

Détection de communautés dans les graphes bipartites

The Anh Dang, Emmanuel Viennet*

*L2TI - Université Paris 13, France
emmanuel.viennet@univ-paris13.fr

Résumé. La recherche de communautés est un problème important pour de nombreux problèmes d'analyse des réseaux sociaux. Nous nous intéressons dans ce travail à la détection de communautés en utilisant uniquement le graphe des relations, dans la lignée des travaux de Newman. Le problème se formule donc comme la recherche de la partition du graphe maximisant un critère de qualité, comme la modularité. Le traitement des graphes bipartites est important pour de nombreuses applications (clients achetant des produits, objets associés à des étiquettes, etc.). La modularité proposée par Newman ne peut pas s'appliquer au cas des graphes bipartites, aussi plusieurs variantes ont été récemment proposées. Dans cet article, nous présentons deux formulations de critères et un algorithme d'optimisation heuristique similaire à celui de Louvain. Des résultats sur des graphes synthétiques et naturels sont présentés et discutés.

1 Introduction

Un graphe bipartite est un graphe comportant deux types de nœuds, tel que chaque arête du graphe connecte deux nœuds de types différents. Nous considérons ici des graphes non orientés, et distinguons les nœuds « gauche » V_X et les nœuds « droits » V_Y (figure 1). Les graphes bipartites sont utilisés pour représenter les relations entre des entités de types différents, comme les clients et les produits dans un système de vente, ou les auteurs et les articles, ou encore les utilisateurs et les étiquettes (*tags*) dans une plate-forme collaborative comme `delicious.com` (ce type d'application peut aussi donner lieu à des graphes tripartites, si l'on considère les relations utilisateurs-étiquettes-produits).

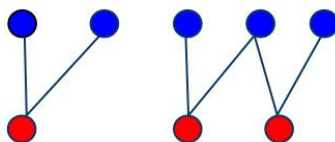


FIG. 1 – Un graphe bipartite.

Un découpage bipartite $C = \{C_k\}_k \cup \{C_l\}_l$ partitionne les nœuds gauche et droits en deux sous-ensembles disjoints : $\cup \{C_k\}_k = V_X$ et $\cup \{C_l\}_l = V_Y$ (figure 2).

Détection de communautés dans les graphes bipartites

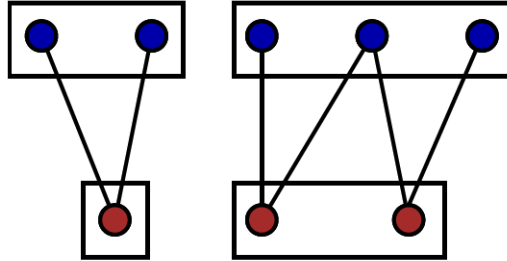


FIG. 2 – *Communautés dans un graphe bipartite : on a ici 4 communautés, chacune est constituées d'un ensemble de nœuds de même type.*

Dans une première partie, nous rappelons la définition du critère de modularité qui permet de quantifier simplement la qualité d'une partition d'un graphe. Nous indiquons comment étendre ce critère au cas des graphes bipartites. Nous présentons ensuite un algorithme heuristique d'optimisation permettant de calculer un partition quasi-optimale, puis concluons par des expérimentations sur de nombreux types de graphes qui permettent de mettre en évidence les qualités des mesures et algorithmes présentés.

2 Modularité bipartite

La modularité est une mesure simple de la qualité de la partition d'un graphe introduite par Newman (2006) dans le cas unipartite. Les partitions avec une grande valeur de la modularité ont une modularité élevée.

Nous présentons différentes définitions proposées pour mesurer simplement la qualité d'une partition dans les graphes bipartites. Les partitions considérées ici distinguent toujours les nœuds « droits » et les nœuds « gauches ». Dans un graphe client/produits, on aura donc des communautés (sous-ensembles) de clients et des communautés de produits (figure 2).

