Triclustering pour la détection de structures temporelles dans les graphes

Romain Guigourès[*,**] **– Marc Boullé**[*] **– Fabrice Rossi**[**]

[*]*Orange Labs*
2 avenue Pierre Marzin
F-22300 Lannion
prenom.nom@orange.com

[**]*SAMM EA 4543, Université Paris 1*
90 rue Tolbiac
F-75013 Paris
prenom.nom@univ-paris1.fr

RÉSUMÉ. *Cet article présente une technique permettant la découverte de structures dans les graphes temporels. La méthode est basée sur une approche de coclustering en trois dimensions ne requérant aucun paramètre utilisateur. Les nœuds source, les nœuds cible, ainsi que la variable temporelle sont simultanément segmentés afin de construire des intervalles de temps et des clusters de nœuds entre lesquels la distribution des arcs est stationnaire par intervalle de temps. L'intérêt de cette approche réside dans la discrétisation temporelle qui est directement déduite de l'évolution de la distribution des arcs entre les nœuds, ce qui permet d'éviter une pré-discrétisation du temps. Des expérimentations sur un jeux de données synthétiques illustrent le bon comportement de la technique, et une étude d'un jeu de données réelles montre le potentiel de l'approche proposée pour l'analyse exploratoire des données.*

ABSTRACT. *This paper introduces a novel technique to track structures in time evolving graphs. The method is based on a parameter free approach for three-dimensional co-clustering of the source vertices, the target vertices and the time. All these features are simultaneously segmented in order to build time segments and clusters of vertices whose edge distributions are similar and evolve in the same way over the time segments. The main novelty of this approach lies in that the time segments are directly inferred from the evolution of the edge distribution between the vertices, thus not requiring the user to make an a priori discretization. Experiments conducted on a synthetic dataset illustrate the good behaviour of the technique, and a study of a real-life dataset shows the potential of the proposed approach for exploratory data analysis.*

MOTS-CLÉS : *Coclustering, Blockmodeling, Fouille de Graphe, Graphe temporel*

KEYWORDS: *Coclustering, Blockmodeling, Graph Mining, Time Evolving Graph*

1. Introduction

Dans les problématiques actuelles, on s'intéresse de plus en plus à l'évolution temporelle des interactions entre individus. Par exemple, dans le cadre des réseaux de collaborations entre scientifiques, lorsque de nouveaux doctorants sont recrutés ou terminent leur thèse, quand des chercheurs changent de laboratoire ... Comprendre l'évolution des interactions dans un graphe implique de découvrir à la fois les structures du graphe et leur évolution au cours du temps. Dans cet article, on aborde ce problème en introduisant une méthode de *blockmodeling temporel*.

Le concept de blockmodeling est à l'origine des premiers travaux d'analyse des structures dans les graphes menés par les sociologues dès les années 1950 dans le contexte de l'analyse des réseaux sociaux. Les nœuds d'un graphe représentent les acteurs du réseau, alors que les arcs modélisent les interactions sociales qui les lient. Parmi ces travaux, les sociologues ont proposé de structurer les relations entre les acteurs en rôles [BOT 57, NAD 57], conduisant à définir la notion d'équivalence structurelle [LOR 71] : deux acteurs sont dits *structurellement équivalents* s'ils jouent le même rôle dans le réseau social, c'est-à-dire s'ils interagissent avec les mêmes acteurs. En regroupant les acteurs structurellement équivalents, qui correspondent aux nœuds dans le graphe correspondant, on obtient une version simplifiée et synthétique du graphe d'origine. Une généralisation de l'équivalence structurelle a été proposée par la suite afin de se libérer de certaines contraintes considérées comme trop restrictives. Il s'agit de l'équivalence régulière qui consiste à regrouper des acteurs dans des clusters qui interagissent de manière identique avec les autres clusters.

Il est fréquent d'utiliser une représentation matricielle du graphe, portant le nom de matrice d'adjacence, afin de déterminer sa structure sous-jacente. Les lignes et les colonnes représentent les acteurs, et les valeurs de la matrice indiquent s'il y a interaction ou non entre les acteurs représentés. Les premières approches sociologiques suggérèrent de réorganiser les lignes et les colonnes dans le but de découper la matrice en blocs homogènes. Cette technique est appelée *Blockmodeling*. Une fois les blocs extraits, une partition des acteurs représentés à la fois par les lignes et les colonnes peut être réalisée. Cette segmentation simultanée est appelée *coclustering*. Une manière de représenter le coclustering est sous la forme d'un *graphe image*, dont les nœuds modélisent les clusters d'acteurs identifiés dans le blockmodeling et les arcs les rôles tels que les définit l'équivalence régulière. En outre, il y aura un arc entre deux clusters s'il y a interactions entre les nœuds qui les composent.

