Characterizing community detection algorithms and detected modules in large-scale complex networks

Vinh-Loc Dao

UNIVERSITE BRETAGNE LOIRE MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPÉRIEURE MINES-TELECOM ATLANTIQUE
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE
COMUE **U**NIVERSITE **B**RETAGNE **L**OIRE

Ecole Doctorale N°601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Informatique

Par

Vinh-Loc DAO

Characterizing community detection algorithms and detected modules in large scale complex networks

Thèse présentée et soutenue à IMT ATLANTIQUE - CAMPUS DE BREST, le 17 décembre 2018
Unité de recherche : Lab-STICC UMR 6285 - CNRS
Thèse N° : 2018IMTA0108

Rapporteurs avant soutenance :

Anne BOYER	Professeur, Université de Lorraine
Renaud LAMBIOTTE	Associate Professor, University of Oxford

Composition du jury :

Président et rapporteur :	Anne BOYER	Professeur, Université de Lorraine
Rapporteur :	Renaud LAMBIOTTE	Associate Professor, University of Oxford
Examinateur :	Vincent LABATUT	Maître de conférences, Université d'Avignon
Dir. de thèse :	Philippe LENCA	Professeur, IMT Atlantique
Co-encadrant :	Cécile BOTHOREL	Maître de conférences, IMT Atlantique
Invité(s)	Cédric BACHER	PhD, HDR, IFREMER

IMT ATLANTIQUE

DOCTORAL THESIS

Characterizing community detection algorithms and detected modules in large scale complex networks

Author: Vinh-Loc DAO

Supervisor: Cécile BOTHOREL
Director: Philippe LENCA

A thesis submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

Thesis prepared at

Department of Logic of Uses, Social and Information Sciences
DECIDE - Lab-STICC - CNRS, UMR 6285
IMT Atlantique

December 17, 2018

Abstract

Characterizing community detection algorithms and detected modules in large scale complex networks

by Vinh-Loc DAO

It is widely believed that real-world networks are organized in a way that their nodes establish modular groups. Attracted by this remark, many efforts have been devoted to developing methods that can efficiently highlight these hidden structures inside networks, yielding a new research domain called community detection and eventually becoming a fundamental task in network analysis. Many applications of community detection nowadays that can be mentioned such as: identifying groups of similar users in social networks; discovering communities of malicious web domains in network security; detecting plausible candidates for biological modules in protein-protein interaction networks, etc.

The problem that raises up our research question is: there is not any universal accepted definition of community structure due to the contextual-dependency of the definition *community* itself. By consequence, there exists a fundamental difficulty in the evaluation and the interpretation of the results of community discovery without a priori information on expected criteria. This thesis provides a recommendation for choosing appropriate methods of community detection. In order to do that, first we introduce theoretical concepts of popular methods existing in the literature to illustrate different classes of mechanisms that they employ given that these mechanisms strongly impact the final results. Then we point out some defective instances of traditional evaluation metrics and propose a descriptive approach for verifying and interpreting detected communities. Specifically, this approach helps to describe internal and external structure of communities in low-dimensional spaces to assist one in analyzing community structure produced by different detection methods. Interestingly, our empirical study exploiting this approach uncovers that networks across different categories including communication, technological, information, biological and social networks might have different community structures and can be described by distinguishable characterized topologies corresponding to popular graph models in the literature. Finally, we demonstrate a study on the community structural similarity of detection methods based on the likeliness of their outputs produced from a large dataset of real world networks. Our results show that some methods might identify statistically comparable community structures provided that a particular quality is given. The outcome of our analysis supplies proofs to convince practitioners in which kind of situations a suitable choice of method is crucial or insignificant. The result is also important in the sense that it helps to decide an analysis strategy, whether an expensive solution need to be pursued or a simple solution suffices. Instead of providing ready-to-use *formula*, we analyze a large spectrum of instances in three principle dimensions: network, method and quality metric. The analysis provides empirical supports on which one can rely to determine best methods for particular cases.

Acknowledgements

First, I would like to thank my thesis advisors Cécile Bothorel and Philippe Lenca without whom this thesis would not be feasible. They have been all with me since the beginning days of the project and worked so hard in order to make it financially realizable. I would not be able to put my first steps into this interesting world of network science without their efforts. I specially thank Cécile Bothorel for devoting her time to help me both in resolving scientific matters as well as correcting the manuscripts that I wrote during these years. I really appreciate that.

