Extraction hiérarchique de fenêtres de temps basée sur la structure communautaire

Thomas Aynaud et Jean-Loup Guillaume

LIP6 – CNRS – *Université Pierre et Marie Curie*
4 place Jussieu
75005 Paris, France

RÉSUMÉ. *Dans cet article nous décrivons une méthode de décomposition du temps en fenêtres de temps dans un graphe dynamique. Une particularité de la méthode est que le résultat est un regroupement hiérarchique : les fenêtres de temps sont elles-mêmes susceptibles d'en contenir. En outre, les fenêtres n'ont pas besoin d'être contiguës ce qui permet par exemple de détecter une structure se répétant. De plus, chaque fenêtre est associée à une décomposition en communautés représentant la structure topologique du réseau durant cette fenêtre. Nous appliquons ensuite cette méthode à trois graphes de terrain dynamiques ayant des caractéristiques différentes pour montrer que les fenêtres identifiées correspondent bien à des phénomènes observables.*

ABSTRACT. *In this paper, we describe a way to cluster the time in time windows in a dynamic network. The result is a tree and thus time windows can themselves contain smaller ones. Moreover, the windows do not have to be consecutive and this allows for instance to detect repeated structure. Each window is also associated to a community decomposition that represents the topological structure of the network during this window. We then apply the method to three dynamic networks to show that observed time windows correspond to observable phenomena.*

MOTS-CLÉS : *Communautés, Fenêtres de temps, Hiérarchie, Graphes de terrain, Réseaux dynamiques*

KEYWORDS: *Communities, Time Windows, Hierarchy, Complex Networks, Dynamic Networks*

1. introduction

L'étude des graphes de terrain est un domaine de recherche qui touche aujourd'hui de nombreuses disciplines : en informatique, l'Internet peut être vu comme un ensemble de routeurs interagissant via des câbles [LAT 08] ; en biologie, le cerveau est un ensemble de neurones interagissant entre eux [CHA 10] ; en sociologie, l'étude des réseaux sociaux conduit à étudier comment interagissent divers agents [WAL 09]. L'étude de la structure de ces réseaux a conduit à définir la notion de communautés. Bien que les réseaux soient en général très peu denses, on peut les diviser en plusieurs groupes de nœuds très densément connectés à l'intérieur mais avec peu de liens vers l'extérieur [NEW 04].

Le découpage en communautés permet une description générale de la forme des réseaux statiques. Or la majorité des graphes de terrain représentent des objets dynamiques (par exemple, des pages sont ajoutées, modifiées ou supprimées constamment sur le web) et donc beaucoup d'informations sont négligées en n'étudiant qu'un réseau statique. Dans cet article, nous allons étudier la structure des réseaux à la fois d'un point de vue topologique (qui est connecté à qui et comment) et d'un point de vue temporel (quand tel groupe existe par exemple). Pour cela, nous allons tout d'abord discrétiser le temps en un ensemble de pas de temps. Cette opération est en général très naturelle et liée à la méthode de mesure. Par exemple, si on mesure la topologie d'Internet chaque jour, chaque mesure correspond à un pas de temps d'une journée. Ensuite, nous allons regrouper les pas de temps pour lesquels le réseau a une structure semblable et on appellera chacun de ces regroupements une fenêtre de temps.

Ce regroupement sera naturellement hiérarchique. En effet, on imagine aisément qu'une fenêtre de temps peut contenir elle-même d'autres fenêtres de temps. Par exemple, si on étudie un réseau social de qui est en contact avec qui, on pourra trouver une fenêtre correspondant au jour et une fenêtre correspondant à la nuit. Mais la journée pourra elle-même être divisée en une matinée, un repas de midi, une après-midi, etc. Une particularité de notre approche est qu'elle ne force pas les fenêtres de temps à être d'un seul tenant. Une fenêtre peut donc contenir des pas de temps à des moments différents. Cela permet notamment de détecter des structures répétées et offre une plus grande résistance aux bruits pendant la mesure (quand il y a un problème ponctuel de mesure, une fenêtre de temps contiguë impose de découper en deux fenêtres). Ainsi, on peut imaginer que la structure change au cours de la journée mais que les matinées sont plus proches entre elles. Ensuite, les matinées sont plus proches des après-midis (moments typiques ou les gens sont au bureau) tandis que les repas de midi se ressemblent (on ne va pas forcément manger avec ses collègues directs) et que les soirées sont plus en relation entre elles (moments typiquement passés en famille). Cela permet aussi de détecter des événements répétés que les approches de découpage en fenêtres de temps contiguës ne voient en général pas [SUN 07].