De nombreuses méthodes ont été proposées pour construire un graphe image. Certaines se basent sur l'optimisation d'un critère [DOR 04] permettant d'isoler des blocs homogènes en se focalisant sur les blocs vides comme il est suggéré dans [WHI 76]. Des approches déterministes plus récentes se sont intéressées à l'optimisation de critères qui mesurent la qualité du graphe image en termes de résumé du graphe d'origine [REI 07]. D'autres utilisent le blockmodeling stochastique. Dans ces modèles génératifs, une variable latente indiquant l'appartenance ou non à un cluster est associée à chaque nœud. Conditionnellement à leur variable latente, la probabilité d'observer un arc entre deux acteurs suit une loi de probabilité (Bernoulli dans les cas les plus simples) dont les paramètres dépendent uniquement de la paire de clusters désignés par la variable latente. Les premières approches nécessitaient une paramétrisation du nombre de clusters par

l'utilisateur [NOW 01], alors que les méthodes les plus récentes préfèrent le déterminer automatiquement en utilisant un processus de Dirichlet [KEM 06].
Les études des graphes temporels sont relativement récentes. La majorité des méthodes définissent un graphe temporel comme une séquence de graphes statiques auxquels sont attribués une valeur ordinale. Dans certaines approches, les intervalles de temps sont obtenus par classification ascendante, groupant ainsi les graphes statiques à l'aide d'une mesure de similarité [HOP 04]. En ce qui concerne le blockmodeling stochastique, une méthode prenant en compte l'évolution de la variable latente au cours du temps permet d'étudier l'évolution des groupes au sein du graphe [XIN 10].
Dans cet article, une méthode de blockmodeling est adaptée aux graphes temporels sur la base de l'approche MODL [BOU 11]. Comme dans le blockmodeling classique, la méthode non paramétrique présentée ici regroupe les nœuds ayant des distributions d'arcs similaires sur les clusters. En parallèle, une partition du temps en intervalles dans lesquels la structure du graphe est stationnaire est effectuée. Dans le but d'obtenir une représentation plus synthétique du graphe temporel, une méthode de triclustering est introduite. Elle optimise un critère qui permet de segmenter simultanément les nœuds et la variable temporelle. Cette approche est résistante au bruit et fiable dans le sens où aucune structure de coclustering ne sera détectée si le graphe est aléatoire et aucune discrétisation temporelle ne sera produite sur un graphe stationnaire. La section 2 décrit en détail la méthode de triclustering employée et introduit une technique efficace d'analyse exploratoire des données. La section 3 analyse le comportement du critère sur un jeu de données artificielles. Enfin, dans la section 4, la méthode est appliquée à un jeu de données réelles afin de montrer son efficacité sur un cas pratique.

2. Modèle de Graphe temporel

Les graphes étudiés sont à arcs multiples, car la méthode ne se restreint pas aux matrices d'adjacence binaires ou symétriques. Notons un graphe temporel $\mathcal{G} = \langle V_S, V_T, E(t) \rangle$ où l'ensemble des nœuds V_S et V_T sont constants et $E(t)$ est l'ensemble des arcs observés au temps $t \in [T_{min}, T_{max}]$. Cette modélisation de graphe est généralisable aux graphes simples, à arcs multiples, orientés et bipartis.

2.1. *Définition du modèle*

Étant donné que le graphe évolue au cours du temps, la représentation synthétique par un graphe image est remplacée par une séquence de graphes images $\mathcal{IG} = (\mathcal{IG}_n)_{n=1,\ldots,N}$, chacun d'eux étant associé à un intervalle de temps. Les descriptions du graphe et de son image sont données par le Tableau 1. Une fois les différentes caractéristiques du graphe image introduites, leur paramétrisation doit être spécifiée. Un modèle caractérisant un graphe image est défini par :

1) le nombre de clusters source et cible (K_S et K_T) ;

2) le nombre d'intervalles temporels (N) ;

3) la partition des nœuds source (resp. cible) sur les clusters source (resp. cible) ;

4) la distribution des arcs temporels du graphe sur les triclusters, intersections des clusters source, clusters cible et intervalles de temps (c'est-à-dire les arcs du graphe

image) ;

5) pour chaque cluster source (resp. cible), la distribution des degrés sortant du cluster source (resp. entrant dans le cluster cible) sur les nœuds qui le composent.

Graphe $\mathcal{G}$	**Graphe Image $\mathcal{IG}$**
V_S ensemble des nœuds source	C_S ensemble des K_S clusters source
V_T ensemble des nœuds cible	C_T ensemble des K_T clusters cible
T variable temporelle	$I = \{I_1, I_2, ..., I_N\}$ les intervalles
$E(t \in T)$ ensemble d'arcs temporels	$E_{\mathcal{IG}}(I_n)$ arcs inter-clusters à I_n

Tableau 1 – Notations pour le graphe initial et son image synthétique

Notons que les spécifications définies dans les troisième et cinquième points ne sont pas nécessaires pour la variable temporelle. Une bonne discrétisation temporelle doit être invariante par toute transformation monotone du temps en entrée et robuste vis-à-vis des valeurs atypiques (outliers). Ces exigences définies, il a été choisi d'utiliser les rangs des valeurs temporelles plutôt que les valeurs elles-même. Ainsi, il n'est pas nécessaire de spécifier dans le modèle la distribution des valeurs temporelles sur les intervalles de temps puisque les intervalles suivent un ordre logique. Quant à la distribution des arcs sur les rangs localement à chaque intervalle, elle est également implicitement spécifiée puisqu'il y a un rang par valeur temporelle.