I would also express my special gratitude to the members of the jury. Especially, Prof. Anne Boyer and Prof. Renaud Lambiotte generously spent their time on reading the manuscript and gave very precious remarks and ideas that have much enriched my perspective and hindsight. Together with Dr. Vincent Labatut and Dr. Cédric Baucher, I would like to thank you all for having raised very appealing questions and comments during the thesis defense and letting it be a very enjoyable moment for me. Beside, this thesis would have not reach its end without Gwendal Simon, Noël Crespi and Christophe Hurter who accepted to evaluate the progress of my thesis.

I thank all my colleagues in LUSSI department of IMT Atlantique for making these past years be unforgettable memories. Among others, a special thank goes to Sébastien Bigaret who kindly set up and provided me necessary computing resources to realize my experiments. I thank Laurent Brisson and Inna Lyubareva for very supportive discussions that gave me many different viewpoints on the work that I conduct. I also thank my intern Phatrasek Jirabovonvisut, with whom I only worked during a short period but he helped me to resolve many technical issues. Beside, IMT Atlantique is not only about work but also memorable recreational moments. I would never forget the ambiance of the harbor during the summer festivals that I had chance to enjoy with my fellows, the refreshing moments wandering, swimming and picnicking around the beachy paradises of the surrounding areas, the emotional days supporting *Les Bleus* in the Brest Arena until when they went to the top of the world. It pains me a lot not to list you all here but the list would go incredibly long.

This thesis would never be possible without the endless encouragement of my beloved friends and family, especially my mother who always gives me unconditional supports. Words could not express how grateful I am to them and I feel really lucky to have them in my life.

This research is financed by *Fondation Mines-Télécom* via *Futur & Ruptures* program's scholarship. I thank all sponsors contributed to the development of the foundation, and hence this thesis. I also thank Patrick Meyer and John Puentes, heads of DECIDE research team who gave financial aid that helped my work timely promoted. I also appreciate very much the discussions that we had.

Last but not least, thank you dear reader even though your name is unknown to me. I hope that my findings could provide constructive information, even slightly.

Contents

List of Figures

List of Tables

List of Abbreviations

AMI	Ajusted Mutual Information
ARI	Adjusted Rand Index
BA	Barabási-Albert
CCF	Clustering Coefficient
CNM	Clauset-Newman-Moore
CONCLUDE	Complex Network Cluster Detection
DCSBM	Degree Corrected Stochastic Block Model
ER	Erdős-Rényi
FOMD	Fraction of Over Median Degree
GN	Girvan-Newman
LFR	Lancichinetti–Fortunato–Radicchi
LPA	Label Propagation Algorithm
meanODF	Average Out Degree Fraction
MDL	Minimum Description Length
MI	Mutual Information
NMI	Normalized Mutual Information
NVI	Normalized Variation of Information
ODF	Out Degree Fraction
OSLOM	Order Statistics Local Optimization Method
PPI	Protein-Protein Interaction
RB	Reichardt-Bornholdt
RCCLP	Radicchi-Castellano-Cecconi-Loreto-Parisi
RI	Rand Index
SBM	Stochastic Block Model
sdODF	Standard deviation Out Degree Fraction
SLPA	Speaker Listener label Propagation Algorithm
SN	Spectral method of Newman
TPR	Triangle Participation Ratio
VI	Variation of Information
WS	Watts-Strogatz