2.1 Modularité de Newman-Girvan

La modularité de Newman Q compare le nombre d'arêtes connectant des nœuds appartenant à la même communauté avec celui que l'ont aurait mesuré sur un graphe aléatoire ayant la même taille :

$$Q = \sum_{C_i \in C} \left(\frac{\|C_i \rightarrow C_i\|}{|E|} - \frac{\|C_i \rightarrow V\|^2}{|E|^2} \right) \quad (1)$$

où $\|V_1 \rightarrow V_2\|$ dénote le nombre d'arêtes connectant des nœuds de l'ensemble V_1 à des nœuds de l'ensemble V_2 .

Dans un graphe bipartite, on n'a par définition aucune arête entre deux nœuds de la même communauté. L'équation (1) s'écrit donc :

$$Q = - \sum_{C_i \in C} \left(\frac{\| C_i \rightarrow V \|^2}{|E|^2} \right)$$

Q perd alors son sens et est maximisée en plaçant chaque nœuds dans une communauté différente : la modularité de Newman n'a pas de sens pour les graphes bipartites.

2.2 Modularité de Murata

Murata (2009) étend la notion de modularité pour les graphes bipartites. Au lieu de considérer la densité des liens intra-communautaires, il compare le nombre de liens entre chaque paire de communautés (droite, gauche) à sa valeur attendue dans un modèle aléatoire. Si deux communautés sont très liées, elles forment probablement partie d'une bonne partition. Avec les notations précédentes, la modularité de Murata s'écrit :

$$\bar{Q} = \sum_{C_k, C_l \in C} \delta(l, \arg \max_{C_m \in C} (\| C_k \rightarrow C_m \|)) \cdot \left(\frac{\| C_k \rightarrow C_l \|}{|E|} - \frac{\| C_k \rightarrow V \| \| C_l \rightarrow V \|}{|E|^2} \right)$$

où δ est le symbole de Kronecker :

$$\delta(i, j) = \begin{cases} 0 & \text{si } i = j \\ 1 & \text{si } i \neq j \end{cases}$$

2.3 Modularité de Suzuki-Wakita

Dans la définition de \bar{Q} , l'utilisation de $\arg \max$ peut poser un problème de stabilité car une communauté d'un côté peut être liée à plusieurs communautés du côté opposé par un nombre très voisin d'arêtes ; il suffit alors de changer quelques liens pour modifier sensiblement la solution.

Suzuki et Wakita (2009) proposèrent d'éviter ce problème en remplaçant le $\arg \max$ par une pondération, définie par la proportion d'arêtes de C_k à C_l ($\| C_k \rightarrow C_l \|$) par rapport au degré de C_k ($\| C_k \rightarrow V \|$). La modularité est alors une somme des modularités des paires de communautés :

$$\hat{Q} = \frac{1}{2} \sum_{C_k, C_l \in C} \frac{\| C_k \rightarrow C_l \|}{\| C_k \rightarrow V \|} \cdot \left(\frac{\| C_k \rightarrow C_l \|}{|E|/2} - \frac{\| C_k \rightarrow V \| \| C_l \rightarrow V \|}{(|E|/2)^2} \right)$$

Cette approche est en principe plus adaptée aux réseaux bipartites dans lesquels les communautés sont connectés à plusieurs communautés de l'autre côté (*multi-facet*).

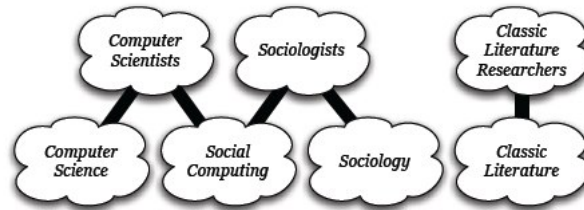


FIG. 3 – Un exemple de réseau multi-facet (image extraite de Suzuki et Wakita (2009)). Dans ce réseau, des chercheurs (en haut) s'intéressent à des champs disciplinaires (en bas). Certains d'entre eux sont pluridisciplinaires sont associés à plusieurs champs. Les communautés correspondantes d'un type sont donc liées à plusieurs de l'autre type.