2.2. *Évaluation d'un modèle*

Dans le but d'inférer la meilleure partition des trois dimensions, un critère est construit suivant une approche MAP (Maximum A Posteriori), selon l'approche MODL [BOU 11]. Le critère est constitué d'une probabilité a priori (ou prior) sur le graphe image $P(\mathcal{IG})$, et de la vraisemblance du graphe connaissant les paramètres du graphe image $P(\mathcal{G}|\mathcal{IG})$.

2.2.1. *Prior*

Apprendre directement le modèle (graphe image) sur les données permettrait au modèle de capturer des phénomènes liés au bruit et donc d'augmenter le risque de surapprentissage. Pour éviter ce type de problèmes, un a priori sur le modèle va pénaliser la vraisemblance. Le prior est construit hiérarchiquement et uniformément à chaque étape afin d'être non-informatif [JAY 03]. L'ensemble des termes du prior est détaillé ci-dessous :

1) Le nombre de clusters source K_S (resp. cible K_T) est uniformément distribué entre 1 et $|V_S|$, le nombre de nœuds source (resp. $|V_T|$, le nombre de nœuds cible). Le cas avec un unique cluster correspond au modèle nul où aucune structure significative dans le graphe n'est capturée. Le cas avec autant de clusters que de nœuds correspond quant à lui au modèle le plus fin, c'est-à-dire au cas où chaque acteur du réseau joue un rôle suffisamment significatif pour être isolé dans un cluster. Ces deux schémas de clustering sont en accord avec la définition de l'équivalence régulière [BOR 88, WHI 83]. De la même manière le nombre d'intervalles de temps N est uniformément distribué entre 1 et $|E|$ le nombre total d'arcs. Le cas avec un unique intervalle modélisera

un graphe stationnaire alors que le cas avec autant d'intervalles que d'arcs sera une discrétisation extrêmement fine, mais comme le temps est une variable numérique, ce cas est pris en compte par la méthode.

$$p(K_S) = \frac{1}{|V_S|} ; p(K_T) = \frac{1}{|V_T|} ; p(N) = \frac{1}{|E|}$$

2) Pour un nombre donné de clusters source (resp. cible), chaque partition des $|V_S|$ nœuds source (resp. $|V_T|$ nœuds cible) en clusters est équiprobable.

$$p(\{C_S\}|K_S) = \frac{1}{B(|V_S|, K_S)} ; p(\{C_T\}|K_T) = \frac{1}{B(|V_T|, K_T)}$$

où $B(|V_S|, K_S) = \sum_{k=1}^{K_S} S(|V_S|, k)$ est une somme des nombres de Stirling de second ordre, c'est-à-dire le nombre de manières de partitionner $|V_S|$ éléments en k sous ensembles non-vides.

3) Pour un graphe image avec K_S clusters source et K_T clusters cible, chaque distribution des arcs sur les triclusters – définis comme l'intersection des clusters source, des clusters cible et des intervalles de temps – est équiprobable.

$$p(E_{\mathcal{IG}}(C_S, C_T, I_n)|K_S, K_T, N) = \frac{1}{\binom{|E|+K_S K_T N-1}{K_S K_T N-1}}$$

4) Pour un cluster source donné $c_i = \{v_i, i = 1..|c_i|\}$(resp. cluster cible $c_j = \{v_j, j = 1..|c_j|\}$), chaque distribution des degrés sortant (resp. entrant) sur les nœuds qui le composent est équiprobable.

$$p(d^{out}(v_i)|d^{out}(c_i), \{c_i\}) = \frac{1}{\binom{d^{out}(c_i)+|c_i|-1}{|c_i|-1}} ; p(d^{in}(v_j)|d^{in}(c_j), \{c_j\}) = \frac{1}{\binom{d^{in}(c_j)+|c_j|-1}{|c_j|-1}}$$

2.2.2. *Vraisemblance*

Une fois les paramètres du graphe image spécifiés, la vraisemblance $P(\mathcal{G}|\mathcal{IG})$ est définie comme la manière la plus probable d'observer le graphe initial connaissant les paramètres de son image.