List of Symbols

Symbol	Description
$\mathcal{G}$	Graph representing a network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$
$\mathcal{V}$	Set of vertices of graph $\mathcal{G}$
$\mathcal{E}$	Set of edges of graph $\mathcal{G}$
n	Number of vertices in graph $\mathcal{G}$, $n = \|\mathcal{V}\|$
m	Number of edges in graph $\mathcal{G}$, $m = \|\mathcal{E}\|$
$\mathcal{P}$	Partition of graph $\mathcal{G}$
C	Community in partition $\mathcal{P}$ of graph $\mathcal{G}$
A	Adjacency matrix representing graph $\mathcal{G}$
B	Modularity matrix
D	Diagonal matrix of vertex degrees/weights of graph $\mathcal{G}$
L	Laplacian matrix, $L = D - A$
P	Stochastic/transition matrix between nodes of graph $\mathcal{G}$
P_{ij}^t	Probability of a random walker to go from i to j in t steps
P_{ij}	Averaged probability of having an edge between i and j
$\mathcal{Q}$	Quality function
e_{ij} or (i, j)	Edge between vertex i and vertex j
d_i or $d(i)$	Degree of vertex i in graph $\mathcal{G}$
k_i or $k(i)$	Degree of vertex i in graph $\mathcal{G}$
w_{ij}	Weight of edge e_{ij}
w_i	Weight of vertex i, $w_i = \sum_{j \in \mathcal{V}} w_{ij}$
a_{ij}	Element of i-th row, j-th column in adjacency matrix A
λ_1	Principle eigenvalue of adjacency matrix A

Résumé

Ces dernières décennies ont vécu une explosion phénoménale des recherches sur des systèmes complexes grâce à une conjoncture favorable de plusieurs facteurs: l'augmentation de puissance des ressources de calculs; la facilité d'échanger, de collecter et d'enregistrer de l'information; et surtout le développement de nouveaux algorithmes pour traiter de très grands graphes. Le concept central de ces études, en modélisant des systèmes complexes par des interactions entre leurs constituants, nous permet d'utiliser des modèles mathématiques tels que la théorie de graphe pour appréhender et expliquer des phénomènes collectifs se produisant et s'expliquant non pas par des individus mais exclusivement par leurs interactions. Une telle modélisation des systèmes complexes nous permet également d'intervenir dans de très nombreux domaines tel que la biologie, la sociologie, la technologie, l'informatique, etc dont les objets d'études peuvent être modélisés par des graphes d'interactions[1].

Parmi plusieurs caractéristiques surprenantes, les réseaux complexes possèdent une propriété structurelle non triviale appelée *structure communautaire* (Fortunato and Hric, 2016; Chakraborty et al., 2017; Labatut and Orman, 2017), consistant en des groupes d'acteurs fortement connectés entre eux et faiblement liés aux autres groupes dans leur réseau d'interactions. Cette propriété se retrouve dans des réseaux de très nombreux domaines et offre des perspectives intéressantes. C'est la raison pour laquelle de très nombreuses méthodes d'exploitation de structure communautaire ont été proposées dans la communauté scientifique depuis l'apparition d'une première méthode proposée par (Girvan and Newman, 2002). Rien qu'en trois années de 2015 à 2018, il y a environ 500 mille publications scientifiques indexées sur la plate-forme Google Scholar concernant le sujet de *community detection* (detection de communautés en anglais). Cette profusion de travaux nous conduit à des algorithmes les plus avancés et efficaces. Pourtant, cet avancement implique également la nécessité de développer en parallèle de nouvelles techniques capables d'évaluer, ou au moins aider à interpréter de manière automatique, des résultats produits par ces algorithmes.

Cette thèse, en s'inscrivant dans ce contexte, a pour l'objectif d'analyser et de comparer des méthodes de détection de communautés proposées dans la littérature. Nous nous intéressons à investiguer des techniques pouvant assister des analystes à choisir une ou plusieurs méthodes qui leur conviennent selon différentes contextes ainsi que des qualités structurelles attendues de communautés. Les analyses et les résultats exposés dans cette thèse sont loin d'être exhaustifs pour aborder tous les aspects de structures communautaires. Ils constituent cependant un des premiers efforts pour rapprocher les développements théoriques au sein de la communauté scientifique du monde des analystes, parfois non spécialistes, qui ont besoin d'étudier des réseaux d'interactions dans des cas concrets pour des simples explorations ou pour des prises de décision.

[1] Parfois appelés: *réseaux complexes*, *réseaux d'interactions*, *graphes de terrain* ou tout simplement *graphes* ou *réseaux*.