Ce papier est décomposé comme suit. Nous allons dans une première section définir la notion de fenêtre de temps et proposer une métrique pour estimer la qualité de communautés durant une fenêtre de temps. Ensuite, nous étudierons comment di-

viser le temps en une hiérarchie de fenêtres de temps. Nous validerons ensuite notre approche en étudiant les résultats obtenus sur différents réseaux avant de conclure.

2. Fenêtres de temps et structure durant une fenêtre

Nous considérons ici qu'un graphe dynamique est une succession de graphes statiques représentant chacun l'état du réseau à un moment donné. On notera $S = \{1, 2, ..., n\}$ l'ensemble des pas de temps et $G = \{G_1, G_2, ..., G_n\}$ le graphe dynamique avec $G_i = (V_i, E_i)$ l'instantané à l'instant i ayant pour nœuds V_i et pour arêtes E_i. On notera aussi $V = \cup_{i \in \{1,...,n\}} V_i$ l'ensemble de tous les nœuds apparaissant à un moment quelconque dans l'évolution du graphe. Dans de nombreuses situations, les pas de temps n'ont pas tous la même importance et il est possible d'assigner un poids w_i au pas de temps i. Cela permet par exemple de considérer le cas ou les pas de temps ne sont pas régulièrement espacés en tenant compte de l'espacement dans le poids.

Dans ces conditions, une fenêtre de temps T est simplement un sous ensemble de S. La durée d'une fenêtre de temps est la somme des poids des pas de temps la composant ou, plus simplement, le nombre de pas de temps qu'elle contient s'il n'y a pas de poids. Une fenêtre de temps est dite contiguë si elle est un intervalle de S.

Afin de représenter la structure du réseau durant une fenêtre de temps, nous allons utiliser la notion de partition multi-pas définie précisément dans [AYN 11]. L'idée est de non pas chercher une partition pertinente sur un pas de temps donné ce que permet la modularité [NEW 04] et la détection de communautés classique [FOR 09], mais de chercher une partition qui soit pertinente sur un ensemble de pas de temps. Ainsi, la partition multi-pas pour une fenêtre de temps $T \subseteq S$ est une unique partition de V et non simplement des nœuds V_i d'un pas de temps donné i.

Pour que cette partition représente la structure du graphe, nous allons utiliser la partition qui maximise une fonction de qualité sur une fenêtre et pour cela on utilise la modularité moyenne durant la fenêtre de temps. Par conséquent, la modularité moyenne de la partition π de V durant la fenêtre de temps T est définie comme :

$$Q_{avg}(G, \pi, T) = \frac{1}{\sum_{i \in T} w_i} \sum_{i \in T} w_i . Q(G_i, \pi) \qquad [1]$$

Où $Q(G_i, \pi)$ est la modularité de la partition π sur le graphe statique G_i en ne considérant que les nœuds de V_i. On appellera celle-ci la modularité statique pour ne pas les confondre. Détecter des communautés sur une fenêtre T revient alors à trouver la partition maximisant la modularité moyenne sur T. Tout comme l'optimisation de la modularité statique [BRA 06], maximiser la modularité moyenne est NP-complet. Nous avons utilisé pour l'optimiser la méthode de Louvain [BLO 08] modifiée pour chercher des partitions multi-pas tel qu'expliqué dans [AYN 11].

Bien qu'il ait été montré qu'on peut effectivement trouver une partition qui soit bonne durant une longue durée [AYN 10a], la fenêtre de temps a ici une importance capitale car il semble vain de tenter de trouver une structure commune sur des instantanés très différents. Nous allons donc proposer une méthode de détection des fenêtres pertinentes dans la section suivante.

3. Détection automatique

Intuitivement on souhaite que les fenêtres de temps puissent elles-mêmes en contenir, nous allons donc chercher une décomposition du temps sous forme de regroupement hiérarchique (un dendrogramme). Chaque feuille correspondra à un pas de temps et les nœuds de l'arbre au regroupement de ses fils, c'est à dire des fenêtres de temps.

