

Visualisation matricielle des graphes et manipulation directe de hiérarchies de clusters

Mohammad Ghoniem

Ecole des Mines de Nantes
4, rue Alfred Kastler. B.P. 20722
F-44307 Nantes Cedex 3, France
Mohammad.Ghoniem@emn.fr

Jean-Daniel Fekete

Inria Futurs / LRI
Bat. 490, Université Paris-Sud
F91405 Orsay Cedex, France
Jean-Daniel.Fekete@inria.fr

RESUME

Dans cet article, nous décrivons VisAdj un outil de visualisation de graphes par matrices d'adjacence. Cette technique permet d'afficher toutes les arêtes du graphe sans occlusion et se prête à sa manipulation directe. Des permutations de lignes et de colonnes sont nécessaires pour faire apparaître les structures sous-jacentes du graphe. Pour les grands graphes où la manipulation individuelle serait trop longue, VisAdj permet le calcul de hiérarchies de clusters et leur manipulation interactive. Ces clusters permettent aussi la représentation agrégée des graphes, facilitant grandement leur compréhension.

MOTS CLES : VisAdj, visualisation, graphes, réseaux, matrice d'adjacence, clustering, agrégation.

ABSTRACT

This paper describes VisAdj, a tool for visualizing graphs as adjacency matrices. Although unfamiliar, this technique displays each link without occlusion and lends itself well to direct manipulation. Row and column permutations are a necessary step towards revealing the underlying structure of a graph. Since individual manipulation of vertices would take too much time in large graphs, VisAdj makes it possible to compute cluster hierarchies and manipulate them interactively. Aggregated views of the graphs can be accessed through these clusters and provide better understanding of their structure.

KEYWORDS : Visualization, networks, graphs, adjacency matrix, clustering, aggregation.

INTRODUCTION

Les diagrammes en nœuds et liens ont longtemps été utilisés pour représenter les graphes. Cette métaphore a été déclinée sous diverses formes tant en 2D qu'en 3D. Toutes ces techniques sont très vite confrontées aux problèmes d'occlusion occasionnés par l'enchevêtrement des arêtes dès que la taille du graphe ou la densité des liens en son sein augmentent (figure 1). Dès lors que la vue d'ensemble du graphe est illisible, il devient difficile pour l'utilisateur d'explorer le graphe et d'interagir avec ses éléments constitutifs.

LA VISUALISATION MATRICIELLE DES GRAPHES

La visualisation matricielle des graphes repose d'un point de vue théorique sur la représentation d'un graphe par sa matrice d'adjacence. Elle offre une alternative intéressante à la métaphore des nœuds et des liens. Un *fish-eye* paramétrable [4] ainsi que l'affichage des labels des arêtes balayées selon la technique du *labeling* excentrique [5] facilitent sensiblement l'exploration du graphe. De plus, dans [3], Jacques Bertin montre qu'il est possible de révéler la structure sous-jacente des réseaux lorsque ceux-ci sont représentés sur une matrice moyennant un jeu de permutations des lignes et des colonnes.

LE CLUSTERING DES GRAPHES

La détection de clusters dans un réseau est une tâche centrale qui permet de déceler sa structure sous-jacente, de déterminer les groupes d'acteurs et de rationaliser les liens qui les rassemblent. Il existe une littérature abondante sur les techniques de clustering issues de disciplines très variées. Une revue assez complète par Berkhin [2] permet d'en obtenir un bon aperçu. Dans le cadre de nos travaux, nous avons implémenté un algorithme hiérarchique agglomératif. Il a l'avantage de produire un clustering satisfaisant assez rapidement (figures 2 et 3). Notre système est extensible à diverses métriques et algorithmes de clustering et pourra à ce titre servir de base de comparaison entre divers algorithmes de clustering. C'est pourquoi nous avons porté nos efforts sur la visualisation et la manipulation directe de hiérarchies de clusters indépendamment de la manière de les produire.

LA VISUALISATION ET LA MANIPULATION DES HIÉRARCHIES DE CLUSTERS

Nous avons implémenté un composant qui sert aussi bien à la visualisation qu'à la manipulation des hiérarchies de clusters. Il s'agit d'une visualisation compacte appelée « arbre à glaçons » ou, *icicle plot* [1]. Cette visualisation a l'avantage de conserver un sens dominant facilitant l'orientation dans l'arborescence. Elle peut être enrichie par l'application de divers patrons de coloration et de pondération qui traduisent des données topologiques de l'arborescence ou des données de l'utilisateur (figure 1 (a)). Les feuilles de l'arborescence de clusters correspondent en effet aux sommets du graphe c'est-à-dire les lignes et les colonnes de la matrice d'adjacence. Il est

donc possible de propager des données de l'utilisateur, selon leur sémantique, de bas en haut dans la hiérarchie.



Figure 1 (a). Application d'un gradient de couleur et d'intensité.

