

# Visualisation interactive multi-échelle des grands graphes : application à un réseau de blogs

Massoud Seifi\*, Jean-loup Guillaume\*  
Matthieu Latapy\*, Bénédicte Le Grand\*

\* [prénom].[nom]@lip6.fr  
Université Pierre et Marie Curie, LIP6, CNRS  
<http://www.complexnetworks.fr>

**Résumé.** De nombreux réseaux du monde réel peuvent être modélisés par des grands graphes. Réduire la complexité d'un graphe de manière à ce qu'il puisse être facilement interprété par l'œil humain est une aide précieuse pour comprendre et analyser ce type de données. Nous proposons une méthodologie de visualisation interactive multi-échelle de grands graphes basée sur une classification des sommets qui nous permet de représenter ces graphes de manière lisible et interprétable. Nous appliquons notre méthodologie à un réseau de blogs francophones.

## 1 Introduction

De nombreux réseaux d'interaction peuvent se représenter par des graphes, dans différents contextes comme l'informatique (réseaux de pages web, d'échanges pair-à-pair), la sociologie (réseaux de collaboration), la biologie (interactions protéiniques), la linguistique (synonymies : il y a eu plusieurs études sur les graphes de synonymie en linguistique pour essayer de détecter automatiquement quels mots sont synonymes d'autres.), etc. Une visualisation lisible et intuitive de ces graphes peut nous aider à comprendre la structure globale des données sous-jacentes. Cependant, les outils actuels de visualisation de graphes sont généralement limités à des ensembles de données de quelques milliers de sommets et sont donc incapables de traiter de très grands graphes. Or, cette limite ne permet pas d'appréhender la plupart des grands graphes que l'on rencontre dans le monde réel, appelés "graphes de terrain".

Afin de pallier ce problème, nous proposons ici de précalculer une structure multi-échelle pour le graphe à visualiser et de n'afficher que des groupes de sommets et les liens entre ces groupes au lieu d'afficher tous les sommets simultanément. Nous ajoutons de plus des fonctionnalités permettant de naviguer dans cette structure multi-échelle, c'est-à-dire d'éclater des clusters en représentant leurs sous-groupes ou, au contraire, de les fusionner pour obtenir une vision plus macroscopique du graphe. Cette méthodologie nous a permis de traiter les réseaux de très grande taille, notamment un réseau contenant 275 millions de sommets et 16 milliards de liens Latapy et al. (2009). Les données étudiées dans cet article sont beaucoup plus petites mais permettent de valider plus clairement l'approche.

L'article est organisé comme suit : dans la section 2 nous présentons les différentes approches nécessaires à la visualisation hiérarchique de graphes de terrains. Dans la section 3

nous présentons un ensemble de structures de données et les principes de base nécessaires à une gestion efficace de l'interaction. Enfin, nous illustrons notre approche dans la section 4 en l'appliquant à un réseau de blogs francophones, avant de conclure et de présenter les perspectives de ces travaux dans la section 5.

## 2 Contexte

Afin de pouvoir visualiser de grands graphes de manière multi-échelle, il est nécessaire de pouvoir les décomposer en une hiérarchie de groupes afin de représenter ces groupes au lieu de l'ensemble des sommets qu'ils contiennent. Notre objectif n'est pas ici de redevelopper un outil de dessin de graphes, nous utiliserons donc un outil existant auquel nous ajouterons des fonctionnalités pour gérer la structure multi-échelle et permettre de naviguer dans celle-ci.

### 2.1 Approches similaires

La visualisation d'information est un domaine très vaste pour lequel un grand nombre d'algorithmes ont été développés, tels que Frick et al. (1994) ou plus récemment Brandes et Pich (2008).

Ces méthodes ont été intégrées dans plusieurs outils, notamment CGV Tominski et al. (2009), GUESS Adar (2006), Tulip Auber et al. (2007), ASK-GraphView Abello et al. (2006) et Pajek Batagelj et Mrvar (2001).

En particulier des approches associent interaction et navigation, comme Moscovich et al. (2009) qui introduisent deux techniques de navigation dans les grands graphes. L'une, intitulée "link sliding", permet de suivre facilement des liens qui nous intéressent, l'autre "Bring & Go" permet d'approcher les voisins d'un sommet. Malgré tout cette approche s'intéresse plus à l'aspect navigation "locale" que multi-échelle.

Parmi ces approches, nous avons choisi l'outil de visualisation (GUESS) avec son algorithme de dessin GEM Layout, décrit dans la section 2.3.