3 Algorithmes d'optimisation

3.1 Algorithme de Louvain pour les graphes bipartites

L'optimisation de la modularité Q est un problème difficile pour lequel il n'existe pas de méthode exacte en temps raisonnable. Plusieurs approches heuristiques ont été proposées ces dernières années, l'une des meilleures étant celle dite de Louvain proposée par Blondel et al. (2008). Il s'agit d'un algorithme de partition de graphe en communautés basé sur une optimisation gloutonne de la modularité de Newman. L'algorithme part d'une situation où chaque nœud est placé dans une communauté séparée (autant de communautés que de nœuds). Ces communautés sont regroupées de manière progressive, il s'agit d'une méthode agglomérative. L'algorithme procède en deux phases. Durant la première, on balaye chaque nœud dans un ordre aléatoire et l'on essaie de l'associer à chaque communauté des nœuds voisins : la communauté voisine permettant plus grand gain de modularité absorbe le nœud inspecté. Lorsqu'on n'arrive plus à obtenir de gain de cette façon, on regroupe les nœuds de chaque communauté pour former un nouveau graphe dans lequel chaque nœud est une communauté du graphe initial. Ce processus est itéré jusqu'à stabilisation.

La convergence de ces heuristiques n'est pas théoriquement démontrée, mais elles fonctionnent assez bien en pratique, ont une complexité quasi-linéaire (en fonction du nombre de nœuds du graphe), et permettent typiquement le traitement d'un graphe de 100 millions de nœuds en moins d'une heure sur un PC.

Pour appliquer directement cette méthode aux graphes bipartites, il est nécessaire de calculer les projections pour construire deux graphes unipartites. Partant d'un graphe bipartite $G = (V_X, V_Y, E)$, on obtient deux graphes $G_1(V_X, E_1)$ et $G_2(V_Y, E_2)$, où $(u, v) \in E_1$ ou E_2 si u et v sont connectés au même nœud dans G . On peut aussi introduire des seuils K_c and K_p sur le nombre de connexions communes. On reliera dans le graphe projeté deux nœuds $(u, v) \in E_1$ (resp. E_2) s'ils sont tous deux connectés à au moins K_c (resp. K_p) nœuds communs dans G (figures 4 et 5).

En pratique, l'opération de projection est problématique car elle demande un temps de calcul importants pour les très grands graphes et que les valeurs des seuils K_c and K_p dépendent de l'application visée et doivent être fixés empiriquement.

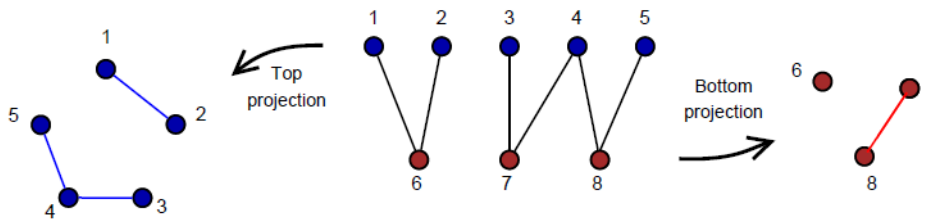


FIG. 4 – Projection d'un graphe bipartite (exemple 1, $K_c = K_p = 1$).

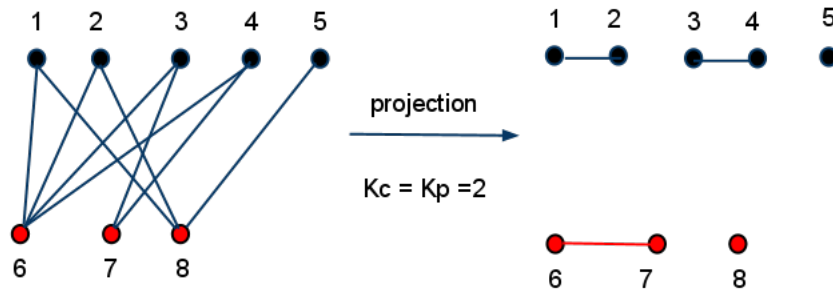


FIG. 5 – Projection d'un graphe bipartite (exemple 2, $K_c = K_p = 2$).

Dans la plupart des applications réelles, on observe certains nœuds avec un très fort degré, comme par exemple un produit que de très nombreux consommateurs achètent. On peut réduire l'impact de ces nœuds (dits « méga-hubs ») en filtrant ceux qui ont un degré supérieur à un seuil W_c (resp. W_p).

