Faites du bruit pour la détection de communautés consensuelles (mais pas trop) !

Antoine Huchet, Jean-Loup Guillaume, Yacine Ghamri-Doudane

Faites du bruit pour la détection de communautés consensuelles (mais pas trop) !

Antoine Huchet[1†], Jean-Loup Guillaume[1] et Yacine Ghamri-Doudane[1]

[1] *L3i, La Rochelle Université, France*

La détection de communautés a montré sa pertinence pour décrire des réseaux réels, qu'ils soient sociaux, technologiques, d'information ou biologiques. La détection de communautés consensuelles permet de résoudre un des problèmes majeurs des algorithmes classiques, qui est celui du non-déterminisme. Cela passe principalement par l'utilisation d'une matrice de consensus, qui synthétise les différents partitionnements possibles. Nous montrons ici que cette matrice ne contient pas que des informations pertinentes, et que ce bruit peut diminuer la qualité des communautés consensuelles obtenues. Nous proposons également des solutions pour filtrer ce bruit et améliorer la performance des algorithmes existants dans l'obtention du consensus.

Mots-clefs : Graphes de terrain, Algorithmique de graphes, Communautés consensuelles, Bruit

1 Introduction

De nombreux réseaux tels que les réseaux sociaux, technologiques, d'informations ou encore issus de la biologie, peuvent être représentés par des graphes. Ces graphes qui modélisent des données issues du monde réel, appelés *graphes de terrain*, présentent des propriétés particulières : distribution des degrés hétérogène, faible distance moyenne, etc. Parmi ces propriétés nous noterons également la répartition des arêtes de ces graphes, laquelle n'est pas homogène : certaines parties du graphe ont tendance à être densément connectées contrairement à d'autres qui le sont peu. Le problème qui consiste à retrouver de telles zones denses s'appelle la *détection de communautés*. Dans un réseau social, cela peut permettre d'identifier des groupes de personnes comme des membres d'une même famille, des amis, ou des collègues de travail.

De nombreux algorithmes de détection de communautés existent. Parmi les principaux, nous pouvons citer Infomap [RAB09] ou la méthode de Louvain [BGLL08]. La majorité de ces algorithmes sont non-déterministes. Ainsi, les communautés trouvées par Louvain dépendent de l'ordre dans lequel les sommets du graphe sont visités. Un tel algorithme peut donc produire différentes partitions. Pour choisir la meilleure il est possible d'utiliser la modularité, qui évalue la qualité d'une partition d'un graphe [NG04]. Cependant, il a été montré qu'il existe de nombreuses partitions structurellement différentes, mais ayant une modularité similaire [GdMC10].

Ainsi, plutôt que de choisir arbitrairement une partition de bonne qualité, il peut être intéressant de combiner les informations de différentes partitions pour en faire des *communautés consensuelles* [LF12]. Cela passe généralement par le calcul d'une matrice de consensus qui contient, pour chaque paire de sommets du graphe, leur fréquence de co-apparition dans une même communauté. Cette matrice représente donc les similarités entre différentes partitions et peut permettre d'identifier un consensus.

Dans ce travail, nous nous intéressons à l'étude de ce consensus, et proposons une double contribution : nous montrons tout d'abord que l'information issue des différentes partitions, donc le contenu de la matrice de consensus, est bruité ; puis, nous montrons comment filtrer une partie de ce bruit, ce qui permet d'améliorer la performance des algorithmes existants.

[†]Ce travail a été partiellement financé par le projet ANR MITIK, Agence Nationale de la Recherche (ANR), PRC AAPG2019.

2 Communautés consensuelles

2.1 Définitions

Un *graphe* $G = (V, E)$ est composé d'un ensemble de sommets V et d'un ensemble d'arêtes $E \subset V \times V$, où $|V| = n$ et $|E| = m$. Les *communautés* forment une partition des sommets d'un graphe. Chaque sommet appartient alors à une et une seule communauté.