### 2.2 Construction de la structure multi-échelle

La structure multi-échelle doit rendre compte de la topologie du réseau étudié et l'approche classique est de construire des groupes de sommets aussi liés que possible entre eux, tout en limitant au maximum le nombre de liens entre les groupes. Cette notion intuitive de partition d'un graphe en groupes, ou communautés, est capturée par la modularité, introduite dans Newman et Girvan (2004), qui associe à une partition  $P = \{p_1, \dots, p_k\}$  des sommets d'un graphe  $G$  la valeur définie comme :

$$Q(P) = \sum_{p \in P} \left( \frac{l_p}{L} - \left( \frac{d_p}{2L} \right)^2 \right)$$

où  $L$  représente le nombre de liens du graphe,  $d_p$  la somme des degrés des sommets dans  $p$  et  $l_p$  le nombre de liens à l'intérieur de  $p$ . Donc  $\frac{l_p}{L}$  représente la proportion de liens du graphe à l'intérieur de  $p$  et  $\frac{d_p}{2L}$  la proportion de liens du graphe connectés à  $p$ . Cette valeur est toujours entre  $-1$  et  $1$  et permet de comparer les qualités de deux partitions pour un même graphe. Une bonne modularité est toujours positive et la qualité augmente avec la modularité.

De nombreuses méthodes d'optimisation de la modularité ont été proposées au cours des dernières années avec deux objectifs principaux : l'amélioration de la qualité et la réduction de la complexité du calcul de l'algorithme. Nous avons ici choisi d'utiliser la méthode heuristique de Blondel et al. (2008) qui est beaucoup plus rapide que la majorité des autres approches tout en étant au moins aussi efficace dans la plupart des cas.

Obtenir une partition du graphe n'est pas suffisant, aussi pour obtenir la structure multi-échelle complète, nous avons tout d'abord partitionné le graphe puis réappliqué l'algorithme de partitionnement sur chaque communauté de manière récursive. Ceci nous donne donc une hiérarchie de communautés, sous-communautés, sous-sous-communautés, etc. Tous les niveaux de cette hiérarchie ne sont pas forcément pertinents mais le problème consistant à déterminer ceux qui le sont dépasse le cadre de cet article. Nous calculons donc ici la structure complète en redécomposant les communautés jusqu'à obtenir des communautés à un seul sommet.

### 2.3 Outil de visualisation : GUESS

L'objectif ici n'est pas redévelopper un outil de visualisation de graphes mais plutôt d'étendre un outil existant avec les fonctionnalités de navigation dans la structure multi-échelle.

Les outils de manipulation de graphes sont nombreux mais, parmi ces outils, nous avons utilisé GUESS (The Graph Exploration System) développé en Java et distribué sous licence GPL Adar (2006). Cet outil contient en plus un langage interprété, nommé Gython (similaire à Python avec des commandes spécifiques pour les graphes) combiné à une interface graphique 2D qui nous permet d'interagir avec le graphe de différentes manières. Les fonctions coûteuses sont donc implémentées directement en Java.

### 2.4 Filtrage

Quel que soit le système de visualisation, un nombre trop élevé de sommets ou de liens perturbe la visualisation. On a donc souvent besoin de filtrer les graphes avant de pouvoir les représenter en enlevant un certain nombre de sommets ou de liens. Cela doit bien sûr être effectué en modifiant aussi peu que possible la topologie sous-jacente.

Un bon filtrage a donc pour objectif de réduire la taille du graphe en enlevant le "bruit" et de mettre en évidence des sommets et des liens importants. Les critères de filtrage sont basés sur les propriétés quantitatives ou qualitatives des sommets ou des liens. Une métrique doit donc être donnée ou calculée sur les sommets ou les liens. Dans Huang et al. (2005) une fonction est proposée pour quantifier l'importance de chaque sommet. Plusieurs méthodes sont possibles pour les sommets (selon le degré, la centralité globale, le coefficient de clustering, etc.) et les liens (force, coefficient de centralité, etc.) Boutin (2005). Dans la visualisation multi-échelle, nous pouvons également faire du filtrage de sommets (qui sont des communautés) selon leur taille (nombre de sommets dans la communauté) ou leur densité de liens (i.e. le nombre de liens entre deux communautés divisé par le nombre maximum de liens qui pourraient exister :  $d_l = l_{c_i, c_j} / (C_{c_i} * C_{c_j})$ , où  $l_{c_i, c_j}$  est le nombre de liens entre  $c_i$  et  $c_j$ , et  $C_{c_x}$  est le nombre de sommets dans la communauté  $c_x$ ). De même pour les liens il est possible de faire des suppressions en fonction du poids ou de la densité.

Des cas plus spécifiques ont aussi été proposés, notamment dans Boutin (2005) qui propose une technique de filtrage contextuelle autour d'un focus, ou Auber et al. (2003) qui propose une métrique pour identifier les liens les plus faibles dans un réseau petit-monde.

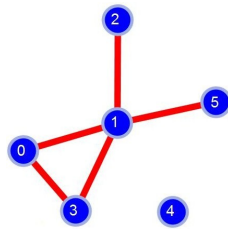
Nous utiliserons plusieurs de ces approches dans la suite.

### 3 Visualisation interactive multi-échelle

#### 3.1 Stockage de graphe

Le stockage de graphe et de la hiérarchie de communautés a une importance primordiale car il conditionne l'utilisabilité de l'outil. En effet, pour que l'interaction soit maximale, il est nécessaire que la visualisation soit rapide et qu'il soit aisé de se déplacer dans la structure hiérarchique. Afin de garantir un temps de réponse le plus court possible, nous avons fait le choix de calculer à l'avance une structure hiérarchique étendue, contenant non seulement les liens à chaque niveau, mais aussi les liens d'appartenance des sommets aux communautés et ce, à chaque niveau.