0.1 Contributions

Les travaux de cette thèse s'organisent autour de plusieurs problématiques sur la détection de structures communautaires dans des réseaux complexes et ont pour but d'aider des analystes à choisir des méthodes d'exploration et à interpréter les résultats obtenus. De nombreuses analyses et études ont été menées selon trois axes principaux correspondant à trois lignes de contributions majeures:

- Premièrement, il s'agit de la caractérisation de structures communautaires. À l'heure actuelle, il n'existe pas encore de moyen intuitif pour décrire de manière systématique des structures communautaires au sein d'un réseau, rendant difficile l'interprétation de différentes solutions de partitions. En conséquence, il est primordial d'inventer des techniques pour caractériser et distinguer différentes structures communautaires d'un réseau. La caractérisation de structure communautaire proposée dans cette thèse utilisant une approche empirique a pour l'objectif de décrire des motifs d'interactions entre les noeuds dans des réseaux. De plus, cette caractérisation nous permet de profiler des communautés identifiées dans des réseaux de différents domaines d'études et les associer à des modèles génératifs de graphe dans la littérature. De cette manière, nous avons montré que des réseaux dans des domaines différents peuvent avoir des motifs d'interactions très différentes (Dao, Bothorel, and Lenca, 2018a).

- Deuxièmement, malgré l'existence de multiples métriques de qualité pour évaluer des structures communautaires ainsi que des communautés, la valorisation de performance d'une méthode selon la qualité de résultat qu'elle produit n'est pas toujours interprétable de manière évidente. C'est pour cette raison que nous proposons une évaluation des communautés identifiées par divers méthodes de l'état de l'art en fonction de leur efficacité à repérer des communautés possédant différentes qualités. Cette évaluation nous permet de conclure que la plupart de méthodes de détection découvre des structures communautaires de très bonnes qualités par rapport à des communautés métadonnées dans le même graphe qui sont souvent pourtant utilisées comme vérités terrains pour la validation de performance de détection (Dao, Bothorel, and Lenca, 2017a), (Dao, Bothorel, and Lenca, 2017b).

- Enfin, pour pouvoir identifier des méthodes de détection appropriées selon le contexte, il faut réaliser des expériences pouvant exposer des différences selon des aspects variés de qualité. Nous abordons des techniques et des analyses pour évaluer de manière comparative des résultats trouvés par ces méthodes dans l'état de l'art sur de très grands jeux de données. Plus précisément, une étude approfondie sur le temps de calcul empirique de chaque méthode a été présentée avec pour objectif de prédire le temps de calcul nécessaire en fonction de taille de réseau à traiter. D'autres analyses sur la distribution de taille de communautés caractérisant les méthodes de détection sont également abordées en comparaison avec des analyses basant sur des mesures de validation traditionnelle (tel que des variantes de l'information mutuelle). Nous proposons en plus une nouvelle mesure de co-performance qualifiant des corrélations entre les méthodes de détection selon leur aptitude à exposer une certaine qualité. La méthodologie proposée permet aux analystes de se munir des informations nécessaires aux choix d'une ou plusieurs méthodes qui leur

conviennent selon le contexte (Dao, Bothorel, and Lenca, 2018c), (Dao, Bothorel, and Lenca, 2018b).

Ces trois axes principaux seront détaillés dans les Section 0.3, 0.4 et 0.5 pour la caractérisation, l'évaluation et la comparaison des structures communautaires respectivement. Mais en tout premier lieu, nous dédions la Section 0.2 à une brève introduction du problème de détection de structure communautaire, aux enjeux dans l'étude de structure communautaire ainsi que quelques approches principales et les méthodes de l'état de l'art prises en compte dans les analyses suivantes.

0.2 La détection de structures communautaires

0.2.1 Définition de problème

Étant donné un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ composé d'un ensemble de $n = |\mathcal{V}|$ sommets (ou noeuds) et d'un ensemble de $m = |\mathcal{E}|$ liens (ou arêtes) étant des paires de sommets, l'objectif de la détection de structures communautaires (ou détection de communautés) est de trouver une partition $\mathcal{P} = (\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_k)$ des sommets du graphe, dans laquelle chaque communauté $\mathcal{C}_i$ représente un sous-graphe densément connecté. Dans cette thèse, nous nous intéressons particulièrement à des problèmes de détection de structures communautaires sur des graphes non-orientés et non-pondérés. Autrement dit, des liens entre des sommets n'ont pas d'ordre et sont tous égaux. Ces graphes peuvent être représentés mathématiquement par des matrices d'adjacences binaires et symétriques A dont chaque élément a_{ij} représente la présence ($a_{ij} = 1$) ou l'absence ($a_{ij} = 0$) d'un lien entre deux sommets i et j. La Figure 1 illustre le problème de détection de structures communautaires qui peut être considéré comme un processus de réorganisation des lignes et des colonnes d'une matrice d'adjacence de manière à ce que les valeurs non-nulles établissent des blocs sur la diagonale de la matrice[2].