3.1. *Algorithme*

Pour construire un tel arbre, nous utilisons un algorithme classique de regroupement hiérarchique qui consiste à regrouper à chaque étape les deux fenêtres de temps les plus similaires : au début, chaque pas de temps (qui est une fenêtre triviale) est tout seul et on regroupe récursivement les deux fenêtres les plus similaires jusqu'à ce qu'il ne reste aucune possibilité (voir algorithme 1). Cet algorithme très standard nécessite une fonction de similarité pour décider quelles fenêtres regrouper.

Algorithme 1 Algorithme de regroupement hiérarchique de fenêtres de temps

G le graphe initial avec pour pas de temps S
L la liste des fenêtres potentielles, initialement vide
pour tout pas de temps t de S **faire**
    Chercher la partition $\pi_{\{t\}}$ sur la fenêtre $\{t\}$
    Mettre $\{t\}$ dans L
fin pour
tant que L n'est pas vide **faire**
    Trouver T_i et T_j dans L qui maximise $Sim(T_i, T_j)$
    Enlever T_i et T_j de L et ajouter $T = T_i \cup T_j$ dans L
    Afficher que T_i et T_j ont été regroupés
    Calculer la partition π_T de V pour la fenêtre T
fin tant que

3.2. *Métrique de similarité*

Pour définir une similarité entre fenêtres de temps, nous allons utiliser les partitions associées. Nous utilisons comme hypothèse que si deux fenêtres de temps sont structurellement similaires, la partition multi-pas de l'une, qui résume sa structure,

sera aussi une relativement bonne décomposition pour l'autre et inversement. Cela peut être évalué pour deux fenêtres T_i et T_j associées aux partitions π_i et π_j par la fonction de similarité :

$$Sim(T_i, T_j) = Q_{avg}(G, \pi_i, T_j) + Q_{avg}(G, \pi_j, T_i) \quad [2]$$

On n'effectue pas une comparaison des partitions à cause de l'instabilité des algorithmes de décomposition [AYN 10b]. Inversement, la modularité est stable et par conséquent nous l'avons utilisée pour dire si deux fenêtres de temps ont des structures similaires.

Quelques extensions de l'algorithme sont possibles. Par exemple, on peut vouloir forcer les fenêtres regroupées à être contiguës : dans ce cas, les résultats sont plus simples à interpréter mais des structures répétées à des moments espacés dans le temps ne seront pas détectées. Par la suite, nous allons aussi interdire de regrouper des fenêtres de temps dont la similarité est négative [1]. En utilisant ces règles, l'algorithme produit une forêt d'arbres disjoints plutôt qu'un seul arbre.

Le résultat de l'algorithme est double. Il y a d'une part le regroupement des pas de temps en une hiérarchie de fenêtres de temps représentant la structure temporelle mais aussi pour chaque fenêtre de temps une partition représentant la structure du réseau d'un point de vue topologique.

3.2.1. *Complexité algorithmique*

L'algorithme produit ces segmentations du temps relativement rapidement. Si le graphe dynamique est composé de n instantanés, il y a en tout $2n$ détection de communautés à effectuer. Il y a au départ une détection à faire sur chaque snapshot, soit n (mais sur des petites parties du graphe dynamique) puis ensuite il y a au plus n regroupements donc au plus n décompositions et on arrive au final à $2n$ applications de la méthode de Louvain. La complexité de la méthode de Louvain est délicate à évaluer. En effet, il existe une borne supérieure élevée connue mais l'algorithme se comporte de manière quasi linéaire en pratique sur des réseaux ayant des communautés. Il faut aussi compter que chaque décomposition est évaluée sur tous les snapshots, mais un calcul de modularité revient à faire un parcours de toutes les arêtes et est linéaire. Il y a donc $2n$ calculs qui sont linéaires. Dans la pratique, la principale limitation est la mémoire pour stocker le graphe en mémoire.

4. Résultats sur les réseaux.

Nous présentons maintenant les arbres obtenus sur différents réseaux présentant des caractéristiques différentes.

---

1. la décomposition triviale qui regroupe tous les nœuds dans une seule communauté a une modularité nulle et par conséquent une modularité négative est signe d'une décomposition vraiment mauvaise !