Le *coefficient de clustering d'arête* (Edge Clutering Coefficient ou *ECC*) [RCC+04] est une mesure de similarité telle que :

$$ECC(i, j) = \frac{nb\_voisin\_comm(i, j)}{\min(\deg(i), \deg(j))},$$

où $nb\_voisin\_comm(i, j)$ est le nombre de voisins communs aux sommets i et j, et $\deg(i)$ est le degré du sommet i, c'est-à-dire le nombre d'arêtes adjacentes à i. Comme l'ECC mesure le nombre de voisins en commun d'une paire de sommets, elle a été régulièrement utilisée pour distinguer les arêtes intra-communautaire (forte ECC) des arêtes entre communautés (faible ECC).

La *Modularité* [NG04] mesure la qualité d'une partition d'un graphe en communautés. Elle compare le nombre d'arêtes intra-communautaires observé, à celui d'un graphe aléatoire ayant le même nombre de sommets, d'arêtes, et la même distribution des degrés. La modularité est utilisée quand la vérité terrain n'est pas connue et également comme fonction objectif dans plusieurs algorithmes de détection de communautés.

Si l'on dispose d'une vérité terrain, l'*information mutuelle normalisée* (Normalised Mutual Information, ou *NMI*) [DDGDA05] peut-être utilisée pour mesurer la similarité entre cette vérité terrain et une partition obtenue par un algorithme. Les graphes de terrain avec une structure communautaire connue sont rares, on utilise donc souvent des graphes générés aléatoirement mais contenant des communautés. Le modèle le plus utilisé est le modèle de Lancichinetti, Fortunato et Radicchi (*LFR*) [LFR08] qui permet de régler finement les propriétés du graphe généré et de ses communautés. Il contient notamment un paramètre μ (variant de 0 à 1) qui représente la proportion d'arêtes reliant un sommet à des sommets hors de sa communauté. Plus μ est élevé, moins les communautés sont marquées, et plus elles sont difficiles à identifier.

2.2 De la matrice de consensus aux communautés consensuelles

La plupart des algorithmes de détection de communautés sont non-déterministes. Si on exécute n_p fois un algorithme de détection de communautés $\mathscr{A}$ sur un graphe G, on obtient autant de partitions (potentiellement) différentes. On définit alors la *matrice de consensus* C, telle que C_{ij} est égal au nombre de fois où i et j ont été placés dans la même communauté par $\mathscr{A}$ au cours des n_p exécutions.

Les communautés consensuelles peuvent ensuite être calculées en construisant le graphe G_C, dont la matrice d'adjacence est C. L'algorithme $\mathscr{A}$ est exécuté n_q fois sur G_C, puis, tant que ces n_q nouvelles partitions ne sont pas identiques, l'opération est répétée, jusqu'à obtenir des communautés consensuelles [LF12]. Une autre approche consiste à fixer un seuil λ, puis à remplacer les entrées $C_{ij} < \lambda$ par 0. Les composantes connexes du graphe G_C ainsi modifié sont alors les communautés consensuelles [SG12].

La matrice de consensus étant de taille $n \times n$, l'opération de remplissage est coûteuse en temps et en mémoire. Certains auteurs ont travaillé à l'optimisation du remplissage de cette matrice, comme l'algorithme de Tandon [TAT+19] ou ECG [PT18], dans lesquels on ne calcule que les entrées C_{ij} pour les arêtes (i, j) du graphe. Ceci permet de réduire la taille de cette matrice à m éléments, le nombre d'arêtes du graphe.

3 Identification et élimination du bruit

L'objectif de notre étude est de montrer que, d'une part, remplir toute la matrice introduit du bruit et limite la qualité des résultats et que, d'autre part, en se limitant au calcul de certains C_{ij}, il est possible de réduire encore plus la complexité de l'opération de remplissage de C.

Pour montrer que l'information contenue dans la matrice de consensus est bruitée, nous prenons un exemple de génération de communautés consensuelles. Plus spécifiquement, nous générons un graphe G et sa structure communautaire, avec le modèle LFR. Ensuite, nous exécutons n_p fois un algorithme $\mathscr{A}$ (ici, Louvain) sur G. Nous ordonnons les paires de sommets de G par ordre décroissant de leur ECC, puis nous remplissons la matrice de consensus C dans cet ordre. Pendant l'opération de remplissage, nous exécutons

