puisse plus étendre les séquences étudiées. On obtient alors un motif solution (non redondant). Au delà du motif P, cette succession d'extension permet de générer de façon incrémentale tous les motifs commençant à t_1, tous ceux commençant à t_2, etc. En effet, si un motif de taille 1 obtenu à partir de $\{t_2, t_3\}$ ne peut être utilisé pour étendre un motif construit "au temps précédent", il constitue le point de départ pour un nouveau motif.
Avec cette approche, le motif P sera généré et étendu 4 fois (à partir de $\{t_1, t_2\}$, de $\{t_1, t_3\}$, de $\{t_2, t_3\}$, et de $\{t_1, t_2, t_3\}$). A chaque génération, les temps associés au motif sont mis à jour. Bien que l'étude de la combinaison $\{t_1, t_2, t_3\}$ n'apporte pas d'informations par rapport à P, elle permet de découvrir ou d'étendre d'autres motifs. Toutes ces combinaisons d'intersections sont donc nécessaires, d'où l'importance du pré-traitement décrit précédemment pour limiter le coût de cette opération.

3.4 L'algorithme RPMiner

L'algorithme 1 présente en détail l'approche proposée. La ligne 1 correspond à l'extraction des composantes connexes de chaque graphe. Les lignes 3-7 construisent les motifs de taille 1 dont la fréquence est supérieure au seuil minimum et dont au moins une occurence commence au temps t_1. Pour cela, l'algorithme calcule toutes les combinaisons de temps contenant t_1 (T_1^k, ligne 4), puis génère des motifs de taille 1 en faisant l'intersection des composantes connexes apparaissant à ces temps (ligne 5, méthode *ExtractIntersect*). Les autres temps sont ensuite traités les uns après les autres. Pour chaque temps t_i, **RPMiner** construit de nouveau toutes les combinaisons de temps contenant t_i (T_i^k, ligne 11), et extrait des motifs P_i de taille 1 (ligne 12). Ensuite chaque motif P généré à l'itération précédente (ligne 13-14) est étendu. Si P' résultant de l'extension de P avec P_i vérifie la contrainte de continuité, on l'ajoute dans les motifs générés au temps t_i (ligne 15-16). Sinon, on retient le motif P comme solution généré à l'étape précédente et on sauvegarde P_i pour une future extension. A la fin (ligne 26), on regroupe les solutions construites à chaque temps, en prenant soin de mettre à jour les temps des motifs identiques, mais générés à partir de temps différents.

Algorithm 1 *RPMiner* : Extraction des évolutions récurrentes

Require: $\mathcal{G}$: un graphe dynamique attribué , $minsup$: fréquence minimale, $minvol$: volume minimum, $mincom$: nombre minimum de nœuds en commun dans le temps
Ensure: Sol : l'ensemble des évolutions vérifiant les contraintes

$\mathbb{C} = \{\mathbb{C}_i$ ensemble des composantes connexes de $G_i \mid \forall c \in \mathbb{C}_i, c = (V, E, \lambda), |V| \geq minvol\}$
$Cand_i = \emptyset, \forall i \in \{1, 2, ..., |\mathcal{T}|\}$
for $k = minsup$ à $|\mathcal{T}|$ **do**
  for chaque $T_1^k \subseteq \mathcal{T}$ tel que $\|T_1^k\| = k$ et $t_1 \in T_1^k$ **do**
    $Cand_1 = Cand_1 \cup \{P_1 \in ExtractIntersect(\mathbb{C}, T_1^k) \mid vol(P) \geq minvol\}$
  end for
end for
$Sol_i = \emptyset, \forall i \in \{1, 2, ..., |\mathcal{T}|\}$
for $i = 2$ à $|\mathcal{T}|$ **do**
  for $k = minsup$ à $|\mathcal{T}|$ **do**
    for chaque $T_i^k \subseteq \mathcal{T}$ tel que $\|T_i^k\| = k$ et $t_i \in T_i^k$ **do**
      for chaque $P_i \in ExtractIntersect(\mathbb{C}, T_i^k)$ tel que $vol(P) \geq minvol$ **do**
        for chaque $P = (S, T_P)$ tel que $P \in Cand_{i-1}$ et $T_P = T_i^k$ **do**
          $P' = ExtendWith(P, P_i)$
          if $com(P') \geq mincom$ **then**
            $Cand_i = Cand_i \cup \{P'\}$
          else
            $Sol_{i-1} = Sol_{i-1} \cup \{P\}$
            $Cand_i = Cand_i \cup \{P_i\}$
          end if
        end for
      end for
    end for
  end for
end for
$Sol = MergeUpdate(\bigcup_{\forall i \in \mathcal{T}} Sol_i)$

4 Résultats expérimentaux

L'algorithme a été implémenté en $C++$ sur un PC avec un processeur IntelCore 3.5GHz et 8 Go de mémoire. Nous avons utilisé pour nos tests 2 jeux de données réelles et 28 jeux de données synthétiques.