1) Tous les tirages multinomiaux de $|e(c_i, c_j, I_n)|$ arcs entre les clusters (vus comme les nœuds du graphe image) à l'intervalle de temps I_n à partir des $|E|$ arcs du graphe d'origine sont équiprobables :

$$P(E|C_S, C_T, I) = \frac{\prod_{c_i \in C_S} \prod_{c_j \in C_T} \prod_{I_n \in I} |e(c_i, c_j, I_n)|!}{|E|!}$$

2) Pour chaque cluster source (resp.cible), toutes les manières de distribuer le degré du clusters sur les nœuds qui le composent sont équiprobables :

$$P(V_S|C_S) = \frac{\prod_{v_i \in V_S} d^{out}(v_i)!}{\prod_{c_i \in C_S} d^{out}(c_i)!} ; P(V_T|C_T) = \frac{\prod_{v_j \in V_T} d^{in}(v_j)!}{\prod_{c_j \in C_T} d^{in}(c_j)!}$$

3) Chaque distribution du rang temporel des arcs est équiprobable au sein de chaque intervalle de temps :

$$P(T|I) = \frac{1}{\prod_{I_n \in I} |I_n|!}$$

Le produit du prior et de la vraisemblance resulte en la probabilité a posteriori du modèle. Le logarithme négatif de cette dernière probabilité est utilisé pour construire le critère. En l'optimisant, les nœuds dont les arcs entrant (resp. sortant) sont distribués de manières similaires seront groupés et le temps sera discrétisé en intervalle dans lesquels la distribution des arcs est stationnaire. Ces propriétés sont illustrée en Section 3.

Définition (Coût du Graphe Image)**.** *Le graphe image $\mathcal{IG}$, représentation synthétique d'un graphe $\mathcal{G}$ est optimal s'il minimise le critère suivant :*

$$c(\mathcal{IG}) = -\log\left[P(\mathcal{IG})\right] - \log\left[P(\mathcal{G}|\mathcal{IG})\right] \tag{1}$$

D'un point de vue théorie de l'information, un logarithme négatif de probabilité correspond à une longueur de codage [SHA 48]. Ainsi, le logarithme négatif du prior est la longueur de codage du graphe image alors que le logarithme négatif de la vraisemblance est la longueur de description du graphe pour une paramétrisation du graphe image donnée. Minimiser la somme de ces deux termes a donc une interprétation naturelle en terme de MDL (Minimum Description Length) [GRÜ 07].
Le meilleur modèle est obtenu en optimisant le critère avec des algorithmes détaillés dans [BOU 11], qui ont des propriétés pratiques en terme de scalabilité, avec une complexité de $O(|E|)$ en espace et de $O(|E|\sqrt{|E|}\log|E|)$ en temps, exploitant ainsi pleinement l'aspect creux du graphe temporel.

2.3. *Simplifier le graphe image*

Quand des graphes volumineux sont étudiés, le nombre de clusters de nœuds et d'intervalles de temps peut devenir trop important pour une interprétation simple. Ce problème a été soulevé par [WHI 76] qui propose une méthode agglomérative en tant qu'outil d'analyse exploratoire.
La méthode d'analyse exploratoire proposée ici consiste à fusionner successivement les clusters et les intervalles de temps de la manière la moins coûteuse jusqu'à ce que le graphe image soit suffisamment synthétique pour être interprété. Depuis le modèle optimal selon le critère détaillé dans l'équation 1, les clusters source, les clusters cible et les intervalles de temps sont fusionnés itérativement. À chaque étape, les clusters fusionnés (ou les intervalles) sont ceux qui entrainent la plus faible augmentation du critère.

Théorème. *Asymptotiquement – c'est-à-dire quand le nombre d'arcs tend vers l'infini – la variation Δc du critère quand deux clusters sont fusionnés est égal à la divergence de Jensen-Shannon entre les distributions des arcs sur les clusters (ou intervalles de temps contigus) fusionnés.*

$$\begin{aligned}\Delta c(\cup(c_1, c_2)) &= (|c_1| + |c_2|) JS^{\alpha_1,\alpha_2}(P_1, P_2) \\ &= (|c_1| + |c_2|)\left(\alpha_1 KL(P_1||P_{1\cup 2}) + \alpha_2 KL(P_2||P_{1\cup 2})\right)\end{aligned} \tag{2}$$

où c_1 et c_2 sont les clusters (ou intervalles de temps) fusionnés en un cluster (ou intervalle de temps) $c_{1\cup2}$. P_1, P_2 et $P_{1\cup2}$ sont les distributions respectives de c_1, c_2 et $c_{1\cup2}$ sur les triclusters. Dans le cas d'une fusion de deux clusters source par exemple :

$$P_{i\in\{1,2\}} = \left\{ \frac{|e(c_i, c_j, I_n)|}{|E|} \right\}_{c_j \in C_T, I_n \in I} \quad P_{1\cup2} = \alpha_1 P_1 + \alpha_2 P_2 \quad \alpha_{i\in\{1,2\}} = \frac{|c_i|}{|c_1| + |c_2|}$$

