Degree-corrected block model : une nouvelle approche et une initialisation efficace pour l'inférence

Alexandra DACHE[1] Nicolas GILLIS[1] Arnaud VANDAELE[1]

[1]Faculté Polytechnique, Université de Mons, 9 Rue de Houdain, 7000 Mons, Belgique

Résumé – La détection de communautés est un outil essentiel en analyse de données. Une approche couramment utilisée repose sur le clustering basé sur un modèle de graphe (block modeling), qui permet d'identifier des structures sous-jacentes variées. Le degree-corrected block model (DCBM) est un modèle établi qui prend en compte l'hétérogénéité des degrés des nœuds. De nombreuses méthodes d'inférence existent pour estimer les paramètres du DCBM à partir du graphe, mais elles sont toutes basées sur des heuristiques. En raison de cette nature heuristique, la qualité des résultats dépend fortement de l'initialisation, qui est effectuée de manière aléatoire. Dans cet article, nous proposons une nouvelle approche fondée sur la factorisation matricielle pour l'estimation des paramètres du DCBM, ainsi qu'une méthode efficace pour leur initialisation pour l'inférence. Nous montrons que notre modèle donne d'excellents résultats sur des réseaux de référence, et que notre stratégie d'initialisation permet d'obtenir de meilleures solutions tout en accélérant la convergence des méthodes d'inférence existantes.

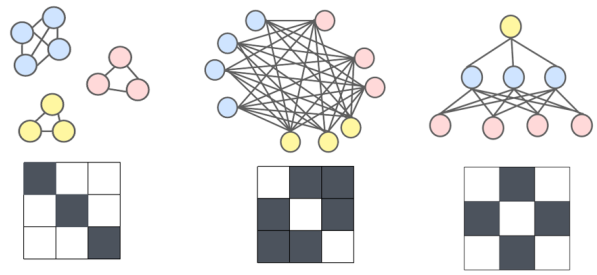
Abstract – Community detection is an essential tool in data analysis. A commonly used approach relies on clustering based on a graph model (block modeling), which allows the identification of various underlying structures. The degree-corrected block model (DCBM) is an established model that accounts for the heterogeneity of node degrees. Many inference methods exist to estimate the parameters of the DCBM from the graph, but they are all based on heuristics. Due to this heuristic nature, the quality of the results strongly depends on the initialization, which is done randomly. In this article, we propose a new approach based on matrix factorization for estimating the parameters of the DCBM, as well as a theoretically well-grounded method for their initialization for the inference. We show that our model yields excellent results on benchmark networks and that our initialization strategy leads to significantly better solutions while reducing the number of iterations needed by existing inference methods.

1 Introduction

Le stochastic block model (SBM), introduit par [7], modélise le réseau avec des blocs de nœuds, où chaque interaction entre deux nœuds suit une distribution de probabilité basée sur les blocs auxquels les nœuds appartiennent. Un modèle SBM avec n nœuds répartis en r blocs ou communautés peut être entièrement caractérisé par deux matrices de paramètres. La première est une matrice $n \times r$, notée Z, qui encode la communauté à laquelle appartient chaque nœud : $Z(i,k) = 1$ si le nœud i appartient à la communauté k, et $Z(i,k) = 0$ sinon. La seconde est une matrice $r \times r$, notée θ, de probabilités, où $\theta(k,l)$ représente la probabilité qu'une arête existe entre deux nœuds appartenant respectivement aux communautés k et l. Un graphe non-orienté avec une matrice d'adjacence A suit un modèle SBM si $A(i,j) = A(j,i) \sim \text{Bernoulli}((Z\theta Z^\top)_{i,j})$, chaque arête étant distribuée selon une loi de Bernoulli étant donné les communautés des nœuds. Étant donnés Z et θ, la vraisemblance d'observer la matrice d'adjacence A est

$$P(A|Z,\theta) = \prod_{j \leq i}^{n} (Z\theta Z^\top)_{i,j}^{A_{i,j}} (1 - (Z\theta Z^\top)_{i,j})^{(1-A_{i,j})}.$$

Une tâche cruciale, appelée l'inférence, consiste à estimer les paramètres les plus probables Z et θ à partir de la matrice d'adjacence pour maximiser cette vraisemblance. Les SBM doivent leur succès à leur simplicité et à la diversité des structures de réseau qu'ils peuvent modéliser. En effet, contrairement à la plupart des méthodes qui ne permettent d'identifier que des structures associatives (c'est-à-dire davantage de connexions entre les nœuds d'un même groupe), les SBM peuvent identifier une grande variété de structures (voir Figure 1) et de combinaisons de celles-ci. Les SBM ont été