Jeux de données synthétiques. Des séquences de graphes ont été générées aléatoirement suivant une distribution uniforme. Pour étudier les performances de notre approche, nous avons fait varier différents paramètres tels que le nombre de nœuds, le nombre d'attributs, le nombre d'arêtes et la taille de la séquence.

DBLP. Ce jeu de données utilisé dans Desmier et al. (2012) représente les auteurs publiant des articles répertoriés dans DBLP (>10 publications) et leurs co-publications entre 1990 et 2010. Il est composé de 2,723 nœuds par date (auteurs), 10,737 arêtes en moyenne, 43 attributs (conférences/revues) et 9 dates ([1990-1994], [1992-1996]...[2006-2010]).

Trafic aérien. Ce jeu de données utilisé dans Kaytoue et al. (2014) représente le trafic aérien aux USA pendant la période cyclonique de 01/08/2005 à 25/09/2005. Il est composé de 280 nœuds par date (aéroports), 1206 arêtes en moyenne (liaisons aériennes), 8 attributs (e.g. nombre de départs/arrivées, nombre de vols annulés), et 8 dates (dates regroupées par semaine).

Résultats quantitatifs. Fig. 4 (i) présente le temps d'exécution et le nombre de solutions pour 12 jeux de données synthétiques avec un nombre croissant de nœuds et d'arêtes (le

nombre d'attributs est fixé à 30 et le nombre de dates à 6). Comme le montre ces résultats, notre approche reste relativement extensible pour des graphes larges (20,000 nœuds et 80,000 arêtes par date) et des seuils bas. Fig. 4 (ii) montre l'impact du nombre de temps sur notre algorithme. Cet impact est important mais les performances de notre approche incrémentale restent comparable à celles des méthodes proposées dans Desmier et al. (2012, 2013), alors que notre algorithme extrait des motifs plus généraux et plus complexes. Fig. 4 (iii) présente les résultats de performances avec un nombre d'attributs différents. Le temps d'exécution reste quasiment inchangé quand le nombre d'attributs augmente. Fig. 4 (iv) et (v) montrent les performances sur les données DBLP en fonction de différents seuils de fréquence et volume. **RPMiner** est toujours efficace sur ces données réelles même pour des seuils bas. L'impact du seuil de volume est moins important que celui avec le seuil de support car les volumes des composantes connexes sont larges (nombreuses co-publications). Les temps d'exécution pour le trafic aérien (jeu de données plus petit) ne sont pas présentés ici à cause de la limitation d'espace mais ils sont très petits (120 sec. dans le pire des cas).

Interprétation qualitative Nous avons choisi quelques motifs extraits des deux jeux de données réelles pour les interpréter ($minvol = 2$, $minsup = 2$ et $mincom = 1$). Un exemple de motif extrait dans les données DBLP est $(\langle\langle NingZhong : TKDE+, PAKDD-, PKDD- \mid SetsuoOhsuga : PAKDD-, PKDD-)(NingZhong : DMKD+, ICDM-, PAKDD-, JIntellInfSys-)(NingZhong : IEEEIntSys+, PAKDD-, KDD+)\rangle, \{[98-02], [00,04]\}\big)$. Il met en évidence une séquence de taille 3 de publications de 2 co-auteurs. Elle représente une évolution sur une période de 8 ans. Cette séquence se répète deux fois de 1996 à 2004 (i.e. [96-00], [98-02] et [00-04]) et de 2000 à 2006 (i.e.[98-02], [00-04] et [02-06]). Ce motif montre une diminution des publications dans PAKDD et PKDD, avec en parallèle une augmentation des publications dans certaines revues. A noter que cette séquence ne pourrait pas être extraite par l'approche de Desmier et al. (2012, 2013) car les nœuds/auteurs changent tout comme les attributs et leur valeurs.