Afin d'accélérer le chargement en mémoire dans le logiciel, tous les fichiers sont convertis à l'avance dans un format binaire simple. Pour un graphe simple, le fichier contient : le nombre de sommets, les degrés des sommets de manière cumulative, la liste des voisins de chaque sommet, voir le tableau 1.



nombre de sommets :	6					
degrés cumulés :	2	6	7	9	9	10
voisins du sommet 0 :	1	3				
voisins du sommet 1 :	0	2	3	5		
voisins du sommet 2 :		1				
voisins du sommet 3 :		0	1			
voisins du sommet 5 :			1			

TAB. 1: *Stockage de graphe au format binaire*

Chaque lien est stocké dans les deux sens, ce qui augmente la taille du fichier mais accélère l'accès aux voisins de chaque sommet. La taille du fichier est donc  $(1 + \text{nombre de sommets} + 2 * \text{nombre de liens}) * 4$  octets (si chaque entier est stocké sur 4 octets). Les éventuelles pondérations sont toujours stockées dans un fichier annexe afin de pouvoir les modifier sans changer le fichier du graphe lui-même.

#### 3.2 Stockage de la hiérarchie

La hiérarchie est aussi pré-calculée et stockée dans plusieurs fichiers distincts :

- un fichier contenant le nombre de niveaux de l'arbre ainsi que le nombre de sommets/communautés à chaque niveau ;
- un fichier stockant la communauté père de chaque sommet/communauté ;
- un fichier stockant le nombre de fils (sous-communautés) de chaque communauté puis la liste des fils (sous-communautés) de chaque communauté ;
- un fichier contenant le nombre de sommets dans chaque communauté.

- D'autres propriétés de sommets et de liens, notamment des propriétés précalculées nécessaires pour un filtrage spécifique, peuvent être fournies dans un fichier à part. Le système autorise des propriétés uni-dimensionnelles (par exemple, un poids), ou multi-dimensionnelles (par exemple un ensemble de mots clefs).

A partir de ces différents fichiers (construits via la sortie de l'algorithme de détection de communautés), il est possible de construire le graphe complet de la structure hiérarchique contenant un *graphe de communautés* pour chaque niveau de la hiérarchie ainsi que tous les liens de parenté entre tous les sommets de tous les niveaux. Ce graphe peut être créé par l'algorithme 1. Un exemple est donnée sur la Figure 2, pour lequel un lien entre deux sommets génère des liens entre les communautés parentes sur plusieurs niveaux ainsi que des liens entre les sommets/communautés des différents niveaux.

---

**Algorithme 1** *Algorithme de création de la hiérarchie complète*

---

```

1: pour chaque niveau de la hiérarchie  $n$  faire
2:   pour chaque lien du graphe au niveau  $n$  entre  $s_i$  et  $s_j$  faire
3:      $p_i \leftarrow$  le père de  $s_i$ 
4:      $p_j \leftarrow$  le père de  $s_j$ 
5:     tantque  $p_i \neq p_j$  faire
6:       si pas de lien entre  $p_i$  et  $p_j$  alors
7:         créer un lien entre  $p_i$  et  $p_j$ 
8:       fin si
9:       créer un lien entre  $s_i$  et  $p_j$ 
10:      créer un lien entre  $s_j$  et  $p_i$ 
11:       $p_i \leftarrow$  le père de  $p_i$ 
12:       $p_j \leftarrow$  le père de  $p_j$ 
13:    fin tantque
14:  fin pour
15: fin pour

```

---

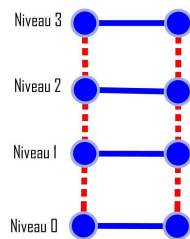


FIG. 1: *Hiérarchie simplifiée pour chaque lien du graphe*

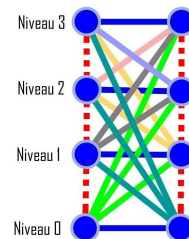


FIG. 2: *Hiérarchie complète pour chaque lien du graphe*

Il faut noter que la structure hiérarchique complète simplifie significativement la navigation mais le temps de calcul de cette hiérarchie peut être élevé. Sur des graphes de taille raisonnable

cela peut être fait sans problème. Par contre, sur de très grands graphes, il est soit possible de pré-calculer la structure complète ce qui simplifie la navigation mais prend du temps, soit de se restreindre à la structure simplifiée. La seule structure simplifiée limite l'interactivité, car il est alors nécessaire de générer les parties manquantes à la volée (voir la Figure 1).

