Centralite du second ordre : Calcul distribue de l'importance de noeuds

Anne-Marie Kermarrec, Erwan Le Merrer, Bruno Sericola, Gilles Tredan

Centralité du second ordre : Calcul distribué de l'importance de noeuds

A.-M. Kermarrec, E. Le Merrer, B. Sericola, G. Trédan

IRISA/INRIA - Campus de Beaulieu - 35042 Rennes - France

Dans le contexte de la théorie des graphes pour les réseaux sociaux, la notion de centralité a été introduite pour mesurer l'importance relative de noeuds dans une topologie donnée. Connaître cette importance est un enjeu majeur pour assurer la robustesse des systèmes distribués. De nombreuses formes de centralités ont déjà été définies ; dans le contexte des systèmes distribués, elles sont cependant soit d'un intérêt limité (centralité des degrés), soit difficilement distribuables (centralité d'intermédiarité). Dans cet article, nous introduisons une nouvelle forme de centralité : la centralité du second ordre. Celle-ci est calculée de façon totalement distribuée, au moyen d'une marche aléatoire. Elle attribue à chaque noeud une valeur indicatrice de son importance dans le graphe. Pour cela, chaque noeud conserve les temps écoulés entre deux visites de la marche et calcule l'écart type de ces temps. Nous montrons que cet écart type est une mesure de centralité qui permet également de caractériser globalement la topologie d'un graphe donné.

Keywords: Centralité, Caractérisation de graphes, Marches Aléatoires

1 Introduction

Les réseaux large-échelle actuels, tels que les réseaux pair-à-pair, sociaux, ou sans fil, exhibent des structures de taille et de complexité importantes. Afin d'exploiter efficacement l'information contenue dans ces réseaux (modélisables par des graphes), il est important d'en extraire les caractéristiques globales, et de réduire le volume d'information grâce à des méthodes d'agrégation. La taille de ces graphes rend cependant impossible - ou très coûteuse - leur manipulation de manière centralisée. De plus, l'accès à la topologie complète du graphe n'est pas toujours possible (*p.ex.* réseaux p2p cryptés) ; un autre cas de figure est l'industrie qui préfère exploiter la puissance de calcul des machines de ses clients plutôt que d'investir dans des fermes de calcul coûteuses en entretien. Il est alors nécessaire de concevoir des méthodes distribuées d'analyse des graphes.

L'analyse des graphes statiques, particulièrement des réseaux sociaux, est un sujet étudié depuis longtemps par les physiciens et les sociologues. Ces travaux ont donné naissance à la notion de *centralité* [Bar04, BP07, Fre77, New05], qui capture l'importance de chaque noeud dans un graphe d'interactions. Un système distribué peut grandement bénéficier de telles informations, par exemple pour éviter que la topologie ne dépende que de quelques noeuds, ou encore pour prévoir les partitionnements, ou répartir la charge applicative. Comme un noeud peut être considéré comme important à plusieurs titres, il existe différents types de centralité. Malheureusement, la majeure partie de ces centralités sont conçues pour un calcul centralisé (analyse offline), et sont difficilement adaptables aux contextes distribués. Dans cet article, nous introduisons la *centralité du second ordre*, qui repose sur un algorithme distribué simple à base d'une marche aléatoire.

2 Travaux connexes

Il est possible de distinguer deux niveaux de caractérisation d'un graphe : un niveau global, concernant la structure émergeant de l'ensemble des noeuds le composant, et un niveau local, ou l'on caractérise individuellement chaque noeud.

Au niveau global, les propriétés de connexité globale du graphe sont souvent capturées par les notions de *connectivité algébrique* et de *conductance* ; ces métriques indiquent la qualité de la connectivité d'un

graphe ; elles permettent *e.g.* de prévoir à quel point l'efficacité d'algorithmes exécutés sur le graphe est affectée par la topologie (conductance et temps de mixage d'une marche aléatoire sont liés).