(a) Associative (b) Dissociative (c) En chaîne

FIGURE 1 : Exemples de structures détectables par un SBM avec 3 blocs, illustrées à l'aide de la matrice θ, où les entrées présentant des valeurs élevées sont représentées en noir.

largement étudiés [1]. Cependant, ce modèle présente une limitation majeure : il suppose qu'au sein d'une communauté, tous les nœuds ont la même importance et ont les mêmes probabilités de connexion, ce qui conduit à une distribution des degrés identique pour tous les nœuds d'une même communauté. Or, dans les réseaux réels, les degrés des nœuds sont généralement hétérogènes. Le SBM classique peine alors à détecter les communautés et tend à regrouper les nœuds principalement en fonction de leur nombre de connexions [8].

Pour prendre en compte l'hétérogénéité des degrés, une manière de procéder [8] est d'autoriser la matrice Z à avoir des poids non-binaires. Dans ce cas, la probabilité d'avoir une arête

entre les nœuds i et j est donnée par $Z(i,k_i)\theta_{k_i,k_j}Z(j,k_j)$ où $Z(i,k_i)$ peut être interprété comme le niveau de "sociabilité" du nœud i. Plus $Z(i,k_i)$ est grand, plus i a de chances d'être connecté à d'autres nœuds. Ce modèle général est appelé le degree-corrected block model (DCBM) [8] [5]. Pour faciliter les calculs, le DCBM de Karrer et Newman [8] approxime la loi de Bernoulli par une loi de Poisson :

$$A_{i,j} = A_{j,i} \;\sim\; \text{Poisson}\left(\left(Z\theta Z^\top\right)_{i,j}\right),$$

qui intègre la possibilité de multi-arêtes. Pour des réseaux sparses, la distribution de Poisson diffère de manière négligeable de la distribution de Bernoulli comme le nombre attendu d'arêtes devient très proche de la probabilité d'une arête. L'avantage de cette modélisation avec une loi de Poisson, comparée à Bernoulli, est que les valeurs optimales des paramètres Z et θ sont directement calculables pour une affectation fixée des nœuds aux communautés. La vraisemblance est ainsi directement calculable pour des communautés fixées, ainsi que sa variation lorsqu'un nœud est déplacé d'une communauté à une autre. Pour estimer une bonne partition des nœuds en r communautés à partir du graphe sans limitation stricte quant au type de structure du graphe[1], il existe de nombreuses heuristiques qui, à partir d'une partition initiale des nœuds, effectuent des déplacements de nœuds pour maximiser la vraisemblance. L'affectation initiale des nœuds est générée de manière aléatoire et peut parfois être mauvaise, ce qui peut conduire à une convergence de l'algorithme vers de mauvais minima locaux. Ce problème devient particulièrement gênant pour les graphes de grande taille, où le nombre d'essais doit être augmenté et où le temps de convergence devient long.

Dans cet article, nous proposons une nouvelle approche basée sur la factorisation matricielle pour l'estimation des paramètres du DCBM. Ce développement nous a conduit à trouver une méthode efficace pour l'initialisation des paramètres Z et θ, une étape cruciale pour l'inférence.

2 Nouvelle approche basée sur la factorisation matricielle

Le DCBM peut être reformulé comme un problème de factorisation matricielle. Étant donné la matrice d'adjacence d'un graphe, $A \in \{0,1\}^{n\times n}$, et un nombre de communautés, r, nous cherchons à résoudre le problème suivant :

$$\min_{Z\in\mathbb{R}_+^{n\times r},\theta\in\mathbb{R}_+^{r\times r}} d(A, Z\theta Z^\top) \quad \text{tel que} \quad Z^\top Z = I_r,$$

où $d(A,B)$ mesure l'erreur entre les matrices A et B. La contrainte d'orthogonalité sur Z, en plus de la positivité, garantit que ses colonnes ont des supports disjoints et impose leur normalisation avec une norme ℓ_2. Cette normalisation peut être effectuée sans perte de généralité, comme les colonnes de Z ne sont définies qu'à un facteur multiplicatif près, lequel peut être absorbé dans les lignes et colonnes correspondantes de θ. En effet, on peut multiplier Z par une matrice diagonale D tout en préservant à la fois le support de Z et le produit $Z\theta Z^\top$ (avec une transformation appropriée de θ) :

$$Z\theta Z^\top = (ZD)(D^{-1}\theta D^{-1})(ZD)^\top.$$

[1] Les méthodes spectrales ou basées sur la modularité ne sont pas abordées dans le cadre de cet article, car elles sont limitées à des structures associatives ou nécessitent de connaître le type de structure à l'avance.