### 3.3 Visualisation

#### 3.3.1 Visualisation multi-échelle

Le concept de base de la visualisation multi-échelle est de ne pas dessiner tout le graphe, ce qui de toute manière est pratiquement impossible pour les très grands graphes et n'apporte que peu d'information même pour des graphes ayant des milliers de sommets. L'objectif est plutôt de dessiner les graphes entre communautés aux niveaux les plus hauts de la structure hiérarchique en réduisant la complexité du graphe par filtrage des sommets et des liens si nécessaire. En outre, la possibilité de navigation dans la hiérarchie nous permet d'explorer le graphe en éclatant ou en fusionnant des communautés.

Comme la Figure 3 l'illustre, à chaque étape plusieurs ensembles de sommets sont remplacés par la communauté correspondante.

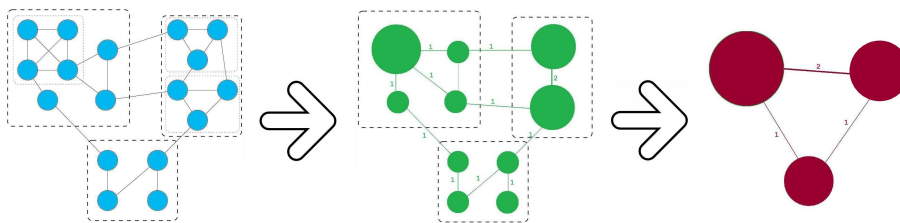


FIG. 3: Remplacement des sommets par des communautés

#### 3.3.2 Taille des sommets et des liens

Plusieurs approches sont possibles pour représenter la taille des communautés et des liens. La contrainte qui se pose est que cette taille est très variable, ce qui nécessite de considérer des fonctions de ces valeurs plutôt que les valeurs elle-mêmes.

Ainsi, de manière assez classique, nous choisissons ici de représenter une communauté sous la forme d'un cercle avec une surface proportionnelle au nombre de sommets qu'elle contient, donc avec un rayon en racine du nombre de sommets dans la communauté.

De même, l'épaisseur d'un lien entre deux communautés correspond au logarithme du poids de ce lien pour les graphes pondérés, où le poids d'un lien entre deux communautés est égal à la somme des poids des liens entre les sommets des communautés qui les contiennent. Pour les graphes non pondérés, on peut considérer la densité en termes de liens.

### 3.4 Interaction et navigation

La navigation et l'interaction sont des fonctionnalités essentielles dans le domaine de la visualisation de l'information. Elles peuvent aider les utilisateurs à révéler des structures spéci-

fiques dans les grands graphes. Nous pouvons éclater une communauté en sous-communautés, ou au contraire fusionner un ensemble de sous-communautés en une seule communauté. Ceci nous permet d'explorer le graphe à différents niveaux de hiérarchie.

## 4 Application à un réseau de blogs

Le corpus analysé dans cet article a été obtenu par l'exploration quotidienne de 3975 blogs pendant 4 mois en 2008/2009. Ces blogs ont été choisis en fonction de leur popularité et de leur activité dans la blogosphère française. Cet ensemble de données présente l'avantage que les blogs aient été classifiés et étiquetés à la main, ce qui nous permettra d'étudier l'impact des traitements différents sur la visualisation. Ce classement manuel fournit trois niveaux d'abstraction : continent, région et territoire, (du plus général au plus spécifique). La couche continent (resp. région, territoire) est composée de 3 (resp. 16, 124) groupes : Loisirs, Société et Individualité. Un exemple d'étiquette de blog sur ces 3 niveaux est "loisirs/culture/musique".

A partir de ces blogs, nous construisons un graphe dans lequel les sommets sont les blogs. Chaque blog est constitué d'un ensemble de billets (posts) qui peuvent contenir des liens hypertextes vers des posts antérieurs (appartenant ou non au même blog). Deux blogs sont reliés dans le graphe s'il existe au moins un post de l'un de ces blogs qui cite un post de l'autre blog via un lien hypertexte. Les liens sont pondérés par le nombre de liens hypertextes.

### 4.1 La structure hiérarchique du graphe de blogs

Afin de disposer d'une structure hiérarchique pour ce graphe, nous avons utilisé l'algorithme de détection de communautés décrit précédemment. L'algorithme a produit une structure à 7 niveaux contenant au niveau le plus élevé 61 communautés, voir le tableau 2.

niveau	nombre de sommets	nombre de liens
0 (le graphe)	3975	18024
1	3946	17957
2	3398	15942
3	1718	10537
4	738	5891
5	269	1735
6	61	129

TAB. 2: Nombre de sommets et nombre de liens à chaque niveau de la structure multi-échelle pour le réseau de blogs

### 4.2 Visualisation du graphe de blogs

Le principal avantage des données du corpus étudié est que les blogs ont été classifiés et étiquetés manuellement, ce qui permet de distinguer visuellement les différents blogs, par exemple en utilisant des couleurs. Nous avons donc choisi de représenter les communautés avec

une couleur(une forme géométrique) liée au classement manuel des sommets : rouge(cercle), vert(carré) et bleu(losange) pour respectivement société, individualité et loisirs.