JS est la Divergence de Jensen-Shannon généralisée et KL la divergence de Kullback-Leibler [LIN 91]. La preuve du théorème n'est pas détaillée ici pour des raisons de concision, mais le calcul de la variation de la vraisemblance en utilisant l'approximation de Stirling ($\log(n!) = n\log(n) - n + O(\log(n))$) permet d'arriver à ce résultat.
La divergence de Jensen-Shannon possède certaines propriétés intéressantes. Il s'agit d'une mesure définie positive et symétrique, elle est nulle lorsque deux distributions strictement identiques sont comparées. Bien qu'il ne s'agisse pas d'une distance, car non subadditive, elle bénéficie des propriétés minimales que requiert une mesure de dissimilarité lors d'un processus agglomératif [DHI 03]. Pour maîtriser la dégradation du modèle, une mesure d'informativité est calculée à chaque étape du processus agglomératif.

Définition (Informativité du graphe image)**.** *Le modèle nul $\mathcal{IG}_\emptyset$ est la paramétrisation du graphe image telle qu'il n'y ait qu'un cluster source, un cluster cible et un intervalle de temps. Ce modèle est caractéristique des graphes aléatoires (sans structure sous-jacente). Connaissant le graphe image optimal $\mathcal{IG}^*$ obtenu par minimisation du critère précédemment défini, l'informativité d'un graphe image $\mathcal{IG}$ est donnée par :*

$$\tau(\mathcal{IG}) = \frac{c(\mathcal{IG}) - c(\mathcal{IG}_\emptyset)}{c(\mathcal{IG}^*) - c(\mathcal{IG}_\emptyset)}$$

Par définition, $\tau(\mathcal{IG}) \leq 1$; Notons que $\tau(\mathcal{IG}) < 0$ est possible quand le graphe image est une modélisation non-pertinente du graphe $\mathcal{G}$ (par exemple si $\mathcal{IG} \neq \mathcal{IG}_\emptyset$ quand $\mathcal{G}$ est un graphe aléatoire).

3. Expérimentations sur des données artificielles

Des expérimentations ont été menées sur des données artificielles dans le but d'étudier les propriétés de l'approche décrite dans l'article. Pour ce faire, des graphes artificiels avec une structure sous-jacente connue ont été générés.

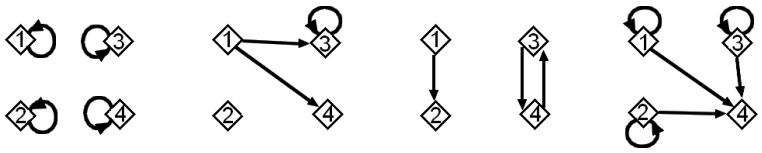
3.1. *Expérimentations sur des graphes avec structure*

Le graphe synthétique contient 40 nœuds et un nombre variable d'arcs. Les nœuds sont groupés en 4 clusters et le temps est divisé en 4 intervalles pour lesquels sont associés 4 graphes images avec des structures différentes (voir Figure 1).
Le jeu de données est généré en tirant les arcs suivant le processus suivant :

1) Un nœud source (et donc le cluster source auquel il est associé) et un timestamp (sur $[0, 100]$ et avec 10 décimales) sont choisis aléatoirement .

2) Le timestamp est associé à l'intervalle de temps correspondant ce qui permet de connaître le graphe image correspondant (Figure 1).

3) Si le cluster source est relié à un cluster cible dans le graphe image, alors un nœud cible est choisi aléatoirement parmi les nœuds cible éligibles et l'arc est tracé.



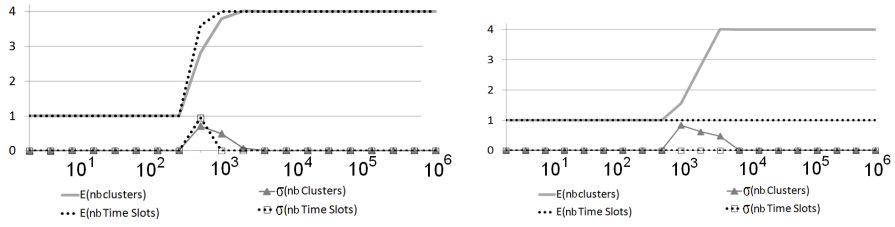
(a) I_1 (b) I_2 (c) I_3 (d) I_4

Figure 1 – Graphes images de chaque intervalle de temps

Une fois les arcs tracés, un bruit blanc est ajouté au jeu de données en réattribuant aléatoirement à 30% des arcs un nœud source et un nœud cible. Ce jeu de données a été généré avec un nombre variable d'arcs. Dans le but d'obtenir des résultats fiables, 100 graphes ont été construits par nombre d'arcs générés.
Les résultats sont présentés dans la Figure 2.(a). Pour un faible nombre d'arcs générés, la méthode ne détecte aucune structure car le volume de données est trop faible pour qu'un quelconque motif soit significatif. C'est le cas où le prior domine la vraisemblance. Puis le nombre de clusters trouvés augmente avec les arcs : certaines structures commencent à émerger. Enfin, au-delà de 2000 nœuds, il y a suffisamment d'arcs pour que la méthode retrouve les vraies structures sous-jacentes, et ce même pour un nombre important d'arcs générés. On observe donc empiriquement une convergence asymptotique.