Dans le cas du DCBM de Karrer et Newman [8] qui repose sur une distribution de Poisson, maximiser la vraisemblance est équivalent à minimiser la Kullback-Leibler (KL) divergence entre A et $Z\theta Z^\top$.

A la place de la KL divergence, nous proposons d'utiliser la norme de Frobenius pour mesurer l'erreur entre A et $Z\theta Z^\top$:

$$d(A, Z\theta Z^\top) = \left\|A - Z\theta Z^\top\right\|_F^2 = \sum_{i,j}\left(A - Z\theta Z^\top\right)_{i,j}^2.$$

Ce modèle est référé comme la trifactorisation matricielle symétrique non-négative orthogonale (OtrisymNMF).

Étudiée dans un précédent article [2], nous avions proposé une méthode d'initialisation pour OtrisymNMF. Nous proposons également de l'utiliser comme initialisation pour les méthodes d'inférence du DCBM. Cette initialisation se base sur la propriété que dans le cas sans bruit, nous avons $A = WZ^\top$ avec $W = Z\theta$, où Z est séparable, c'est-à-dire qu'il existe un ensemble d'indices K de cardinalité r tel que $Z(K,:) = \text{diag}(z)$, avec un vecteur positif $z \in \mathbb{R}_+^r$, et $\text{diag}(\cdot)$ représentant une matrice diagonale avec z sur sa diagonale. Plus simplement, Z est séparable car elle contient, à permutation et mise à l'échelle près, la matrice identité comme sous-matrice. Cela découle du fait que Z est orthogonale. On a :

$$A(:,K) = WZ(K,:)^\top = W\,\text{diag}(z).$$

Cela signifie que $A(:,K)$ est égal à W à des facteurs près. Trouver cet ensemble K de colonnes de A est résoluble en temps polynomial à l'aide d'algorithmes robustes au bruit [6, Chapter 7]. Dans notre cas, supposer que Z est séparable revient à supposer qu'il existe au moins un nœud par communauté. En pratique, on observe généralement plusieurs nœuds appartenant à chaque communauté. C'est une situation plus favorable, car il existe plusieurs colonnes de A proches de chaque colonne de W. Pour estimer W sous cette hypothèse plus forte, nous utilisons la méthode SVCA (smoothed vertex component analysis) proposée par Nadisic et al. [10]. SVCA améliore la robustesse en sélectionnant p colonnes de A (au lieu d'une seule qui peut être très bruitée) pour estimer chaque colonne de W. Dans notre situation, on cherche idéalement à estimer chaque colonne de W en utilisant p colonnes de A, où p est le nombre de nœuds dans la communauté correspondante. Comme les tailles des communautés sont généralement inconnues, nous choisissons $p = \max\left(2, \left\lfloor 0{,}1 \cdot \frac{n}{r}\right\rfloor\right)$, en supposant qu'il existe au moins $\max\left(2, \left\lfloor 0{,}1 \cdot \frac{n}{r}\right\rfloor\right)$ nœuds par communauté. Nous choisissons SVCA plutôt que SSPA [10], car elle est moins sensible au bruit non-gaussien. Notons que SVCA est une méthode aléatoire. Une fois W déterminée via SVCA, nous retrouvons Z en résolvant le problème suivant :

$$\min_{Z\geq 0} \|A - WZ\|_F^2 \quad \text{tel que} \quad Z^\top Z = I_r,$$

qui est lié à la factorisation orthogonale non-négative (ONMF) [12], et où Z peut être calculée exactement, comme proposé dans [12]. Cela revient à assigner chaque colonne de A au centroïde le plus proche parmi les colonnes de W, la proximité étant mesurée en termes d'angles.

Pour OtrisymNMF, cette méthode permet d'initialiser Z et $\theta = Z^\top AZ$. Quant à l'initialisation du DCBM de Karrer et Newman via SVCA, il suffit d'utiliser la partition des nœuds fournie par la matrice Z ainsi obtenue.