L'approche la plus classique pour affecter une couleur à une communauté consisterait à représenter les couleurs avec des dégradés en fonction de la couleur des nœuds qu'elles contiennent, mais les résultats obtenus pour des communautés mixtes seraient peu interprétables. Nous avons donc choisi d'utiliser une couleur vive pour les communautés contenant plus de  $x\%$  de sommets du type majoritaire, une couleur claire pour celles contenant entre  $y\%$  et  $x\%$  et du gris pour celles contenant moins de  $y\%$ .

La Figure 4 représente la distribution cumulée des groupes majoritaires dans chaque communauté. Nous pouvons voir qu'il y a très peu de communautés ayant moins de 50% de sommets du groupe majoritaire, nous choisissons donc  $y = 50\%$ . De même, la forte proportion de communautés très ciblées pour lesquelles la grande majorité des sommets sont du même type nous incite à prendre une valeur de  $x$  très élevée, nous avons choisi  $x=90\%$ . Les Figures 6 à 9, illustrent le graphe des blogs aux différents niveaux de hiérarchie avec les couleurs associées, sans aucun filtrage.

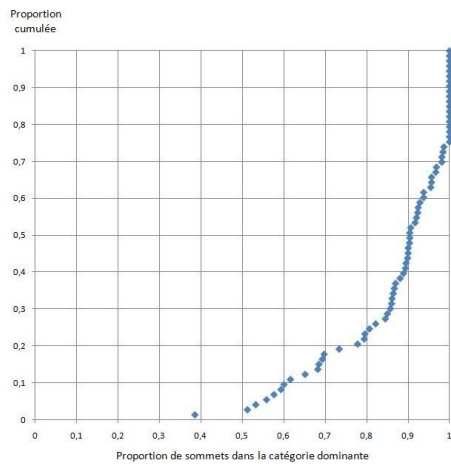


FIG. 4: Distribution de la proportion de la catégorie majoritaire dans les communautés

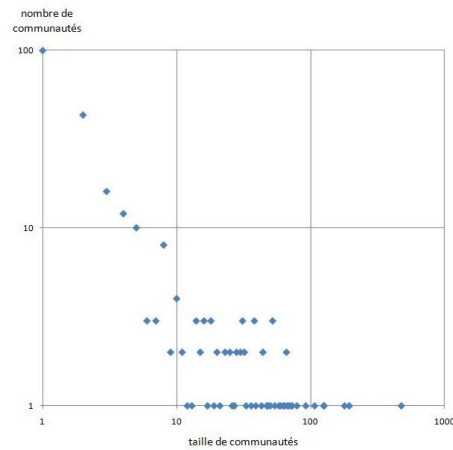


FIG. 5: Distribution de la taille des communautés

#### 4.2.1 Filtrage

La présence d'une grosse communauté grise au niveau 6 (voir la Figure 9) montre que le niveau plus haut n'est pas forcément le plus intéressant même s'il contient moins de sommets et de liens. C'est la raison pour laquelle nous avons décidé d'explorer un niveau plus bas (niveau 5), dans lequel la grosse communauté grise n'existe plus (voir la Figure 8). Puis, nous avons décidé de filtrer en fonction de la taille des communautés afin d'alléger le dessin. Afin de choisir la valeur du seuil de filtrage nous avons utilisé la distribution de taille des communautés (voir la Figure 5). Nous avons décidé d'éliminer les communautés dont la taille est inférieure à 10. La Figure 10 illustre l'efficacité de ce filtrage. Le dessin est beaucoup plus lisible et on peut remarquer que les trois couleurs sont bien séparées après ce filtrage.