Après ces étapes de projection et filtrage, on peut appliquer l'algorithme de Louvain sur les deux graphes projetés pour obtenir les communautés de G . Notons que l'opération de projection entraîne une perte importante d'informations sur le graphe (en particulier, elle empêche évidemment de découvrir des communautés impliquant simultanément des nœuds des deux types).

3.2 Algorithme proposé

Nous proposons d'optimiser directement la modularité de Suzuki-Wakita ou celle de Murata, en suivant une approche semblable à celle de Louvain. A chaque étape de notre algorithme, un nœud est choisi au hasard. On essaie de placer ce nœud dans la communauté de l'un de ses voisins, et on calcule à chaque fois la variation de modularité. Si un gain positif est trouvé, le nœud est associé à la communauté donnant le gain maximum. Cette phase continue jusqu'à ce que l'on ne trouve plus de gains. On peut ensuite, suivant l'idée de Louvain, regrouper les nœuds de chaque communauté pour former un nouveau graphe de communauté et reprendre l'optimisation locale.

Si on part d'une situation où chaque nœud est seul dans sa communauté, l'algorithme est relativement lent, et la solution obtenue comporte souvent un nombre élevé de communautés.

Détection de communautés dans les graphes bipartites

Il est souvent plus efficace de l'initialiser avec les communautés trouvées par Louvain dans les graphes projetés. L'optimisation de la modularité bipartite intervient alors comme une phase de raffinement permettant d'améliorer la partition en prenant en compte les connections bilatérales.

L'efficacité de l'algorithme repose sur l'expression de l'incrément $\Delta\hat{Q}(i, k, k')$ lorsque le noeud i est passé de la communauté k à la communauté k' , qui s'exprime assez facilement car la modularité se décompose en somme des modularités calculées sur les paires de communautés. Ainsi, le gain de la modularité de Suzuki-Wakita quand on passe le noeud i de la communauté k à la communauté k' peut s'exprimer comme :

$$\Delta\hat{Q}(i, k, k') = \sum_{\substack{l \in V \\ l \text{ dans l'autre} \\ \text{côté de } k}} \left(\Delta\hat{Q}(k, l) + \Delta\hat{Q}(k', l) + \Delta\hat{Q}(l, k) + \Delta\hat{Q}(l, k') \right)$$

où

$$\begin{aligned} \hat{Q}(k, l) &= \frac{\|C_k \rightarrow C_l\|}{\|C_k \rightarrow V\|} \cdot \left(\frac{\|C_k \rightarrow C_l\|}{|E|/2} - \frac{\|C_k \rightarrow V\| \|C_l \rightarrow V\|}{(|E|/2)^2} \right) \\ \Delta\hat{Q}(k, l) &= \frac{\|C_k \rightarrow C_l\| - \|i \rightarrow C_l\|}{\|C_k \rightarrow V\| - \|i \rightarrow V\|} \\ &\quad \times \left(\frac{\|C_k \rightarrow C_l\| - \|i \rightarrow C_l\|}{|E|/2} - \frac{(\|C_k \rightarrow V\| - \|i \rightarrow V\|) \|C_l \rightarrow V\|}{(|E|/2)^2} \right) \\ &\quad - \hat{Q}(k, l) \\ \Delta\hat{Q}(k', l) &= \frac{\|C'_k \rightarrow C_l\| + \|i \rightarrow C_l\|}{\|C'_k \rightarrow V\| + \|i \rightarrow V\|} \\ &\quad \times \left(\frac{\|C'_k \rightarrow C_l\| + \|i \rightarrow C_l\|}{|E|/2} - \frac{(\|C'_k \rightarrow V\| + \|i \rightarrow V\|) \|C_l \rightarrow V\|}{(|E|/2)^2} \right) \\ &\quad - \hat{Q}(k', l) \\ \Delta\hat{Q}(l, k) &= \frac{\|C_l \rightarrow C_k\| - \|i \rightarrow C_l\|}{\|C_l \rightarrow V\|} \\ &\quad \times \left(\frac{\|C_l \rightarrow C_k\| - \|i \rightarrow C_l\|}{|E|/2} - \frac{\|C_l \rightarrow V\| (\|C_k \rightarrow V\| - \|i \rightarrow V\|)}{(|E|/2)^2} \right) \\ &\quad - \hat{Q}(l, k) \\ \Delta\hat{Q}(l, k') &= \frac{\|C_l \rightarrow C'_k\| + \|i \rightarrow C_l\|}{\|C_l \rightarrow V\|} \\ &\quad \times \left(\frac{\|C_l \rightarrow C'_k\| + \|i \rightarrow C_l\|}{|E|/2} - \frac{\|C_l \rightarrow V\| (\|C'_k \rightarrow V\| + \|i \rightarrow V\|)}{(|E|/2)^2} \right) \\ &\quad - \hat{Q}(l, k') \end{aligned}$$