4.1. *Réseaux étudiés.*

Blogs [COI 09]. Pendant quatre mois, approximativement six milles blogs ont été monitorés afin de suivre leurs différents articles, commentaires et les liens de citations entre eux. Nous avons utilisé les réseaux entre les blogs contenant les données agrégées jusqu'au jour considéré. Nous commençons donc par un réseau vide au jour 0, puis ajoutons chaque jour les nouveaux blogs et liens entre eux. Nous obtenons donc un réseau croissant constitué de 120 pas de temps. Ce réseau croît lentement et régulièrement.

Mrinfo [PAN 09]. Le deuxième réseau correspond à une cartographie de la topologie des routeurs multicast sur Internet. Elle a été mesurée grâce à l'outil `mrinfo`. Ce programme permet de demander à un routeur multicast la liste de ses voisins. Chaque jour, `mrinfo` a été lancé sur un premier routeur puis ensuite récursivement sur chacun de ses voisins à la manière d'un parcours en largeur jusqu'à ne plus trouver aucun routeur. Cette mesure a été menée durant plusieurs années ce qui a permis d'obtenir une carte dynamique des routeurs multicasts. Pour plus de détails à propos de la mesure, nous renvoyons à [PAN 09]. Nous n'utiliserons ici que les données de l'année 2005, ce qui représente 365 pas de temps contenant chacun 3100 nœuds en moyenne. L'évolution de ce réseau est constituée de 3 phases. La première phase dure pendant les 52 premiers jours durant lesquels le réseau est assez instable et contient de nombreux événements. La seconde phase a lieu entre les jours 52 et 117 et est bien plus stable. Enfin, la troisième phase dure du jour 117 à la fin et ressemble plus à la première phase mais en plus longue, nettement plus stable et avec moins d'événements.

Imote [HUI 05]. Il s'agit d'un réseau de capteurs représentant la proximité entre les participants de la conférence Infocom en 2005. Plusieurs participants se sont vus remettre un périphérique *Bluetooth* appelé Imote permettant de détecter les autres Imote à proximité et de conserver cette information. L'expérience a duré trois jours à partir du soir précédant la conférence jusqu'à la fin de la troisième matinée. La mesure est extrêmement bruitée car les périphériques Bluetooth sont très imprécis et échouent régulièrement à détecter les périphériques alentours. Ainsi, un tout petit mouvement d'un participant peut suffire à changer pour un court instant ses voisins. Le réseau est donc extrêmement dynamique et comporte de nombreux changements. Il contient 41 nœuds et est divisé en 25000 pas de temps correspondant à 250000 secondes. Tous les pas de temps ne sont pas séparés par la même durée. Le réseau est en outre très stable et très peu dense durant la nuit et très instable et plus dense le jour

Nous avons donc utilisé trois réseaux aux caractéristiques très différentes. Mrinfo a une évolution assez lente et nettement divisée en phases tandis qu'Imote est très bruité et a une évolution très rapide. Blogs lui est un graphe croissant lentement. Enfin chaque instantané d'Imote est petit mais il y en a de très nombreux tandis que Mrinfo contient moins de pas de temps mais chacun sont assez gros. L'algorithme a aussi été appliqué sur un graphe de topologie de l'Internet de presque 5000 pas de temps ayant en moyenne 17000 sommets mais les résultats y sont difficiles à interpréter et nous ne les décrirons pas ici.

4.2. *Résultats et validation.*

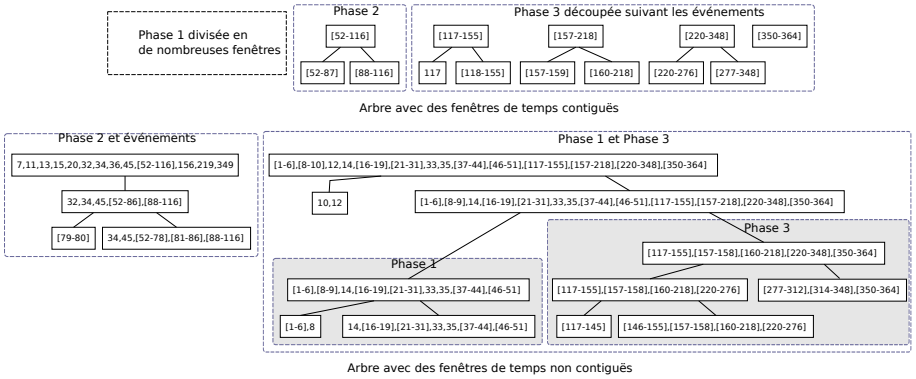

Figure 1. *Premiers niveaux dans la décomposition hiérarchique du temps sur Mrinfo. Les nombres dans les nœuds de l'arbre représentent les fenêtres de temps. Par exemple, "32,34,45,[52-86],[88-116]" correspond à la fenêtre de temps contenant les jours 32, 34, 45, de 52 à 86 et de 88 à 116. Nous avons groupé les fenêtres de temps en blocs avec une explication potentielle.*