3.2. *Expérimentations sur les graphes stationnaires*

Un graphe stationnaire est un graphe dont la structure régulière (distribution des arcs inter-clusters) n'évolue pas au cours du temps. Pour générer un tel graphe, les timestamps ont été réaffectés de manière aléatoire aux arcs du graphe. En faisant ça, on obtient une nouvelle distribution des arcs entre les clusters qui correspond à la distribution moyenne des 4 précédentes distributions. En deux mots, le graphe peut être considéré comme statique et le graphe image reste le même pour l'ensemble de l'intervalle $[0, 100]$, bien qu'il soit devenu plus complexe.
La Figure 2.(b) montre que peu importe le nombre d'arcs, la méthode ne discrétise



(a) Graphe avec structure temporelle (b) Graphe stationnaire

Figure 2 – (a) Résultats pour un graphe avec structure temporelle. (b) Résultats pour un graphe stationnaire. Les deux courbes sont le nombre moyen de clusters et d'intervalles de temps en fonction du nombre d'arcs. Les deux courbes du bas représentent les écart-types respectifs.

jamais la variable temporelle. Elle est donc résistante au bruit en ne créant pas d'intervalle de temps factice. Notons que la méthode a besoin d'un plus grand nombre d'arc pour retrouver la structure statique du graphe. Ceci est dû à la complexité accrue du graphe image.
Des expérimentations sur des graphes aléatoires ont également été menées (ajout d'un bruit de 100% au lieu des 30% des cas précédents). Dans ce cas, peu importe le nombre de nœuds, aucune structure ne sera trouvée par la méthode. On obtient donc un cluster source, un cluster cible et un intervalle de temps, et ce avec une variance nulle.

4. Expérimentations sur un jeu de données réelles

Des expérimentations sur un jeu de données réelles ont été menées pour illustrer l'efficacité de la méthode sur un cas pratique.

4.1. *Les vélos en libre-service à Londres*

Ce jeu de données est un enregistrement de tous les emprunts de vélos en libre-service dans les stations de Londres entre le 31 Mai 2011 et le 4 Février 2012. Les données sont disponibles sur le site de TFL[1]. En tout, ce sont 4,8 millions de trajets entre 488 stations qui ont été enregistrés. Les données sont modélisées sous la forme d'un graphe avec des stations de départ (nœuds source), des stations d'arrivée (nœuds cible) et l'heure d'emprunt (timestamp). L'heure d'emprunt est précise à la minute, on a donc 1440 heures d'emprunts différentes (soit 1440 timestamps différents sur les arcs).

4.2. *Le graphe image*

En appliquant la méthode, on obtient 296 clusters source, 281 clusters cible et 5 intervalles de temps. Le calcul du meilleur modèle a été obtenu en 50 minutes avec une utilisation de 4,5GB de ressource mémoire. La plupart des stations sont seules dans leur clusters, la segmentation est donc extrêmement fine mais ne résulte pas d'un surapprentissage. En fait, le nombre de locations est tel que la distribution des arcs est suffisamment fine pour que chacune des stations soit différenciée des autres. Quant à la discrétisation temporelle, le nombre de périodes est tout à fait raisonnable. Elle se décompose de la manière suivante : l'aube (4h12 à 7h05), le matin (7h06 à 9h27), la journée (9h28 à 15h27), le soir (15h28 à 18h16) et enfin la nuit (18h17 à 4h11).
Si obtenir près de 300 clusters est un résultat permettant une étude intéressante au niveau d'un quartier, ça n'en demeure pas moins difficile à interpréter globalement. C'est pourquoi, le graphe image a été simplifié pour permettre son interprétation. Le graphe image est post-traité selon le processus agglomératif décrit en section 2.3. En fusionnant pas à pas les clusters, on arrive à réduire leur nombre à 20 clusters de stations source et cible pour 5 intervalles de temps, tout en maintenant 70% d'informativité.

1. Transport for London, `http://www.tfl.gov.uk`

La Figure 3 permet de mettre en évidence une certaine corrélation géographique dans les clusters de stations alors qu'aucune hypothèse n'a été faite quant à une éventuelle proximité des stations au sein d'un cluster. Une exception pour un cluster (représenté par des cercles blancs) dont les stations se trouvent partagées entre le Sud et le Nord de la ville, correspondant plus précisément aux stations à proximité des gares de Waterloo et de King's Cross qui sont les principales gares de banlieue de Londres. Malgré la distance qui les sépare, il n'est pas étonnant de les retrouver dans un même cluster : on peut facilement imaginer que les gens y ont le même comportement, par exemple prendre un vélo le matin, le déposer dans un quartier d'affaires et faire l'inverse le soir. Le schéma de clustering est asymétrique. Mais pour la plupart des clusters on retrouve une relative symétrie dans les clusters.