Dans le cas de Murata, nous avons des formules assez similaires pour la modularité \bar{Q} :

$$\Delta\bar{Q}(i, k, k') = \Delta\bar{Q}(k) + \Delta\bar{Q}(k') + \sum_{\substack{l \in V \\ 1 \text{ dans l'autre} \\ \text{côté de } k}} \Delta\bar{Q}(l)$$

où :

$$\bar{Q}(k) = \frac{\|C_k \rightarrow C_l\|}{|E|} - \frac{\|C_k \rightarrow V\| \|C_l \rightarrow V\|}{|E|^2} \quad \text{où } l = \arg \max_{C_m \in C} (\|C_k \rightarrow C_m\|)$$

$$\Delta\bar{Q}(k) = \frac{\|C_k \rightarrow C_{l-}\|}{|E|} - \frac{\|C_k \rightarrow V\| \|C_{l-} \rightarrow V\|}{|E|^2} - \bar{Q}(k)$$

$$\Delta\bar{Q}(k') = \frac{\|C_k \rightarrow C_{l+}\|}{|E|} - \frac{\|C_k \rightarrow V\| \|C_{l+} \rightarrow V\|}{|E|^2} - \bar{Q}(k')$$

$$\Delta\bar{Q}(l) = \frac{\|C_l \rightarrow C_{k*}\|}{|E|} - \frac{\|C_l \rightarrow V\| \|C_{k*} \rightarrow V\|}{|E|^2} - \bar{Q}(l)$$

$$l- = \arg \max_{C_m \in C} (\|C_k \rightarrow C_m\|) \quad (\text{après que } i \text{ soit enlevé de } k)$$

$$l+ = \arg \max_{C_m \in C} (\|C'_k \rightarrow C_m\|) \quad (\text{après que } i \text{ soit ajouté à } k')$$

$$l+ = \arg \max_{C_m \in C} (\|C'_k \rightarrow C_m\|) \quad (\text{après que } i \text{ soit ajouté à } k')$$

$$k* = \arg \max_{C_m \in C} (\|C_l \rightarrow C_m\|)$$

4 Données utilisées

Les différentes approches décrites dans ce travail ont été testées sur plusieurs jeux de données synthétiques ou naturels.

4.1 Graphes synthétiques

– Modèle de Barabási-Albert

Réseaux construits selon le modèle d'attachement préférentiel classique Barabasi (2003), dans lequel la probabilité de créer un lien vers un nouveau nœud est proportionnelle au degré.

– Modèle de Suzuki-Wakita

Petit réseau imaginaire décrivant les abonnements de scientifiques Suzuki et Wakita (2009) :

- il y a n nœuds dans trois champs disciplinaires et n pour chaque groupe de revues (voir figure 2.3).
- chaque paire de nœuds des groupes connectés est relié par une arête avec la probabilité p .

Détection de communautés dans les graphes bipartites

– Modèles de Guillaume

Ces réseaux sont générés par deux modèles introduits par Guillaume et Latapy (2004) : modèle bipartite aléatoire et avec attachement préférentiel.

4.2 Réseaux naturels

– Davis Southern Club Woman¹

Jeux de données collecté par Davis et al. dans les années 1930, concernant l'assistance de 18 femmes à 14 événements sociaux.