Au niveau local, la *centralité* capture l'importance d'un noeud vis-à-vis de la topologie d'un graphe. La plus simple, la centralité dite des degrés, exprime le fait qu'un noeud important possède un degré élevé. Une centralité plus évoluée est la *centralité d'intermédiarité* qui exprime le fait qu'un noeud important est impliqué dans beaucoup de plus courts chemins [Fre77, Bar04]. Dans de nombreux cas cependant, un noeud important n'est pas forcément un noeud présent sur les chemins les plus courts : Newmann propose alors [New05] une mesure de la centralité d'intermédiarité basée sur des marches aléatoires, redonnant ainsi de l'importance aux chemins non optimaux mais toutefois critiques pour l'exécutions de nombre de protocoles. Le principe est simple : chaque noeud émet une marche aléatoire à destination de chaque autre noeud du graphe. Les noeuds comptent le nombre de marches aléatoires qui les ont traversés. Malheureusement, cette approche nécessite une connaissance totale du graphe pour chaque noeud. De plus, sa complexité en messages réduit l'applicabilité de la solution à de petits réseaux. C'est pourquoi nous proposons une nouvelle centralité qui est distribuable à faible coût.

3 Algorithme pour centralité du second ordre

Soit un graphe $G = (V,E)$ de n noeuds et m arrêtes. On suppose que G est symétrique et que la topologie ne change pas au cours du processus de calcul des centralités. Pour un noeud $i \in V$, nous notons V_i son voisinage et d_i son degré. Nous supposons G connexe. Sur ce graphe, une unique marche aléatoire est lancée à partir d'un noeud quelconque. A chaque pas, elle passe d'un noeud i à un noeud de V_i. Cette marche est permanente : elle ne possède pas condition d'arrêt et n'est jamais perdue.

Le calcul de la centralité du second ordre exploite le temps écoulé entre deux visites de la marche aléatoire sur un même noeud (dénommé *temps de retour*) Ce temps peut être exprimé soit de façon absolue, c'est à dire grâce à une horloge présente sur chaque noeud , soit en nombre de sauts réalisés par la marche aléatoire (qui transporte alors un compteur de sauts que chaque noeud incrémente). L'idée de l'algorithme est que l'importance d'un noeud est liée à la régularité des visites d'une marche aléatoire sur ce noeud. Pour connaître son importance, un noeud i enregistre les temps de retour de la marche dans un tableau noté Ξ_i, et calcule l'écart-type de ces valeurs. Ainsi, plus le noeud est important, plus les visites de la marche aléatoire sont régulières.

Formellement, si $\Xi_i(k)$ désigne le $k$ième temps de retour au noeud i, après N échantillons, le noeud i calcule $\sigma_i(N) = \sqrt{\frac{1}{N-1}\Sigma_{k=1}^{N}\Xi_i(k)^2 - [\frac{1}{N-1}\Sigma_{k=1}^{N}\Xi_i(k)]^2}$. Les temps de retour étant indépendants, la loi forte des grands nombres nous permet d'écrire : $\lim_{N\longrightarrow\infty}\sigma_i(N) = \sigma_i$ ou σ_i est la centralité du second ordre du noeud i. Ainsi, plus σ_i est faible, plus le noeud i est important, et inversement.

Dans son article, Newman utilise des marches *simples*, c'est à dire relayées équiprobablement vers chaque voisin de leur position courante. Un tel système avantage les noeuds de haut degré, puisque la distribution stationnaire d'une marche est $\pi_i = d_i/2m$, *i.e.* les chances de voir la marche sont proportionnelles au degré du noeud, et ce même pour les noeuds faiblement centraux. Pour éviter ce phénomène, nous utilisons une marche débiaisée (méthode Metropolis-Hastings [Has70]) de manière à ce que $\pi_i = 1/n, \forall i \in V$. Cette méthode a ainsi l'avantage de ne pas favoriser artificiellement les noeuds de fort degré : la centralité d'un noeud ne dépend alors plus que de sa position dans le graphe. Dans le cas d'un graphe ligne, [KLMST09] montre que les noeuds les plus régulièrement visités par une marche aléatoire sont au centre. Cette propriété n'est pas liée au dégré, puisqu'il est dans ce cas le même pour presque tous les noeuds. Pour la même raison, les marches biaisée et débiaisée sont presques équivalentes dans ce cas. Dans le cas général, où les marches peuvent différer, nous utilisons la marche débiaisée qui a l'avantage de mieux repartir la marche aléatoire parmi les noeuds, sans considération de leur degré. En pratique, le débiaisement consiste à transmettre ou pas la marche avec une probabilité dépendante du rapport des degrés des noeuds impliqués dans la transmission (*cf* lignes 1-8 de l'Algorithme 1).