régulièrement $\mathscr{A}$ sur G_C pour étudier l'impact du taux de remplissage sur la qualité du résultat (mesurée avec la NMI entre les communautés issues de $\mathscr{A}(G_C)$ et les communautés LFR). La figure 1 présente les résultats pour des graphes LFR à 1000 sommets avec 4 valeurs de μ. Outre la dégradation de la NMI avec l'augmentation de μ, nous observons que la NMI est maximale bien avant que la matrice C ne soit entièrement remplie, et que la qualité décroît si la matrice est trop remplie. Ajouter des valeurs C_{ij} pour des paires (i, j) de faible ECC diminue la qualité globale des communautés consensuelles. Il faut donc être capable d'identifier où se situe le pic de NMI ou, dit autrement, d'identifier la valeur de l'ECC au niveau du pic, valeur au-delà de laquelle il ne faut plus remplir la matrice.

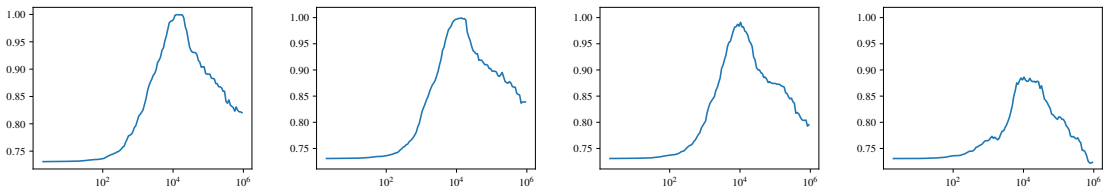

FIGURE 1 : NMI en fonction du nombre d'entrées dans C, $\mu = 0,4$; $\mu = 0,5$; $\mu = 0,6$; $\mu = 0,7$ de gauche à droite.

Dans un contexte réel, la vérité terrain n'est pas connue, on ne peut donc pas utiliser la NMI pour localiser le pic directement. Cependant, nos expériences montrent que la modularité moyenne des n_p exécutions de $\mathscr{A}$ est fortement corrélée à μ, et la figure 2 (gauche) montre que μ est également corrélé à la valeur de l'ECC au niveau du pic. Ces corrélations nous permettent d'améliorer certains algorithmes de détection de communautés consensuelles. Nous nous concentrons ici sur l'algorithme ECG que nous jugeons représentatif des algorithmes de la littérature. Pour cela, nous exécutons n_p fois $\mathscr{A}$ sur G, nous calculons la modularité moyenne des n_p partitions et, via les corrélations précédentes, en déduisons un seuil τ sur l'ECC. Nous ne remplissons que les cellules C_{ij} pour les paires (i, j) vérifiant $ECC(i, j) \geq \tau$. Enfin, nous exécutons ECG(G_C). Nous appelons cette approche ECG_FILTRÉ. A titre de comparaison, nous considérons également la méthode FILTRE_GÉNÉRIQUE, où l'exécution de ECG(G_C) est remplacée par un appel à $\mathscr{A}(G_C)$, où les arêtes de G_C sont pondérées par leur coefficient de consensus.

4 Expériences

Nous avons comparé les algorithmes sur le graphe Football ($n = 115$, $m = 613$, 12 communautés) dans lequel les sommets représentent des équipes de football, les arêtes représentent des parties jouées et les communautés sont les divisions dans lesquelles les équipes évoluent [GN02]. Ici (voir table 1), FILTRE_GÉNÉRIQUE offre les meilleurs résultats ; ECG_FILTRÉ est meilleur qu'ECG mais plus lent ; Tandon offre la NMI la plus faible sans être particulièrement rapide.

Algorithme	NMI	Temps d'exéc.
Filtre_génér.	**0.9246**	**1.07 s**
ECG_Filtré	0.9241	1.99 s
ECG	0.9079	1.10 s
Tandon	0.8976	1.29 s

TABLE 1 : NMI et temps d'exécution des 4 algorithmes sur le graphe Football.