– Réseau de collaboration scientifique de Newman²

Réseau de co-auteurs, basé sur les prépublications de arXiv (section « matière condensée ») entre 1995 et 1999. Contient 16726 auteurs (nœuds du graphe et 22015 articles (liens).

– Jeux de données Del.icio.us³

Extrait du réseau du site collaboratif Del.icio.us, avec 2000 utilisateurs et 2000 étiquettes (*tags*) et adresses (URLs), à partir duquel on génère deux réseaux bipartites : URL-TAG (67 URLs, 1542 tags) and USER-TAG (2135 utilisateurs, 1291 tags).

– Réseau BAO

Le groupe SAMSE⁴ est l'un des principaux acteurs de la vente de produits de bricolage et de matériaux de construction en France. Il compte des points de vente dans 23 départements du quart sud-est du pays, région dans laquelle il est numéro 1. La Boîte à Outils (BAO) est une filiale spécialisée dans le bricolage qui compte 27 magasins. Nous avons utilisé les historiques d'achats des clients fidèles (titulaires d'une carte de fidélité) récoltés entre 2005 et 2008, qui comptent 43.779 clients et 3.425.048 transactions sur la période 2005-2007, plus 30.784 clients et 1.140.510 transactions en 2008. Les produits achetés sont répartis dans 484 catégories (sous-familles).

	# nœuds droits	# nœuds gauches
Davis Southern Club Woman	14	18
Newman's scientific collaboration network	16726	22015
Delicious URL TAG	67	1542
Delicious USER TAG	2135	1291
BAO network	42146	484

TAB. 1 – Jeux de données réels utilisés pour cette étude.

5 Résultats expérimentaux

Nous avons testé différentes approches : Louvain sur les graphes projetés (optimisation de Q unipartite), notre algorithme optimisant \hat{Q} ou \tilde{Q} sur les données décrites ci-dessus.

1. <http://toreopsahl.com/datasets/#southernwomen>
2. <http://toreopsahl.com/datasets/#newman2001>
3. <http://www.public.asu.edu/~mdechoud/datasets.html>
4. <http://www.groupe-samse.fr/>, <http://www.laboiteaoutils.fr/>

5.1 Optimisation avec l'algorithme de Louvain

La table 2 donne les résultats obtenus par l'algorithme de Louvain sur les différents jeux de données.

Network	Q_left	Q_right	\hat{Q}	\bar{Q}
Barabási-Albert model				
(500, 50)	0,68/13	0,42/5	0,42	0,58
(1000,100)	0,86/26	0,58/8	0,55	0,68
(3000, 150)	0,86/78	0,63/7	0,47	0,6
(5000, 300)	0,82/80	0,72/13	0,5	0,61
Guillaume Model				
Model 1 (1000,2000)	0,31/34	0,3/41	0,005	0,07
Model 2 (1000,3511,p=0,6)	0,56/23	0,65/46	0,01	0,08
Real networks				
Davis Southern Club Woman	0,01/2	0,13/2	0,01	0,33
Newman's scientific collaboration network	0,84/1258	0,87/1274	0,31	0,46
Delicious URL TAG	0,06/3	0,53/14	0,09	0,32
Delicious USER TAG	0,38/14	0,34/35	0,03	0,18
BAO network	0,21/6	0,14/10	0,07	0,29

TAB. 2 – Modularités obtenues par l'algorithme d'optimisation de Louvain travaillant sur les graphes projetés, pour les jeux de données étudiés. Q_{left} (resp. Q_{right}) est la modularité de Newman obtenue par Louvain sur le graphe projeté gauche (resp. droit). On indique à chaque fois le nombre de communautés obtenues (0,68/13 indique que l'on a trouvé 13 communautés et une modularité de 0,68). \hat{Q} est la modularité de Suzuki et \bar{Q} celle de Murata, mesurées sur les partitions obtenues par Louvain.