4 Analyse et caractérisation de graphes

Nous fournissons ici les résultats de l'analyse théorique du principe de l'algorithme. Le détail est disponible dans le rapport [KLMST09]. Nous présentons dans cette section la formule permettant de calculer la

Algorithme 1 Centralité du second ordre

```
1: Sur réception de la marche aléatoire par le noeud i:
2:    Choisir un voisin j de Γ_i uniformément au hasard
3:    Demander d_j à j
4:    Générer un nombre aléatoire p ∈ [0,1] uniformément
5:    si p ≤ d_i/d_j alors
6:       Faire suivre la marche aléatoire à j
7:    sinon
8:       La marche aléatoire reste sur i
9:    si Première visite de la marche aléatoire sur i alors
10:      Créer le tableau Ξ_i
11:   sinon
12:      Calculer les temps de retour r depuis la dernière visite
13:      Ajouter r à Ξ_i
14:      si |Ξ_i| ≥ 3 alors
15:         Calculer l'écart-type σ_i(N)
```

centralité de second ordre théorique pour chaque noeud d'un graphe quelconque, étant donnée sa matrice de probabilité de transition.

Soit un graphe G de n noeuds, de matrice de transition $P = (P(i,j))_{i,j\in[1..n]}$. Soit la matrice $Q_j(i,\ell) = \begin{cases} P(i,\ell) & \text{si} \quad \ell \neq j \\ 0 & \text{si} \quad \ell = j. \end{cases}$. Celle-ci permet de calculer la matrice $M = (M(i,j))_{i,j\in[1..n]}$ où $M(i,j)$ représente le nombre moyen de sauts nécessaires à une marche aléatoire, partant de l'état i, pour arriver pour la première fois en l'état j. Nous avons : $M_j = (I - Q_j)^{-1}\mathbb{1}$ avec $M_j(i) = M(i,j)$, $\mathbb{1}$ étant un vecteur colonne dont toutes les entrées valent 1 (voir le rapport pour les détails du calcul). La formule (1) donne la centralité du second ordre d'un noeud j :

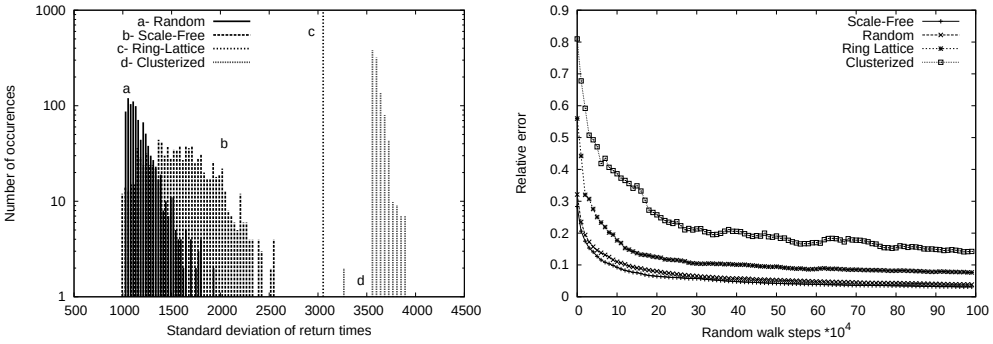
$$\sigma(j) = \sqrt{2\sum_{i\in[1..n]} M(i,j) - n(n+1)}. \tag{1}$$

Cette formule a deux intérêts : le premier est de permettre aux concepteurs de systèmes distribués de calculer des valeurs théoriques pour les structures qu'ils souhaitent déployer. Cela permet ensuite aux noeuds d'avoir une référence des temps de retours attendus, et ainsi de détecter des situations anormales en cas de variance importante. Son second intérêt est de montrer que l'écart type des temps de retour d'une marche aléatoire débiaisée est effectivement une centralité : si on instancie cette formule à une ligne ([KLMST09]), les noeuds ayant un plus faible σ sont les noeuds du centre, autrement dit les plus importants/centraux.