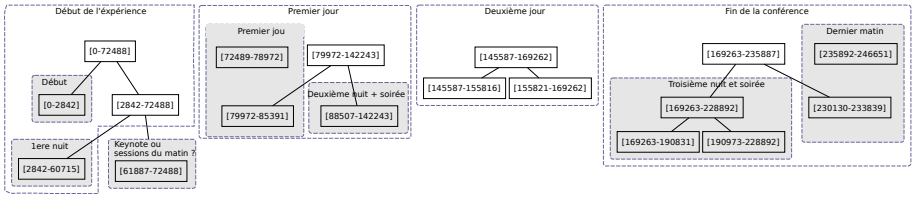

Figure 2. *Premiers niveaux de la décomposition hiérarchique du temps sur Imote. Les temps sont en secondes depuis le début de l'expérience.*

La validation des résultats est une question importante et nous le faisons encore manuellement. Nous avons vérifié que les fenêtres de temps identifiées par l'algorithme correspondent à des phénomènes compréhensibles sur le graphe obtenus soit via des connaissances externes sur le réseau (une idée du déroulement de la conférence sur Imote) soit via des visualisations du réseau. Cela permet de valider à la fois l'algorithme et la fonction de similarité. Nous avons construit les arbres pour chaque jeu de données sans et avec l'obligation de regrouper des fenêtres contiguës. Il faut remarquer que si on oblige les fenêtres de temps à être contiguës, le moindre problème de mesure ou un changement soudain de structure pendant un snapshot empêche les fenêtres d'avant et d'après le changement de se regrouper et implique une coupure en deux arbres disjoints.

Avec Mrinfo en utilisant des fenêtres contiguës (figure 1 en haut), le niveau le plus élevé qui contient les fenêtres de temps les plus longues est plus divisé que les

3 phases décrites précédemment. En effet, il y a de nombreux événements pendant la première phase qui ne peut par conséquent pas être identifiée comme une grande fenêtre. Le groupe de pas de temps entre le jour 52 et le jour 116 correspond à la phase 2 et les fenêtres de temps après correspondent à la phase 3 entrecoupée de 3 événements identifiés dans les données. Le premier niveau semble donc bien correspondre à l'impression que l'on avait de la dynamique et il est aussi possible d'expliquer les séparations aux niveaux inférieurs correspondant à des disparitions et des apparitions de gros blocs dans le réseau.

Quand on regarde la décomposition avec des fenêtres non contiguës (figure 1 en bas), l'effet causé par les événements ponctuels disparaît. Le premier niveau est composé de deux groupes : un contenant la phase 2 et quelques événements et un contenant les phases 1 et 3, qui sont structurellement assez similaires. Ce dernier groupe est ensuite divisé en deux sous groupes contenant la phase 1 et la phase 3. Ensuite, les sous niveaux correspondent à des fenêtres contiguës, séparées par les événements. On peut remarquer que bien qu'on n'impose aucune contrainte de contiguïté, les fenêtres contiennent des pas de temps très rapprochés et sont quasi contiguës. Il est en effet raisonnable de penser que deux snapshots proches dans le temps ont plus de chance d'être similaires structurellement.