Observations :

- les valeurs des modularités bipartites \hat{Q} et \bar{Q} sont plus faibles avec cette méthodes qu'avec les approches directes présentées dans la suite de l'article, ce qui est attendu car nous n'optimisons ici que la modularité unipartite sur les graphes projetés.
- Sur les graphes de Barabási-Albert, on obtient un découpage qui semble acceptable, la valeur moyenne de \hat{Q} et \bar{Q} dépassant 0,5.
- En revanche, les graphes générés par le modèle de Guillaume ne se découpent pas : $\hat{Q} \simeq \bar{Q} \simeq 0$.
- Sur le réseau « Davis Southern Club Woman », les modularités sont très faibles, sauf \bar{Q} (0,33).
- Les réseaux unipartites projetés de Newman se découpent bien ($Q > 0,84$) mais les modularités bipartites \hat{Q} et \bar{Q} sont inférieures à 0,5.
- Les réseaux BAO et Delicious ne se découpent pas bien (faibles valeurs de Q , \hat{Q} et \bar{Q}).
- Pour tous les réseaux, le temps de calcul ne dépasse pas une minute sur un PC de bureau.

5.2 Optimisation de Suzuki-Wakita (\hat{Q})

Les résultats obtenus lorsqu'on optimise directement la modularité de Suzuki-Wakita avec l'algorithme proposé en section 3.2 sont détaillés en table 3.

Comme indiqué plus haut, il y a deux façons de procéder : soit partir des nœuds isolés (comme Louvain), soit partir des communautés trouvées par Louvain et raffiner en optimisant \hat{Q} .

Observations :

- Les valeurs des modularités bipartites \hat{Q} et \bar{Q} sont nettement supérieures à celles trouvées en exploitant les graphes projetés (voir table 2).
- Comme précédemment, les graphes de Barabási-Albert se découpent bien (\hat{Q} and \bar{Q} supérieures à 0, 8).
- Le réseau BAO se découpe un peu mieux mais les valeurs de modularité obtenues restent faibles.
- En partant du résultat de Louvain (« Start with LV »), on obtient de moins bonnes valeurs des modularités bipartites qu'en partant de rien. En revanche, cela permet d'arriver à une solution comportant beaucoup moins de communautés et réduit considérablement le temps de calcul.

5.3 Optimisation de Murata (\bar{Q})

L'optimisation directe de la modularité bipartite de Murata (\bar{Q}) est plus coûteuse en calculs que celle de (\hat{Q}) car il faut recalculer plus de termes pour trouver le maximum (utilisation de argmax). Les résultats sur quelques petits réseaux sont présentés dans la table 4 ; ils ne semblent pas meilleurs que ceux obtenus en optimisant \hat{Q} .

6 Discussion

Nos principales conclusions sont :

- l'algorithme d'optimisation converge rapidement, toujours en une ou deux phases.
- les découpages trouvés par Louvain sur graphes projetés ont des modularités bipartites (\hat{Q} et \bar{Q}) plus faibles que celles obtenues par notre algorithme.
- si notre algorithme est initialisé à partir du résultat de Louvain, \hat{Q} et \bar{Q} sont légèrement moins bons qu'à partir d'une initialisation standard, mais avec un nombre bien plus faible de communautés et gain de temps de calcul.
- sur les grands graphes (plus de 10^5 nœuds), notre algorithme est plus lent que Louvain, car les incréments de \hat{Q} et \bar{Q} sont plus complexes. Les complexités des deux algorithmes sont en $O(n)$, où n est le nombre de nœuds, mais le calcul de $\Delta\hat{Q}$ demande beaucoup plus d'opérations (et chaque modification demande des mises à jour plus coûteuses).
- pour les graphes à appartenance multiples (*multi-facets*), on vérifie bien que la modularité de Murata \bar{Q} est la plus adaptée.

Les techniques décrites dans cet article pourraient en particulier s'appliquer aux systèmes de filtrage collaboratif, utilisés dans les moteurs de recommandation pour effectuer des prévisions sur les intérêts des utilisateurs (clients) sur la base du comportement de clients similaires. La plupart des méthodes actuelles de filtrage collaboratif, qui ont fait l'objet d'une abondante