3 Expériences numériques

Nous avons comparé le modèle OtrisymNMF et le modèle DCBM de Karrer et Newman, ainsi que leurs algorithmes respectifs, sur des réseaux synthétiques et réels couramment utilisés pour évaluer les performances des méthodes de détection de communautés. De plus, nous avons comparé l'initialisation aléatoire avec notre approche d'initialisation SVCA. Pour comparer l'affectation des nœuds trouvée par les méthodes avec l'affectation connue, nous utilisons l'information mutuelle normalisée (NMI) définie comme dans [3]. Elle mesure la similarité entre la partition réelle et la partition obtenue. Le NMI prend une valeur 1 si les partitions sont identiques et 0 si elles sont non-corrélées. Le code source est disponible sur `https://github.com/Alexia1305/DCBM_OtrisymNMF`

3.1 Réseaux synthétiques

Nous comparons notre algorithme, décrit dans [2], pour le modèle OtrisymNMF avec trois méthodes d'inférence pour le modèle DCBM de Karrer et Newman [8]. La première méthode est l'algorithme original proposé par [8] pour le modèle DCBM que nous désignons sous le nom de KN. La deuxième méthode est une adaptation de l'algorithme de Kernighan-Lin, proposée par [4] et appelée KL-EM. Enfin, la troisième méthode est un algorithme Markov Chain Monte Carlo, plus précisément l'algorithme Metropolis-Hastings développé par Peixoto [11], désigné sous le nom de MHA. Pour toutes ces méthodes, nous avons utilisé le code issu de [4].

Pour comparer les performances des méthodes d'inférence, nous avons choisi d'utiliser le benchmark LFR [9], qui, contrairement aux benchmarks synthétiques traditionnels, permet de créer de grands graphes réalistes présentant des degrés hétérogènes et des tailles de communautés variées. Pour ce faire, le benchmark LFR intègre des distributions de loi de puissance, observées dans les graphes réels, à la fois pour les degrés des nœuds et les tailles des communautés, avec des exposants respectifs γ et β. La structure des communautés est déterminée par le paramètre de mélange μ, qui représente la fraction des arêtes situées en dehors de la communauté pour chaque nœud. Pour ajuster le mélange des arêtes internes et externes, la méthode procède à un réarrangement des arêtes. Il convient de noter que le réseau généré ne suit pas exactement un DCBM. Les autres paramètres pour générer les réseaux sont le nombre de nœuds N et le degré moyen $\langle k \rangle$.

Pour les tests, nous avons choisi la même configuration que dans [4] et [9], à savoir 1000 nœuds, $\gamma = 2$, $\beta = 1$, avec des degrés de valeur moyenne égale à 20 et ne dépassant pas 50. Pour chaque valeur du paramètre μ comprise entre 0 et 0.6, nous avons généré 10 réseaux de test à l'aide du code original [9]. Les réseaux obtenus comportent des communautés de 20 à 100 nœuds, ce qui conduit à un total de 16 à 24 communautés par réseau. Nous avons testé chaque méthode sur chaque graphe avec un nombre d'essais égal à 10, en conservant la solution correspondant à la meilleure valeur objective (probabilité pour DCBM et erreur pour OtrisymNMF). Pour la méthode MHA, le nombre d'étapes est fixé à 250 000.

La Figure 2 fournit le NMI moyen et le temps de calcul moyen pour les trois méthodes, à savoir KN, KL-EM et MHA, avec une initialisation aléatoire et une initialisation utilisant SVCA. On observe qu'avec l'initialisation SVCA, les trois