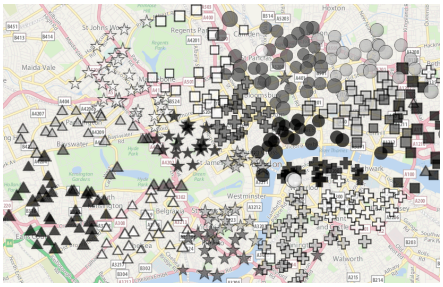


Figure 3 – Clusters de stations source dessinées sur la carte de Londres. Il y a un symbole gris par cluster.

4.3. *Visualisations détaillées*

Dans le but de faire une analyse exploratoire plus détaillée des données, différentes visualisations de l'information sont proposées.

Définition (Information mutuelle entre stations)**.** *Ici, l'aspect temporel est mis de côté afin de se concentrer sur le trafic global entre clusters de stations au cours de la journée. Notons l'information mutuelle entre les partitions des stations source et cible* $MI(C_S, C_T)$ *[COV 06] :*

$$MI(C_S, C_T) = \sum_{c_S, c_T} p(c_S, c_T) \log \frac{p(c_S, c_T)}{p(c_S)p(c_T)}$$

L'information mutuelle est nécessairement positive. Cependant la contribution à l'information mutuelle d'un couple de clusters source/cible peut être positive ou négative suivant que la probabilité jointe observée $p(c_S, c_T)$ est supérieure ou inférieure au produit des probabilités marginales des clusters $p(c_S)p(c_T)$, probabilité attendue en cas d'indépendance. L'utilisation d'une telle mesure permet de quantifier l'excès ou le déficit de trajets entre deux groupes de stations par rapport à la quantité attendue. Ceci est illustré sur par la Figure 4 où les stations en rouge sont les stations où on observe

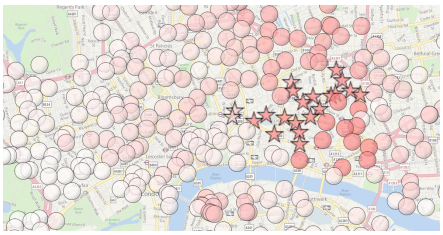


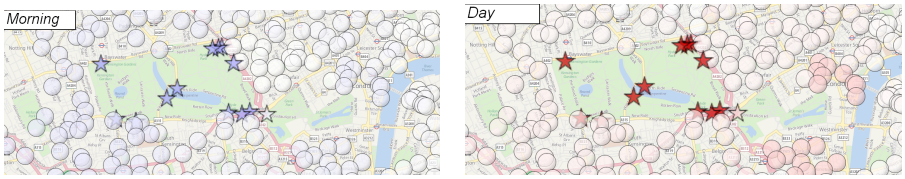
Figure 4 – L'information mutuelle entre le cluster source situé dans la City (stations en forme d'étoiles) et tous les clusters cible. Plus une station est rouge, plus il y a un excès de trafic par rapport à ce qui serait obtenu en cas d'indépendance entre les sources et les cibles.

un excès de trafic et les stations tendant vers le blanc où le trafic observé correspond au trafic attendu. Pour ce cluster source, il n'y a pas de déficit de trafic vers un groupe de stations cible qui apparaitraient en bleu le cas échéant.

Définition (Information mutuelle entre trajets et intervalles de temps)**.** *L'information mutuelle entre couples de stations et périodes de temps $MI[(C_S, C_T), I]$ est calculée afin de mesurer l'évolution des trajets au cours de la journée.*

$$MI[(C_S, C_T), I] = \sum_{c_S, c_T, I_n} p(c_S, c_T, I_n) \log \frac{p(c_S, c_T, I_n)}{p(c_S, c_T)p(I_n)}$$

À l'image de l'information mutuelle entre clusters de stations, cette mesure a pour but de mettre en évidence un excès ou un déficit de trafic entre deux clusters de stations à une période de la journée $p(c_S, c_T, I_n)$ par rapport au trafic habituel observé entre ces deux clusters $p(c_S, c_T)$ et le trafic global sur Londres pendant l'intervalle de temps concerné $p(I_n)$. Pour illustrer cette mesure, on observe sur la Figure 5 un déficit de trafic dans Hyde Park le matin par rapport au trafic habituel et par rapport au trafic dans Londres à la même période. On y observe le phénomène inverse pendant la journée, ce qui peut s'interpréter par le fait que les gens qui utilisent les vélos dans Londres ne le font pas le matin (période de pointe) dans le parc mais plutôt dans la journée qui est pour Londres une période plutôt creuse en termes de locations de vélos. Ceci explique de telles contrastes sur la carte. Quant à la nuit, la carte montrerait des nœuds tous blancs, le trafic y étant nul parce-que le parc est fermé.