Network	\hat{Q}	\bar{Q}	Q_left	Q_right
Barabási-Albert model				
(500,50)	0,82/87	0,89	0,86/44	0,001/43
Start with LV	0,68/11	0,83	0,96/6	0,5/5
(1000, 100)	0,85/179	0,9	0,84/91	0,01/88
Start with LV	0,82/16	0,91	0,97/8	0,80/8
(3000, 150)	0,9/294	0,94	0,91/151	-0,001/143
Start with LV	0,8/14	0,9	0,98/7	0,82/7
(5000, 300)	0,9/570	0,94	0,91/290	0,007/280
Start with LV	0,86/26	0,93	0,96/13	0,73/13
Suzuki-Wakita model ($n = 200, p = 0, 1$)				
Amalgamative clustering	0,32	0,66	0,56	0,42
Symmetric clustering	0,37	0,70	0,56	0,42
Shared clustering	0,38/7	0,54	0,56/3	0,42/4
Guillaume model				
Model 1 (1000,2000)	0,28/1878	0,34	0,003	0,08
Start with LV	0,28/29	0,54	0,06/11	0,21/18
Model 2 (1000,3511,p=0,6)	0,49/796	0,61	0,02/143	0,4/653
Start with LV	0,54/31	0,75	0,6/12	0,6/19
Real networks				
Davis Southern Club Woman	0,21/13	0,41	0,08/7	0,17/6
Start with LV	0,04/4	0,38	0,38/2	0,44/2
Newman's scientific collab. network	0,7/6813	0,8	0,55/3523	0,47/3290
Start with LV	0,83/1339	0,91	0,85/787	0,8/552
Delicious URL TAG	0,46/259	0,59	-0,01/66	0,36/193
Start with LV	0,32/11	0,62	0,37/3	0,64/8
Delicious USER TAG	0,47/553	0,65	0,29/287	0,26/266
Start with LV	0,33/90	0,58	0,32/42	0,31/48
BAO network	0,08/173	0,16	-2.8e-05/156	0,2/17
Start with LV	0,09/6	0,35	0,37/10	0,25/235

TAB. 3 – Résultats de l'optimisation de la modularité de Suzuki-Wakita \hat{Q} . On indique la modularité obtenue (\hat{Q} est optimisée, les autres sont mesuré sur la solution) et le nombre de communautés (seules celles comportant plus d'un nœud sont prises en compte).

Détection de communautés dans les graphes bipartites

Network	\hat{Q}	\bar{Q}	Q_left	Q_right
(500,50)	0,83	0,9/74	0,86/37	0,06/37
(1000,100)	0,85	0,91/160	0,85	0,05
(3000,150)	0,9	0,94/270	0,91 /135	0,01/135
(5000,300)	0,90	0,94/510	0,91/255	0,05/255
Delicious URL TAG	0,48	0,7/60	0,05/30	0,58/30
Delicious USER TAG	0,45	0,67/503	0,46/259	0,37/244

TAB. 4 – Résultats obtenus en optimisant la modularité bipartite de Murata (\bar{Q}).

littérature, construisent des modèles prédictifs à partir de l'ensemble des données (transactions) clients/produits. Si on construit le graphe bipartite clients/produits, on peut le découper en communautés d'utilisateurs et de produits et construire un modèle de recommandation dans chaque segment. Nous cherchons actuellement à combiner ces modèles locaux pour construire un système de recommandation plus performant.

Ce travail a été partiellement financé par les projets ANR Ex DEUSS et DGCIS CEDRES.

Références

- Barabasi, A.-L. (2003). *How Everything Is Connected to Everything Else and What It Means*. Plume.
- Blondel, V. D., J.-L. Guillaume, R. Lambiotte, et E. Lefebvre (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment* 2008(10), P10008 (12pp).
- Guillaume, J.-L. et M. Latapy (2004). Modularities for bipartite networks. *Information Processing Letters* 90(6), 215–221.
- Murata, T. (2009). Modularities for bipartite networks. In *HT '09 : Proceedings of the Twentieth ACM Conference on Hypertext and Hypermedia*, New York, NY, USA. ACM.
- Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103(23), 8577–8582.
- Suzuki, K. et K. Wakita (2009). Extracting multi-facet community structure from bipartite networks. In *CSE '09 : Proceedings of the 2009 International Conference on Computational Science and Engineering*, Washington, DC, USA, pp. 312–319. IEEE Computer Society.