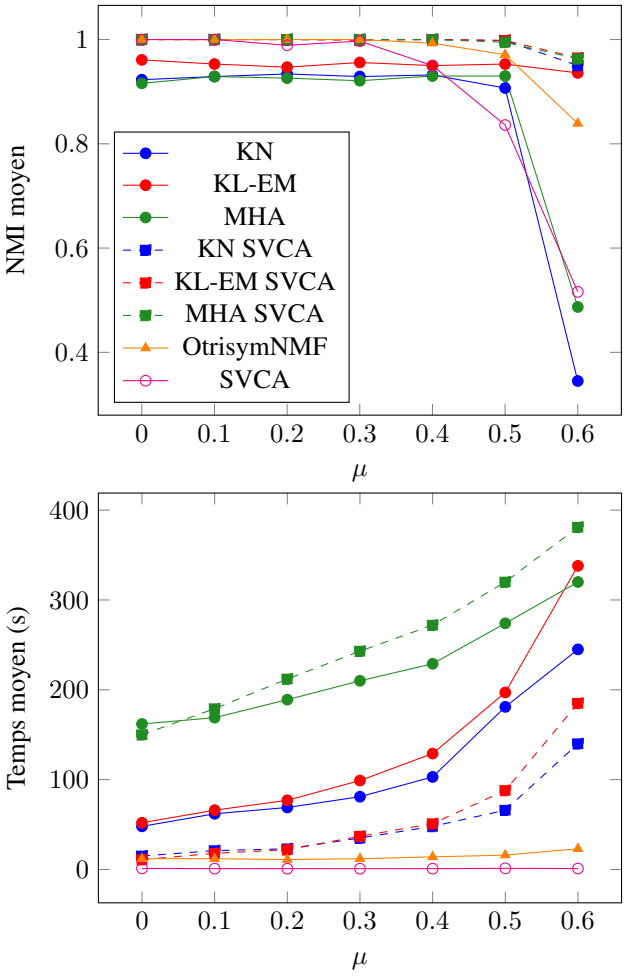


FIGURE 2 : Temps et NMI moyens sur 10 graphes LFR pour μ allant de 0 à 0.6 avec 10 essais pour chaque méthode.

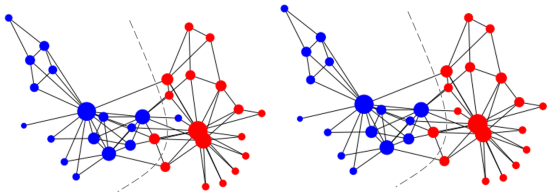
méthodes parviennent à retrouver parfaitement les communautés jusqu'à $\mu = 0.5$, tandis qu'avec l'initialisation aléatoire, elles échouent, même lorsque $\mu = 0$, c'est-à-dire lorsque les communautés n'ont aucune connexion entre elles. En ce qui concerne le temps de calcul, les méthodes KN et KL-EM convergent plus rapidement avec notre initialisation. Le temps de calcul reste relativement constant pour MH, car le nombre d'étapes est fixé à l'avance. La figure présente également les résultats pour OtrisymNMF initialisé avec SVCA, ainsi que pour SVCA seul. Pour SVCA, nous effectuons 10 initialisations et retenons celle ayant l'erreur la plus faible. OtrisymNMF donne des résultats équivalents à ceux de KN initialisé par SVCA, sauf pour $\mu = 0.6$, où il converge plus rapidement vers une solution légèrement moins bonne. L'utilisation de SVCA pour trouver directement les communautés est très rapide et permet d'obtenir d'excellents résultats jusqu'à $\mu = 0.4$. En résumé, SVCA permet de réduire le temps d'inférence (pour KN et KL-EM) et d'améliorer les résultats, dans les deux cas significativement.

3.2 Réseaux empiriques

Le premier exemple est le réseau du karaté club de Zachary, souvent utilisé comme référence pour tester les algorithmes de détection de communautés. Le réseau représente les interactions sociales entre les 34 membres d'un club de karaté. Suite à un conflit interne, le club s'est divisé en deux factions

distinctes. Les partitions obtenues, qui maximise la probabilité du DCBM de Karrer et Newman et qui minimise l'erreur d'OtrisymNMF, sont illustrées sur la Figure 3. La partition trouvée par OtrisymNMF correspond parfaitement aux deux communautés, à l'exception d'un nœud, le même nœud qui est généralement mal classé par les autres algorithmes de détection de communautés. Dans le cas du modèle DCBM de Karrer et Newman, un nœud supplémentaire est mal classé. Pour nous assurer que cela n'était pas dû à de mauvaises heuristiques, nous avons vérifié que la probabilité de la partition sous le modèle était bien plus élevée que celle de la partition avec le nœud fréquemment mal classé, ainsi que celle de la partition exacte. Cette légère différence peut être attribuée au fait que le graphe est relativement dense et petit, et que la modélisation par Poisson introduit des erreurs car la probabilité d'avoir plus d'une arête entre deux nœuds n'est plus négligeable.