(a) Matin (b) Journée

Figure 5 – Information mutuelle entre les trajets provenant d'Hyde Park et les périodes de temps matin et journée. Plus une station sera colorée en rouge (resp. bleu), plus elle connait un excès (resp. déficit) de trafic en provenance d'Hyde Park à la période de temps précisée par rapport au trafic habituel dans Hyde Park et au trafic habituel de la période.

5. Conclusion

Dans cet article, l'évolution des structures au sein d'un graphe a été étudiée. Une méthode, nommée MODL, visant à regrouper des nœuds et à discrétiser le temps a été présentée. Cette technique peut s'apparenter à du coclustering dans le sens où le graphe est considéré comme un ensemble d'arcs décrits par trois variables : la source, la cible et le temps. Chacune d'entre elles est simultanément segmentée dans le but d'obtenir une représentation synthétique du graphe par le biais d'une séquence de graphes images qui modélisent une structure sous-jacente statique pour chacun des intervalles de temps. Cette approche est particulièrement intéressante puisqu'elle ne requiert aucun pré-traitement, comme une agrégation des timestamps sur les arcs par

exemple. Les bonnes propriétés du critère ont été illustrées à l'aide d'un jeu de données synthétiques, où l'on observe la fiabilité de la méthode de part sa résistance au bruit et la convergence asymptotique vers la vraie distribution sous-jacente des arcs. Les aspects pratiques ont également été illustrés par une étude du système de vélos en libre-service Londonien. Dans de futures travaux, il est prévu d'étendre l'approche à un nombre plus important de dimensions, en ajoutant par exemple des labels sur les arcs ou d'autres marqueurs temporels comme le jour de la semaine par exemple.

6. Bibliographie

[BOR 88] BORGATTI S. P., « A comment on Doreian's regular equivalence in symmetric structures », *Social Networks*, vol. 10, 1988, p. 265-271.

[BOT 57] BOTT E., *Family and Social Network*, Tavistock, London, 1957.

[BOU 11] BOULLÉ M., « Data grid models for preparation and modeling in supervised learning », *Hands-On Pattern Recognition : Challenges in Machine Learning, vol. 1*, p. 99–130, Microtome, 2011.

[COV 06] COVER T. M., THOMAS J. A., *Elements of information theory (2. ed.)*, Wiley, 2006.

[DHI 03] DHILLON I. S., MALLELA S., MODHA D., « Information-theoretic co-clustering », *KDD '03*, 2003, p. 89–98.

[DOR 04] DOREIAN P., BATAGELJ V., FERLIGOJ A., « Generalized blockmodeling of two-mode network data », 2004.

[GRÜ 07] GRÜNWALD P., *The Minimum Description Length Principle*, Mit Press, 2007.

[HOP 04] HOPCROFT J., KHAN O., KULIS B., SELMAN B., « Tracking evolving communities in large linked networks », *PNAS*, vol. 101, 2004, p. 5249–5253.

[JAY 03] JAYNES E., *Probability Theory : The Logic of Science*, Cambridge Univ. Press, 2003.

[KEM 06] KEMP C., TENENBAUM J., « Learning systems of concepts with an infinite relational model », *In Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[LIN 91] LIN J., « Divergence measures based on the Shannon entropy », *IEEE Transactions on Information theory*, vol. 37, 1991, p. 145–151.

[LOR 71] LORRAIN F., WHITE H., « Structural equivalence of individuals in social networks », *Journal of Mathematical Sociology*, vol. 1, n° 49-80, 1971.

[NAD 57] NADEL S. F., *The Theory of Social Structure*, Cohen & West, London, 1957.

[NOW 01] NOWICKI K., SNIJDERS T., « Estimation and prediction for stochastic blockstructures », *Journal of the American Statistical Association*, vol. 96, 2001, p. 1077–1087.

[REI 07] REICHARDT J., WHITE D. R., « Role models for complex networks », *The European Physical Journal B*, vol. 60, 2007, p. 217-224.

[SHA 48] SHANNON C. E., « A mathematical theory of communication », *Bell system technical journal*, vol. 27, 1948.

[WHI 76] WHITE H., BOORMAN S., BREIGER R., « Social structure from multiple networks : I. Blockmodels of roles and positions », *Am. J. of Sociology*, vol. 81, n° 4, 1976, p. 730-80.

[WHI 83] WHITE D. R., REITZ K. P., « Graph and semigroup homomorphisms on networks of relations », *Social Networks*, vol. 5, n° 2, 1983.

[XIN 10] XING E. P., FU W., SONG L., « A state-space mixed membership blockmodel for dynamic network tomography », *Annals of Applied Statistics*, vol. 4, n° 2, 2010, p. 535–566.