Les motifs extraits de la base de données du trafic aérien ont mis en évidence des évolutions récurrentes du trafic lors de l'occurrence d'un cyclone. Par exemple, le motif $(\langle\langle Bangor : DelayDeparture+ \mid Boston : DelayArrival+ \mid NewportNews : DelayDeparture+)\ (Augusta : Cancelled- \mid Bangor : Cancelled- \mid Boston : Cancelled-Diverted-)\rangle, \{01/08, 08/08, 29/08, 05/09\})$ montre l'impact des cyclones sur les retards, les annulations et les vols détournés. Les retards augmentent d'abord dans les aéroports "connectés" pendant une semaine. Puis, les annulations et les vols détournés diminuent la semaine suivante. On remarque que tous les aéroports concernés se situent sur la côte Est. Ce motif s'est reproduit 4 fois et correspond aux cyclones survenus pendant cette période. Pour la clarté de la présentation, nous présentons juste un petit nombre d'aéroports impactés par ce motif. En fait, ce motif contient plus de vingt aéroports aux USA.

5 Conclusion

En conclusion, nous avons étudié un problème d'exploration de motifs dans un graphe dynamique attribué. Nous avons proposé un nouveau domaine de motifs et plusieurs contraintes pour extraire des évolutions récurrentes dans de tel graphe. Un algorithme adoptant une stratégie originale a été développé et mis en œuvre. Nous avons conduit des expérimentations sur des jeux de données réelles et artificielles pour mettre en évidence le passage à l'échelle et l'intérêt

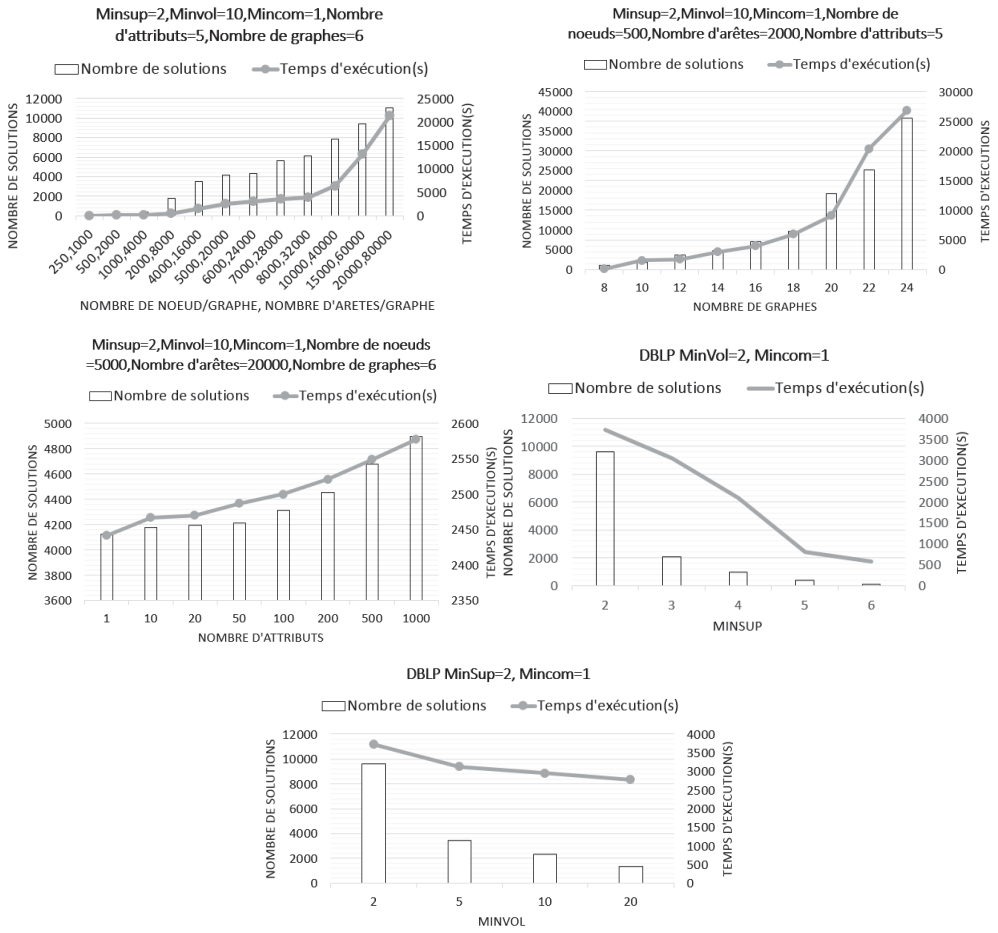

FIG. 4 – *Résultats de performance : (i)Nombre de nœuds et arêtes/graphe, (ii)Nombre de graphes, (iii)Nombre d'attributs, (iv)Minsup(DBLP), (v)Minvol(DBLP)*