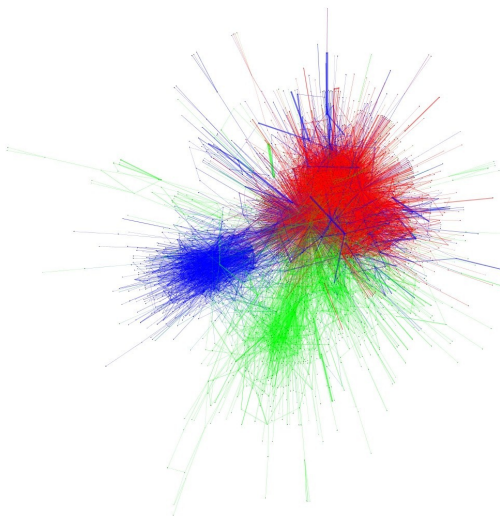


FIG. 6: *Graphe des blogs au niveau 0*

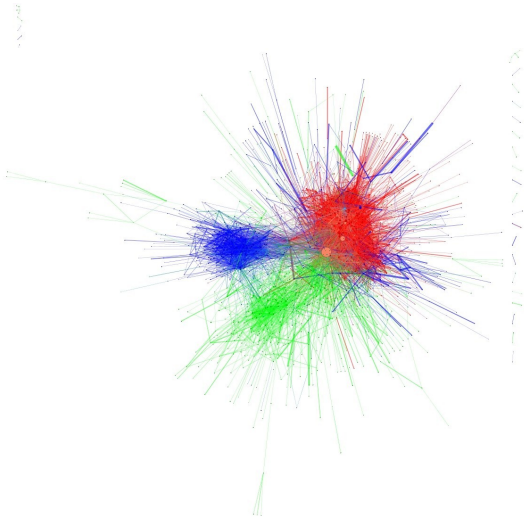


FIG. 7: *Graphe des blogs au niveau 3*

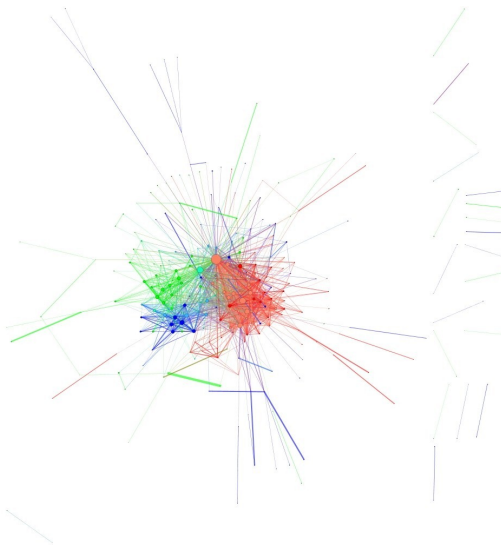


FIG. 8: *Graphe des blogs au niveau 5*

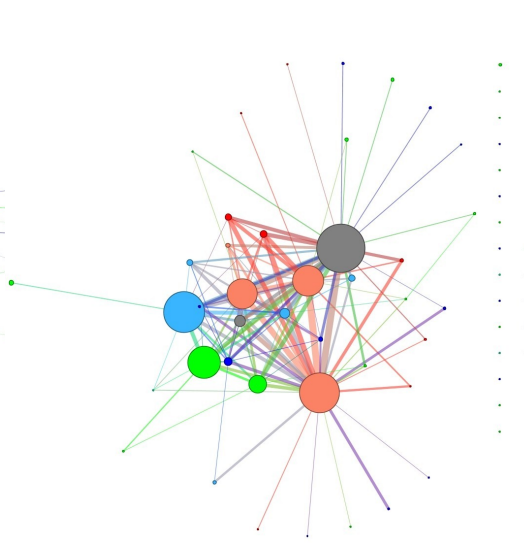


FIG. 9: *Graphe des blogs au niveau 6*

## Visualisation interactive multi-échelle de grands graphes

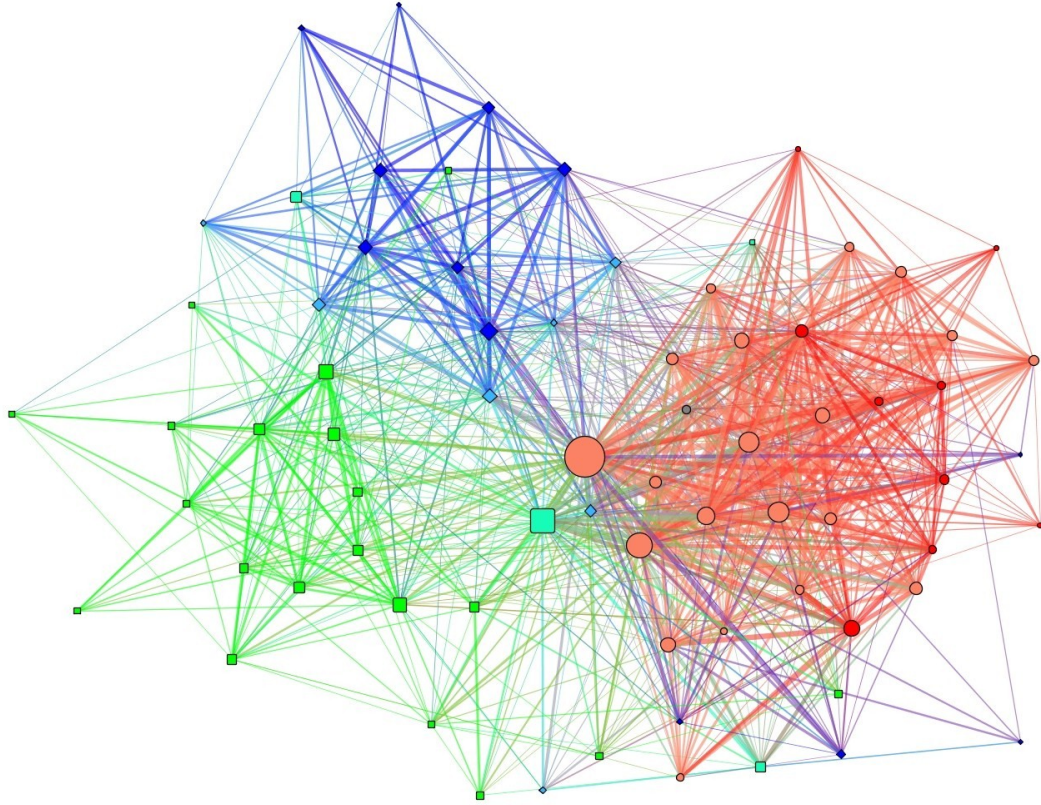


FIG. 10: *Graphe des blogs au niveau 5 - Après filtrage de communautés de petite taille*