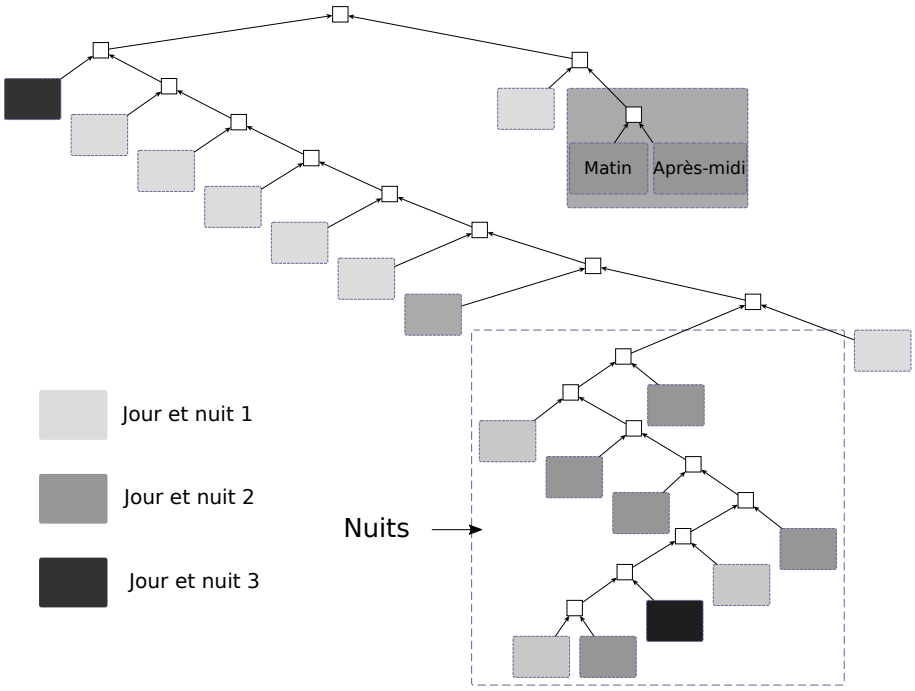

Figure 3. *Plus grand arbre de la décomposition hiérarchique du temps d'Imote avec des fenêtres de temps non contraintes à être contiguës. Les blocs représentent de grands sous arbres afin de permettre une représentation la structure globale.*

Nous obtenons des résultats similaires avec Imote. En utilisant des fenêtres de temps contiguës, le niveau le plus haut contient plusieurs fenêtres qui correspondent à des moments identifiés (voir figure 2). Le premier correspond au début de l'expérience et contient trois groupes : les 45 minutes après le début, une longue période correspondant à la première nuit et deux autres heures et demie qui pourraient correspondre au petit déjeuner et au premier keynote (nous n'avons pas la correspondance exacte entre les temps de l'expérience et de la conférence et ne pouvons donc qu'émettre des hypothèses). Ensuite, il y a une petite fenêtre qui peut correspondre à une partie du premier jour. La fenêtre suivante est la suite du premier jour avec la seconde nuit. La fenêtre suivante contient majoritairement le second jour, qui est lui même divisé en deux parties qui correspondent pratiquement avec le matin et l'après midi. La fenêtre suivante est composée de la dernière nuit et de la dernière matinée et enfin la dernière fenêtre contient les quelques moments restants de la dernière matinée, correspondant a priori au retour des périphériques et la fin de l'expérience.

Nous donnons sur la figure 3 une représentation simplifiée du plus grand arbre (les autres sont très petits) de la décomposition obtenue sans la contrainte de contiguïté. L'arbre est vraiment grand et il y a énormément de sous niveaux, nous ne pouvons donc pas le représenter en entier mais juste donner une idée de sa structure. Ainsi, chaque bloc de la figure 3 correspond en fait à un sous arbre. On constate que les sous arbres sont tous constitués de pas de temps appartenant à la même journée, ce qui tend à montrer que la structure identifiée a du sens. Les deux plus grands groupes sont le jour 2 et les nuits. Le jour 2 est lui décomposé en deux parties, matin et après midi, tandis que le groupe contenant les nuits contient plusieurs sous arbres contenant des snapshots du même jour. Ainsi, malgré la forte dynamique du réseau, l'algorithme arrive à identifier des motifs suffisamment récurrents.

Avec le jeu de données Blogs, les arbres de fenêtres contiguës et non contiguës sont très proches. Ils sont tous deux très déséquilibrés et presque dégénérés en une sorte de liste de pas de temps, s'ajoutant un par un pour former un grand peigne. Ceci est consistant avec le fait que le réseau ne fait que croître : il n'y a pas ici de fenêtres de temps clairement identifiées. C'est pourquoi nous n'avons représenté que l'arbre du cas non contiguë sur la figure 4. Il y a très peu de longues fenêtres de temps. La plus grande fenêtre contient les premiers snapshots, qui sont en effet les plus différents de la structure moyenne. Nous n'avons pas réussi à interpréter les autres fenêtres par manque de connaissance sur l'objet mesuré. Nous savons uniquement qu'un événement de mesure a eu lieu au jour 40 et cet événement apparaît effectivement dans les deux fenêtres les plus basses.