des motifs extraits. Une des perspectives est d'appliquer **RPMiner** à une problématique réelle en intégrant d'autres contraintes relatives au domaine d'application. La deuxième perspective est d'utiliser une autre stratégie de parcours pour améliorer la performance de **RPMiner**. Enfin nous envisageons de proposer un post-traitement pour grouper des motifs similaires.

Références

Aggarwal, C. C. et H. Wang (Eds.) (2010). *Managing and Mining Graph Data*, Volume 40. Springer.

Ahmed, R. et G. Karypis (2015). Algorithms for mining the coevolving relational motifs in dynamic networks. *ACM (TKDD) 10*(1), 4.

Araujo, M., S. Günnemann, S. Papadimitriou, C. Faloutsos, P. Basu, A. Swami, E. E. Papalexakis, et D. Koutra (2016). Discovery of "comet" communities in temporal and labeled graphs com^2. *KaIS 46*(3), 657–677.

Berlingerio, M., F. Bonchi, B. Bringmann, et A. Gionis (2009). Mining graph evolution rules. In *ECML-PKDD2009*, pp. 115–130. Springer.

Berlingerio, M., M. Coscia, F. Giannotti, A. Monreale, et D. Pedreschi (2011). Foundations of multidimensional network analysis. In *ASONAM'11*, pp. 485–489.

Borgwardt, K. M., H. Kriegel, et P. Wackersreuther (2006). Pattern mining in frequent dynamic subgraphs. In *ICDM'06*, pp. 818–822.

Bringmann, B. et S. Nijssen (2008). What is frequent in a single graph ? In *PAKDD'08*, pp. 858–863.

Cook, D. J. et L. B. Holder (2006). *Mining graph data*. John Wiley & Sons.

Desmier, E., M. Plantevit, C. Robardet, et J.-F. Boulicaut (2012). Cohesive co-evolution patterns in dynamic attributed graphs. In *DS'12*, pp. 110–124.

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

Fiedler, M. et C. Borgelt (2007). Subgraph support in a single large graph. In *Workshops Proceedings of the 7th IEEE (ICDM 2007)*, pp. 399–404.

Inokuchi, A. et T. Washio (2012). Frissminer : Mining frequent graph sequence patterns induced by vertices. *IEICE Transactions 95-D*(6), 1590–1602.

Kaytoue, M., Y. Pitarch, M. Plantevit, et C. Robardet (2014). Triggering patterns of topology changes in dynamic graphs. In *ASONAM'14*, pp. 158–165.

Ozaki, T. et T. Ohkawa (2009). Discovery of correlated sequential subgraphs from a sequence of graphs. In *ADMA'09*, pp. 265–276.

Prakash, B. A., J. Vreeken, et C. Faloutsos (2014). Efficiently spotting the starting points of an epidemic in a large graph. *KaIS 38*(1), 35–59.

Robardet, C. (2009). Constraint-based pattern mining in dynamic graphs. In *IEEE ICDM*, pp. 950–955.

Sanhes, J., F. Flouvat, C. Pasquier, N. Selmaoui-Folcher, et J. Boulicaut (2013). Weighted path as a condensed pattern in a single attributed DAG. In *IJCAI'13*, pp. 1642–1648.

Summary

A great number of applications require to analyze a single attributed graph that changes over time. This task is particularly complex because both graph structure and attributes associated with each node can change. In the present work, we focus on the discovery of recurrent patterns in such a graph. These patterns are sequences of subgraphs which represent recurring evolutions of subsets of nodes w.r.t. their attributes. Various constraints have been defined (frequency, volume, connectivity, non-redundancy and temporal continuity) and an original algorithm has been developed. Experiments performed on synthetic and real-world datasets have demonstrated the interest of our approach and its scalability.