### 4.2.2 Navigation

La Figure 11 illustre la fonctionnalité de l'interaction dans notre méthodologie. En haut à gauche, bien que la majorité des communautés d'une même couleur soient proches, il est possible d'identifier quelques communautés qui sont plus éloignées de leur groupe. En particulier, nous avons isolé une communauté bleue (loisirs) plus proche des communautés rouges (société). Une vue éclatée de cette communauté (en haut à droite) ainsi qu'un zoom (en bas à gauche) a été présenté. Il s'avère que cette communauté ne contient presque que des blogs classés dans la catégorie "football".

Une recherche par mot-clé sur les communautés relatives au football (voir en bas à droite) montre qu'il s'agit essentiellement de communautés de couleur rouge, ce qui tend à montrer que la thématique "football" aurait dû être associée à la catégorie Société plutôt qu'à la catégorie Loisirs.

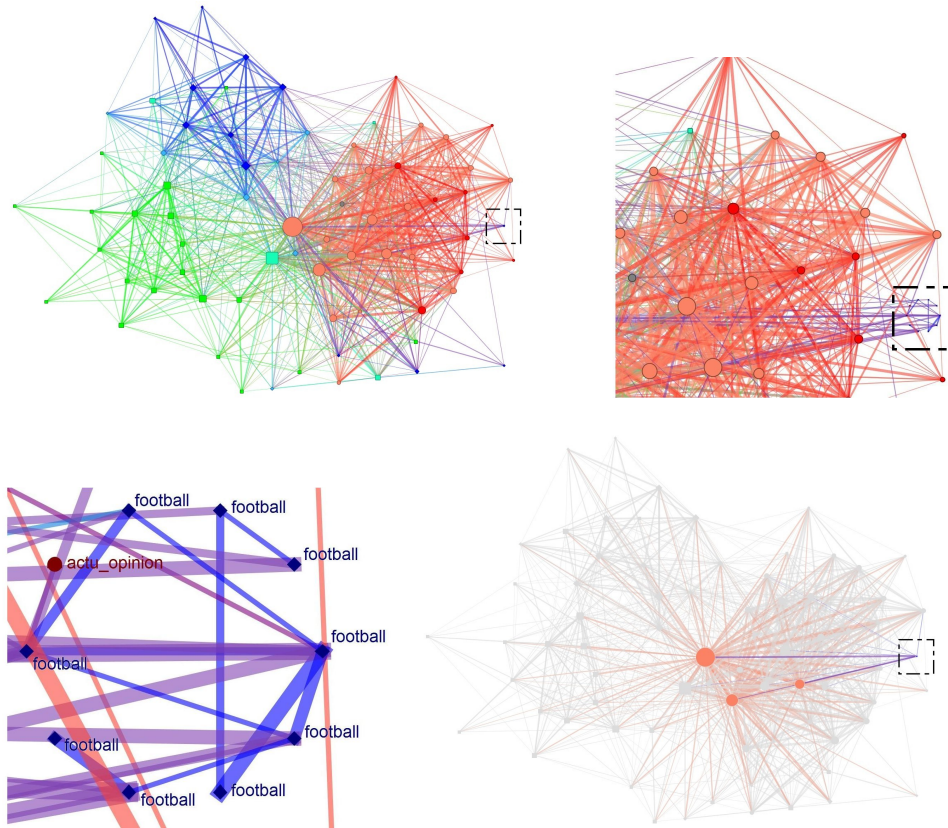


FIG. 11: Graphe des blogs au niveau 5 (en haut à gauche); éclatement d'une communauté mal placée (en haut à droite); zoom sur cette communauté (en bas à gauche). L'ensemble des communautés contenant des blogs sur le football sont représentés en bas à droite.

### 4.3 Interprétation

Les différentes visualisations vérifient le bon fonctionnement de l'algorithme de détection de communautés que nous avons exécuté sur le réseau de blogs, en particulier une fois que nous avons éliminé les petites communautés qui gênent le dessin. L'intérêt du filtrage se voit quand on compare la Figure 8 à la Figure 10, qui est beaucoup plus lisible. Nous avons constaté que les blogs qui parlent de l'individualité peuvent se diviser en deux parties : une partie liée à loisirs et d'autre partie liée à la société.

Grâce au système d'interaction permettant de passer d'un niveau d'abstraction à un autre, nous avons pu explorer plusieurs sous-communautés, notamment une communauté visiblement éloignée des communautés proches et nous avons découvert qu'il s'agissait de blogs provenant d'une catégorie précise (football en l'occurrence) qui est plutôt liée à la société qu'aux loisirs.