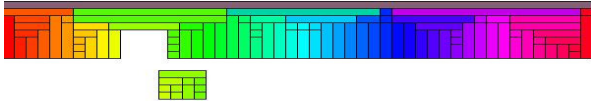


Figure 1 (b). Edition d'une hiérarchie de clusters par manipulation directe. Les feuilles de l'arbre sont alignées vers le bas.

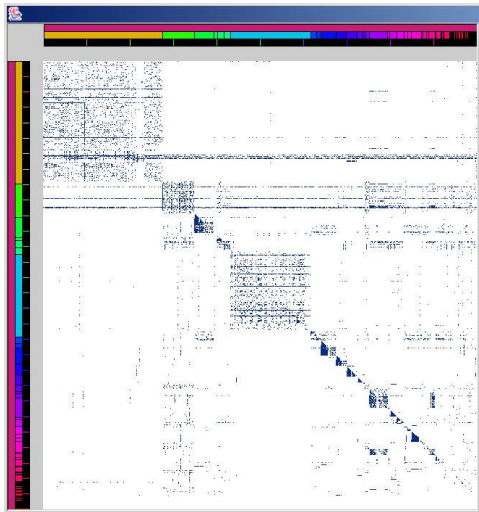


Figure 2. Représentation matricielle d'un site web de 589 pages. La topologie du graphe apparaît malgré la présence de bruit. Certains gros blocs sont formés de plusieurs sous-clusters que l'on peut fusionner. Les feuilles, trop nombreuses, sont repliées ; seuls les deux niveaux supérieurs de l'arborescence de clusters sont visibles.

Nous nous servons de cette visualisation pour manipuler directement les hiérarchies de clusters. L'utilisateur peut ainsi permuter, fusionner, réordonner, replier ou déployer les clusters. Toutes ces opérations d'édition de la hiérarchie de clusters sont effectuées directement à l'aide de la souris (figure 1 (b)) et ont des répercussions immédiates sur la matrice d'adjacence du graphe, notamment la réalisation d'un zoom sémantique.

L'originalité de VisAdj réside dans l'utilisation d'un composant élaboré de visualisation d'information comme les arbres à glaçons en tant qu'outil de contrôle et de manipulation d'une structure de données. L'utilisateur peut intervenir directement sur la hiérarchie de clusters en permutant, en fusionnant et en éditant les clusters, autant d'opérations d'édition qui se répercutent immédiatement sur la représentation matricielle du graphe. Cela revient à

manipuler les sommets du graphe par paquets au lieu d'avoir à manipuler des feuilles, beaucoup trop petites.

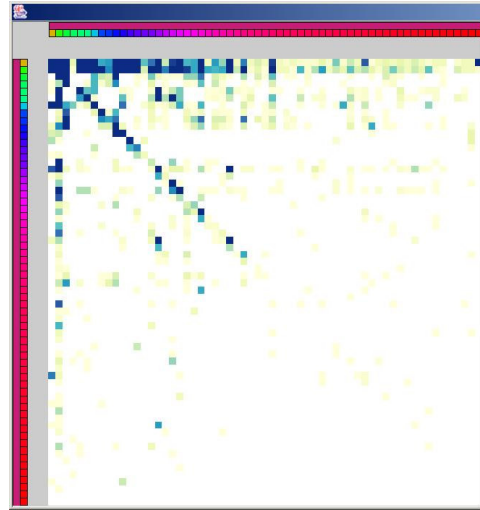


Figure 3. Vue agrégée représentant le même graphe que sur la figure 2. Les arêtes inter-clusters sont pondérées par le nombre d'arêtes qu'elles mettent en jeu dans le graphe d'origine. Les clusters fortement peuplés apparaissent en foncé.

REMERCIEMENTS

Ce travail a été réalisé dans le cadre du projet RNTL OADymPPaC (contraintes.inria.fr/OADymPPaC). Nous souhaiterions remercier l'ensemble de nos partenaires pour les échanges fructueux que nous avons eu avec eux.

BIBLIOGRAPHIE

1. Barlow, T. and Neville, P. A Comparison of 2-D Visualisations of Hierarchies. IEEE Symposium on Information Visualization 2001 (Infovis'2001), 22-23 October 2001, San Diego, CA, USA. IEEE Computer Society, pp. 131-138.
2. Berkhin, P. *Survey of Clustering Data Mining Techniques*. Accrue Software, Inc. San Jose, CA, 2002.
3. Bertin, J. *Sémiologie graphique : Les diagrammes – Les réseaux – Les cartes*. 1967. Editions de l'EHESS 1998.
4. Carpendale, M. S. T. and Montagnese, C. *A framework for unifying presentation space*. Proceedings of the 14th annual ACM symposium on User interface software and technology, November 2001, ACM Press, pp. 61-70
5. Fekete, J.-D. and Plaisant, C. *Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization*. Proceedings of the International Conference on Human Factors in Computing Systems (CHI 99), pp. 512-519. Ed. Kate Ehrlich and William Newman, ACM mai 1999.