Nous avons également testé sur des graphes LFR avec 10 000 sommets et un paramètre μ variant de 0.4 à 0.7. Nous exécutons les algorithmes ECG_FILTRÉ, FILTRE_GÉNÉRIQUE, ainsi que les algorithmes de Tandon et ECG sur ces graphes[‡]. Nous choisissons $n_p = 16$ exécutions de l'algorithme $\mathscr{A}$ = Louvain. Les NMI et temps d'exécution obtenus sont résumés dans la figure 2 (milieu et droite). L'algorithme ECG_FILTRÉ présente une amélioration de la NMI par rapport à l'algorithme ECG, mais un temps d'exécution supérieur, dû au calcul de l'ECC. Notre filtre parvient donc à filtrer une partie du bruit, et à améliorer la qualité du résultat. L'algorithme FILTRE_GÉNÉRIQUE présente quant à lui une NMI plus importante, toutefois légèrement inférieure à celle de l'algorithme de Tandon, mais permet de travailler sur des graphes plus importants que Tandon grâce à son temps d'exécution bien inférieur.

‡. Toutes les implémentations sont disponibles sur Software Heritage.

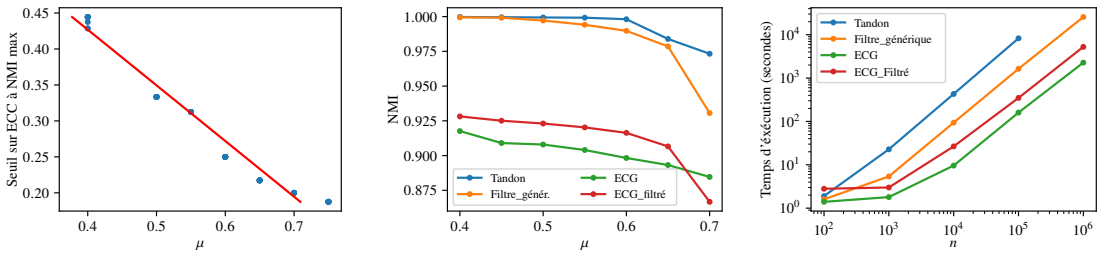

FIGURE 2 : Seuil sur l'ECC en fonction de μ (gauche) ; NMI en fonction de μ, graphes LFR de 10 000 sommets (milieu) ; Temps d'exécution en fonction de la taille du graphe, graphes LFR avec $\mu = 0.6$ (droite).

5 Conclusion et perspectives

Les communautés consensuelles permettent de régler le problème du non-déterminisme dans la détection de communautés. Ces communautés consensuelles sont calculées à l'aide d'une matrice de consensus. Nous avons montré que l'information contenue dans la matrice de consensus est bruitée et nous avons ensuite présenté une manière de filtrer une partie de ce bruit pour améliorer certains algorithmes de la littérature. Les perspectives de ce travail sont multiples : étudier l'impact du filtre sur d'autres algorithmes ; profiter de la stabilité induite par le consensus pour suivre les communautés dans le temps ; ou encore appliquer ces méthodes consensuelles à la détection de communautés dans les graphes multi-couches.

Références

[BGLL08] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics : theory and experiment*, 2008(10) :P10008, 2008.

[DDGDA05] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of statistical mechanics : Theory and experiment*, 2005(09) :P09008, 2005.

[GdMC10] B.H. Good, Y.-A. de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Phys. Rev. E*, 81 :046106, Apr 2010.

[GN02] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12) :7821–7826, 2002.

[LF12] A. Lancichinetti and S. Fortunato. Consensus clustering in complex networks. *Scientific reports*, 2(1) :1–7, 2012.

[LFR08] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4) :046110, 2008.

[NG04] M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69 :026113, Feb 2004.

[PT18] V. Poulin and F. Théberge. Ensemble clustering for graphs. In *International Conference on Complex Networks and their Applications*, pages 231–243. Springer, 2018.

[RAB09] M. Rosvall, D. Axelsson, and C.T. Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1) :13–23, 2009.

[RCC+04] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the national academy of sciences*, 101(9) :2658–2663, 2004.

[SG12] M. Seifi and J.-L. Guillaume. Community cores in evolving networks. In *Proceedings of the 21st International Conference on World Wide Web*, pages 1173–1180, 2012.

[TAT+19] A. Tandon, A. Albeshri, V. Thayananthan, W. Alhalabi, and S. Fortunato. Fast consensus clustering in complex networks. *Physical Review E*, 99(4) :042301, 2019.