## 5 Conclusion

La visualisation graphique est un bon moyen pour faciliter la compréhension de quantités importantes de données abstraites. Elle peut en particulier attirer notre attention sur ce qui n'est pas au centre. En revanche, les outils actuels pour la visualisation de graphes sont limités à des ensembles de données de taille moyenne et sont incapables de traiter de très grands graphes. De plus, l'œil humain a du mal à distinguer les graphes complexes.

Afin de résoudre ce problème, nous avons proposé une méthodologie de visualisation interactive multi-échelle qui peut révéler la structure globale de données. Elle est basée sur le calcul d'une structure hiérarchique de communautés et la possibilité de naviguer dans cette structure en éclatant/fusionnant les communautés à différents niveaux afin d'arriver à une vue intéressante.

Plusieurs problèmes se posent, notamment que les niveaux les plus hauts comportent souvent une composante connexe principale à noyau très dense d'où il est difficile d'extraire des motifs pertinents. Nous avons utilisé plusieurs filtrages qui peuvent nous aider à réduire la complexité de ces graphes.

Nous avons en particulier appliqué notre méthodologie à l'étude d'un réseau de blogs qui avait été étiqueté manuellement. Cela nous a permis de vérifier la qualité de l'algorithme de détection de communauté choisi, de valider le concept de visualisation interactive multi-échelle, ainsi que de détecter des cas où la classification manuelle semble peu satisfaisante.

Une perspective de ce travail est de trouver des techniques de filtrage plus efficaces ainsi que des méthodes d'évaluation de la lisibilité de graphe aux différents niveaux hiérarchiques. L'impact de la structure hiérarchique doit aussi être étudié plus en profondeur.

## Références

- Abello, J., F. van Ham, et N. Krishnan (2006). Ask-graphview : A large scale graph visualization system. *IEEE Trans. Vis. Comput. Graph.* 12(5), 669–676.
- Adar, E. (2006). Guess : a language and interface for graph exploration. In R. E. Grinter, T. Rodden, P. M. Aoki, E. Cutrell, R. Jeffries, et G. M. Olson (Eds.), *CHI*, pp. 791–800. ACM.
- Auber, D., Y. Chiricota, M. Delest, J.-P. Domenger, P. Mary, et G. Melançon (2007). Visualisation de graphes avec tulip : exploration interactive de grandes masses de données en appui à la fouille de données et à l'extraction de connaissances. In M. Noirhomme-Fraiture et G. Venturini (Eds.), *EGC*, Volume RNTI-E-9 of *Revue des Nouvelles Technologies de l'Information*, pp. 147–156. Cépaduès-Éditions.
- Auber, D., Y. Chiricota, F. Jourdan, et G. Melançon (2003). Multiscale visualization of small world networks. In *INFOVIS*. IEEE Computer Society.
- Batagelj, V. et A. Mrvar (2001). Pajek - analysis and visualization of large networks. In *Graph Drawing*, pp. 477–478.
- Blondel, V. D., J.-L. Guillaume, R. Lambiotte, et E. Lefebvre (2008). Fast unfolding of community hierarchies in large networks. *CoRR abs/0803.0476*.  
Français

- Boutin, F. (28/11/2005). *Filtrage, partitionnement et visualisation multi-échelles de graphes d'interactions à partir d'un focus*. Ph. D. thesis, Université Montpellier II - Sciences et Techniques du Languedoc.
- Brandes, U. et C. Pich (2008). An experimental study on distance-based graph drawing. In I. G. Tollis et M. Patrignani (Eds.), *Graph Drawing*, Volume 5417 of *Lecture Notes in Computer Science*, pp. 218–229. Springer.
- Frick, A., A. Ludwig, et H. Mehldau (1994). A fast adaptive layout algorithm for undirected graphs. In R. Tamassia et I. G. Tollis (Eds.), *Graph Drawing*, Volume 894 of *Lecture Notes in Computer Science*, pp. 388–403. Springer.
- Huang, X., P. Eades, et W. Lai (2005). A framework of filtering, clustering and dynamic layout graphs for visualization. In V. Estivill-Castro (Ed.), *ACSC*, Volume 38 of *CRPIT*, pp. 87–96. Australian Computer Society.
- Latapy, M., C. Magnien, R. Fournier, et M. Seifi (2009). Maps of paedophile activity.
- Moscovich, T., F. Chevalier, N. Henry, E. Pietriga, et J.-D. Fekete (2009). Topology-aware navigation in large networks. In *CHI '09 : Proceedings of the 27th international conference on Human factors in computing systems*, New York, NY, USA, pp. 2319–2328. ACM.
- Newman, M. E. J. et M. Girvan (2004). Finding and evaluating community structure in networks. *Phys. Rev. E* 69(2), 026113.
- Tominski, C., J. Abello, et H. Schumann (2009). Cgv, an interactive graph visualization system. *Computers & Graphics*.

## Summary

Many real-world networks can be represented as large graphs. Reducing the complexity of a graph so that it can be easily interpreted by the human eye is very helpful to understand and analyze the data. We propose a methodology for interactive multi-scale visualization of large graphs based on a classification of the nodes which allows us to represent the graph in a legible and interpretable way. We apply our methodology to a network of French-speaking weblogs.