Finalement, ces résultats semblent assez prometteurs. La décomposition est souvent pertinente quand les graphes ont une évolution divisée en phases. Les deux approches, à fenêtres contiguës ou non, sont complémentaires : les fenêtres contiguës sont plus simples à interpréter et produisent plus directement des arbres exploitables tandis que les cas non contiguës sont moins sensibles aux événements et peuvent détecter des répétitions de structures, comme entre la phase 1 et 3 de Mrinfo.

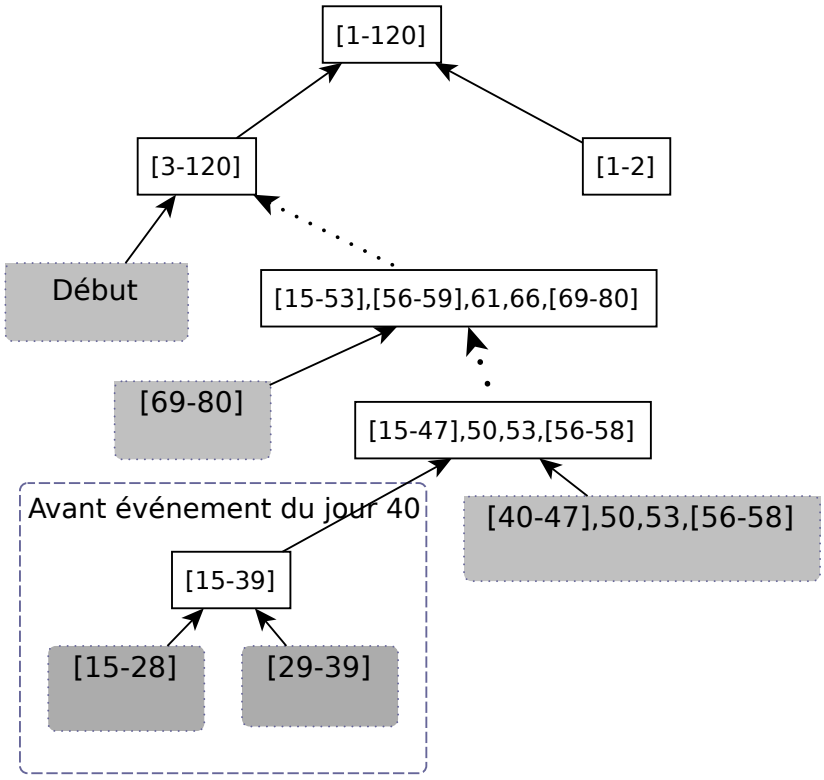

Figure 4. *Arbre de la décomposition du temps de Blogs. Les blocs gris représentent des sous arbres et les flèches pointillées de longs peignes agrégeant les snapshots un par un.*

5. Conclusions

Nous avons proposé une nouvelle méthode pour analyser la structure des réseaux dynamiques. Elle permet d'analyser d'une part la structure d'un point de vue topologique en utilisant une extension de la modularité mais aussi et surtout la structure temporelle en montrant que certains pas de temps ont une structure topologique similaire. Contrairement aux diverses méthodes de détection de fenêtres de temps basées sur la détection de changements majeurs, notre méthode permet de détecter des événements qui se répètent ou par exemple un changement temporaire de structure (comme le moment du repas dans une journée de travail).

Ces résultats donnent de nouvelles informations sur les réseaux étudiés. Le processus de construction de Blogs a un effet significatif sur les conclusions que l'on peut obtenir et on peut imaginer qu'il faut diminuer le poids des liens vus longtemps auparavant ou considérer le réseaux des liens vus sur une durée donnée plutôt que depuis

le début de la mesure. Nos algorithmes sont plus efficaces sur Mrinfo et Imote où les phases que nous arrivons à détecter existent effectivement.