L'observation de la distribution des σ, calculés pour chaque noeud, permet d'établir la « signature » d'un graphe. La figure 1(a) représente les distributions des valeurs de σ sur différents graphes types. Ces graphes types sont : (a) aléatoire, (b) sans-échelle (modèle d'attachement préférentiel), (c) « ring lattice », c'est à dire un anneau ou chaque processus est connecté à k de ces voisins à droite et à gauche, et (d) clusterisé : deux graphes aléatoires connectés par un seul pont. Tous ces graphes sont égaux en nombre de noeuds ($n = 1000$) et en densité moyenne ($d = 20$). Un point sur cet histogramme, *e.g.* $(x = 2500, y = 3)$ signifie que 3 noeuds ont un σ égal à 2500. Cette figure permet plusieurs observations ; tout d'abord le graphe ayant les plus petites valeurs de σ est le graphe aléatoire, suivi du sans-échelle, puis du ring-lattice et enfin du graphe clusterisé. Cela correspond à un classement possible de leur robustesse : le graphe aléatoire est difficile à déconnecter (suppression aléatoire de noeuds), à l'inverse du graphe clusterisé puisqu'il suffit de supprimer un des noeuds du pont pour couper le graphe en deux composantes. Concernant l'étalement des distributions : le sans-échelle est très étalé (confirmant la disparité des importances), et certains noeuds ont des σ très bas, ce sont les *hubs* (noeuds de plus hauts degrés). Les noeuds du ring-lattice, une structure symétrique, ont tous le même écart-type. Le graphe clusterisé exhibe lui un ensemble groupé de valeurs dans la plage $3500 - 4000$, mais aussi deux noeuds avec des σ à 3200 : ce sont les deux noeuds du pont reliant les clusters.

5 Évaluation

Nous présentons ici une simulation qui montre la vitesse de convergence de l'algorithme vers les valeurs théoriques calculées à l'aide de la formule 1. A cette fin, l'algorithme a été simulé grâce à Peersim, sur



(a) (b)

FIG. 1: *a*) Signatures de différents graphes ; *b*) Vitesse de convergence de l'algorithme vers le σ théorique

la palette de graphes présentés section 4. La figure 1(b) synthétise ces évaluations : elle représente l'erreur relative entre l'estimation des noeuds et la valeur théorique calculée avec une connaissance totale du graphe. Ainsi le point $(x = 100.10^4, y = 0.1)$ signifie qu'après 10^6 pas de la marche aléatoire, en moyenne une erreur de 10% entre la centralité estimée des noeuds et leur centralité théorique existe. Chaque point de la courbe est la moyenne de 20 expériences indépendantes.

La première constatation valide la convergence de l'algorithme vers la valeur théorique de σ. La seconde concerne la vitesse de convergence ; en théorie, la vitesse de convergence de l'algorithme dépend du temps de *couverture* du graphe par la marche aléatoire. Dans le cas d'une marche simple, la borne supérieure est $\frac{4}{27}n^3 + o(n^3)$ pour le graphe *lollipop* [Fei95]. Il est raisonnable d'estimer que la borne est du même ordre pour une marche débiaisée. Nous pouvons observer sur cette figure que pour les graphes qui ne sont pas trop dégénérés et à diamètre faible (aléatoire et sans-échelle), la convergence est beaucoup plus rapide, aussi, après 10^6 pas les noeuds possèdent déjà un estimateur relativement précis. La Figure 1(a) montre que le σ moyen du graphe clusterisé est 2 à 3 fois plus grand que celui du graphe aléatoire ; ainsi, malgré 20% d'erreur un noeud peut décider de manière sûre s'il se trouve dans un graphe dégénéré ou pas. Les vitesses de convergence de l'algorithme respectent ici bien les propriétés de navigabilité des différents graphes.

[LMT09] compare la centralité de second ordre avec les autres centralités classiques. Cette comparaison étudie la dégradation des propriétés d'un graphe lorsque les noeuds les plus importants pour une centralité donnée sont retirés. Elle montre que la centralité de second ordre donne de bons résultats à un coût compétitif.

En conclusion, la centralité de second ordre permet d'attribuer aux noeuds, de manière distribuée, un indice caractérisant leur importance dans la topologie.

Références

[Bar04] M. Barthelemy. Betweenness centrality in large complex networks. *EUR.PHYS.JOUR.B*, 38 :163, 2004.

[BP07] U. Brandes and C. Pich. Centrality estimation in large networks. *International Journal of Bifurcation and Chaos*, pages 2303–2318, 2007.

[Fei95] U. Feige. A tight upper bound on the cover time for random walks on graphs. *RSA : Random Structures & Algorithms*, 6, 1995.

[Fre77] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1) :35–41, 1977.

[Has70] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1) :97–109, 1970.

[KLMST09] A.-M. Kermarrec, E. Le Merrer, B. Sericola, and G. Trédan. Rr-6809 inria - second order centrality :distributed assessment of nodes criticity in complex networks, 2009.

[LMT09] E. Le Merrer and G. Trédan. Centralities : Capturing the fuzzy notion of importance in social graphs. In *2nd ACM Workshop on Social Network Systems*, 2009.

[New05] M. E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1) :39–54, January 2005.