Le second réseau réel que nous avons testé est le Scotland



(a) DCBM (b) OtrisymNMF

FIGURE 3 : Partitions pour DCBM et OtrisymNMF pour le réseau karaté club. La ligne pointillée indique la partition réelle observée.

corporate interlock network [13], souvent utilisé comme benchmark biparti. Ce réseau décrit les connexions entre 136 administrateurs et 108 grandes sociétés. Le réseau étant déconnecté, nous nous concentrons uniquement sur sa plus grande composante, qui comprend 131 administrateurs et 86 entreprises. Pour les modèles OtrisymNMF et DCBM, la partition optimale (celle correspondant à la meilleure valeur objective) est bien la partition réelle du graphe (administrateurs et grandes sociétés). Cependant, cet optimum n'est généralement pas atteint avec une initialisation aléatoire. En moyenne, sur 100 tests avec 5 essais par initialisation aléatoire, le NMI est de $(16 \pm 19)\%$ pour KL-EM et de $(70 \pm 37)\%$ pour OtrisymNMF. En revanche, avec l'initialisation SVCA, le NMI atteint $(100 \pm 0)\%$ pour KL-EM et OtrisymNMF, retrouvant exactement la partition à chaque test. Les algorithmes améliorent considérablement les initialisations SVCA, qui ont en moyenne un NMI de 8%. Cela montre que, bien que SVCA fournisse une bonne initialisation, elle ne permet pas directement d'obtenir de bons résultats.

4 Conclusions

Dans cet article, nous avons utilisé la factorisation matricielle, et plus précisément l'OtrisymNMF, pour modéliser et estimer les paramètres d'un degree-corrected block model (DCBM). Grâce à cette nouvelle approche, nous avons proposé une méthode basée sur SVCA, robuste et théoriquement fondée, pour initialiser l'assignation des nœuds aux communautés lors de l'inférence du DCBM.

Lors de tests sur des réseaux réels et synthétiques, nous avons montré qu'OtrisymNMF rivalise avec le DCBM classique de Karrer et Newman en termes de performance. Dans le cas de graphes denses, tels que le réseau du karaté club, l'approximation due à l'utilisation de la distribution de Poisson dans le DCBM peut conduire à une moins bonne modélisation. Dans ces situations, OtrisymNMF pourrait constituer une meilleure alternative. Au cours de ces tests, nous avons également montré que notre initialisation basée sur SVCA améliore considérablement les résultats des méthodes d'inférence classiques du DCBM par rapport à une initialisation aléatoire. En fournissant un meilleur point de départ, elle permet de converger vers de meilleures solutions. Ce travail ouvre ainsi la voie une utilisation plus robuste du DCBM.

Références

[1] E. ABBE : Community detection and stochastic block models : recent developments. *J. Mach. Learn. Res.*, 18(177):1–86, 2018.

[2] A. DACHE, A. VANDAELE et N. GILLIS : Orthogonal symmetric nonnegative matrix tri-factorization. *In International Workshop on MLSP*. IEEE, 2024.

[3] L. DANON, A. DÍAZ-GUILERA, J. DUCH et A. ARENAS : Comparing community structure identification. *J. Stat. Mech. : Theory Exp.*, 2005(09):P09008, 2005.

[4] T. FUNKE et T. BECKER : Stochastic block models : A comparison of variants and inference methods. *PLOS ONE*, 14:1–40, 04 2019.

[5] C. GAO, Z. MA, A. Y. ZHANG et H. H. ZHOU : Community detection in degree-corrected block models. *Annals of Statistics*, 46, 07 2016.

[6] N. GILLIS : *Nonnegative Matrix Factorization*. SIAM, Philadelphia, 2020.

[7] P. HOLLAND, K. B. LASKEY et S. LEINHARDT : Stochastic blockmodels : First steps. *Social Networks*, 5:109–137, 1983.

[8] B. KARRER et M. E. J. NEWMAN : Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, 2011.

[9] A. LANCICHINETTI, S. FORTUNATO et F. RADICCHI : Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110, 2008.

[10] N. NADISIC, N. GILLIS et C. KERVAZO : Smoothed separable nonnegative matrix factorization. *Linear Algebra and its Applications*, 676:174–204, 2023.

[11] T. P. PEIXOTO : Efficient monte carlo and greedy heuristic for the inference of stochastic block models. *Physical Review E*, 89(1):012804, 2014.

[12] F. POMPILI, N. GILLIS, P.A. ABSIL et F. GLINEUR : Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing*, 141:15–25, 2014.

[13] J. SCOTT et M. HUGHES : *The anatomy of Scottish capital : Scottish companies and Scottish capital, 1900-1979*. Routledge, 2021.