Une difficulté restant est la taille des arbres produits et le fait que ce soit des arbres binaires. On obtient des arbres ayant une profondeur correspondant au nombre de pas de temps et ils sont difficiles à analyser. Pour regrouper un nombre donné de pas de temps, il faut les ajouter un par un et cela transforme l'arbre en un long peigne. Il faudra donc définir un filtrage adéquat permettant de vraiment mettre en valeur les regroupements intéressants. Pour l'instant, nous avons représenté la structure globale de l'arbre à la main après un filtrage simplifiant les peignes mais cela manque de généralité et reste inefficace.

Tous les découpages de l'arbre n'ont pas la même pertinence. Si on se remémore les motivations de la modularité, cette fonction servait avant de devenir une fonction objectif à trouver dans un dendrogramme le niveau le plus pertinent. On peut imaginer de la même manière un moyen de trouver des partitionnements pertinents des pas de temps en fenêtres de temps à différentes échelles.

D'autres fonctions de similarité peuvent aussi exister. La fonction utilisée a le défaut de mesurer à la fois si deux fenêtres sont similaires et modulaires. Ainsi, deux fenêtres de temps identiques seront d'autant plus proches que leur structure est modulaire et une normalisation adaptée pourrait corriger ce défaut. Nous en avons essayé quelques-unes sans que l'amélioration des résultats soit probante.

Remerciements

Ce travail a été financé par une bourse de l'*Agence Nationale de la Recherche*, avec pour références ANR-10-JCJC-0202 et par le projet WebFluence #ANR-08-SYSC-009.

6. Bibliographie

[AYN 10a] AYNAUD T., GUILLAUME J.-L., « Détection de communautés à long terme dans les graphes dynamiques », *Journée thématique : Fouille de grands graphes*, vol. d, 2010, p. 1–4.

[AYN 10b] AYNAUD T., GUILLAUME J.-L., « Static community detection algorithms for evolving networks », *Wireless Networks*, vol. 2010, 2010, p. 508–514.

[AYN 11] AYNAUD T., GUILLAUME J.-L., « Multi-Step Community Detection and Hierarchical Time Segmentation in Evolving Networks », *SNAKDD*, vol. 11, 2011.

[BLO 08] BLONDEL V. D., GUILLAUME J.-L., LAMBIOTTE R., LEFEBVRE E., « Fast unfolding of communities in large networks », *J. Stat. Mech*, vol. 10008, 2008, p. 1–12.

[BRA 06] BRANDES U., DELLING D., GAERTLER M., GOERKE R., HOEFER M., NIKOLOSKI Z., WAGNER D., « Maximizing Modularity is hard », *ArXiv Physics e-prints*, , 2006.

[CHA 10] CHAVEZ M., VALENCIA M., NAVARRO V., LATORA V., « Functional modularity of background activities in normal and epileptic brain networks », *Physical review letters*, , 2010, p. 1–5.

[COI 09] COINTET J.-P., ROTH C., « Socio-semantic Dynamics in a Blog Network », *2009 International Conference on Computational Science and Engineering*, , n° 6, 2009, p. 114–121, Ieee.

[FOR 09] FORTUNATO S., « Community detection in graphs », *Physics Reports*, , 2009.

[HUI 05] HUI P., CHAINTREAU A., SCOTT J., GASS R., CROWCROFT J., DIOT C., « Pocket switched networks and human mobility in conference environments », *Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking - WDTN '05*, , 2005, p. 244–251, ACM Press.

[LAT 08] LATAPY M., MAGNIEN C., OUÉDRAOGO F., « A Radar for the Internet », *2008 IEEE International Conference on Data Mining Workshops*, n° 1, IEEE, décembre 2008, p. 901–908.

[NEW 04] NEWMAN M. E. J., GIRVAN M., « Finding and evaluating community structure in networks », *Physical Review E*, vol. 69, n° 2, 2004, page 26113, APS.

[PAN 09] PANSIOT J., MÉRINDOL P., DONNET B., BONAVENTURE O., « Extracting Intra-Domain Topology from mrinfo Probing », *Passive and Active Measurement*, 2009.

[SUN 07] SUN J., FALOUTSOS C., PAPADIMITRIOU S., YU P., « Graphscope : parameter-free mining of large time-evolving graphs », *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM New York, NY, USA, 2007, p. 687–696.

[WAL 09] WALLACE M. L., GINGRAS Y., DUHON R., « A new approach for detecting scientific specialties from raw cocitation networks », *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, 2009, p. 240–246, John Wiley \& Sons, Inc.