La performance d'une méthode de détection est souvent évaluée à travers des fonctions des qualités $\mathcal{Q}$ qui associent des indices de qualité à toute partition $\mathcal{P}$ d'un graphe $\mathcal{G}$ selon certains critères conformant à différentes notions de structure communautaire. Ces fonctions de qualité prennent en compte des informations telles que des densités de liens à l'intérieur et entre les communautés, des homogénéités stochastiques des connexions des sommets dans les communautés ainsi que des qualités représentant des processus dynamiques qui ont lieu dans les communautés, etc. Parmi plusieurs fonctions de qualité, la plus communément utilisée est la *modularité* mesurant la différence entre la fraction de liens observés à l'intérieur des communautés et cette fraction dans un graphe associé dont la structure communautaire est démolie en réservant la distribution des degrés des sommets (appelé *null model* en anglais) (Newman and Girvan, 2004). Dans le cas où la structure communautaire d'un graphe est connue, la performance d'une méthode est déterminée par la ressemblance entre la structure découverte et cette structure (appelée la vérité terrain). Pourtant, en réalité, la partition attendue est souvent inconnue et il peut y avoir plusieurs solutions significatives correspondantes chacune à des besoins spécifiques. C'est pour cette raison que depuis deux dernières décennies, plusieurs méthodes de détection ont été proposées, chacune avec différents mécanismes et parfois

[2]Cette équivalence est susceptible d'une généralisation de concept provenant d'une redéfinition du problème de détection de communautés ces dernières années. Cependant, dans cette thèse, on parle de la notion primitive de structure communautaire considérée par la plupart de méthodes de détection.

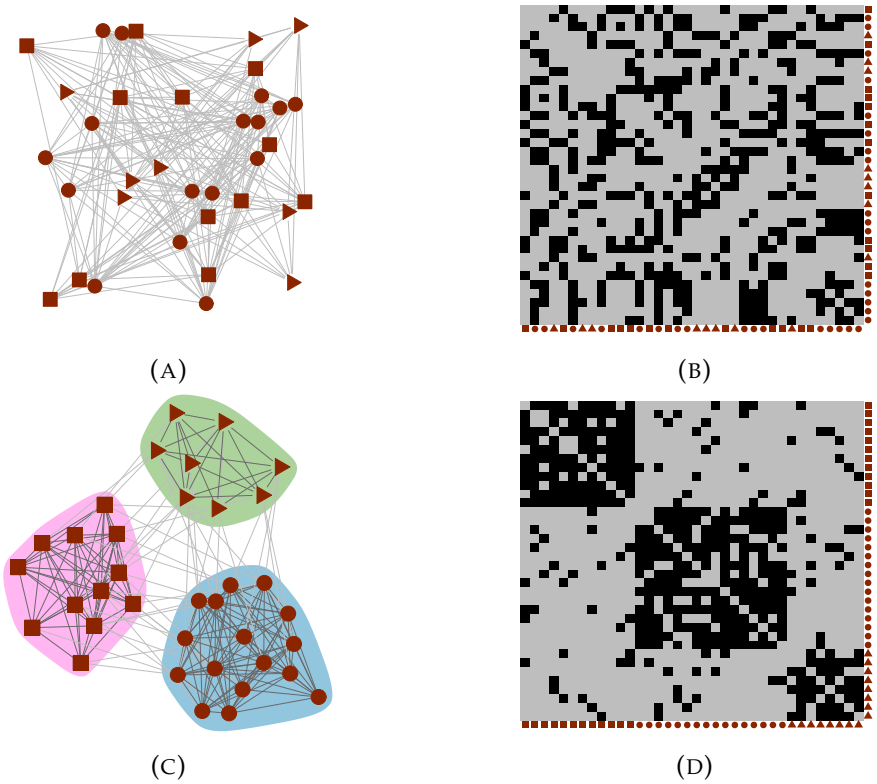


FIGURE 1: (A) Un graphe se composant de 36 sommets et 217 liens avec une structure communautaire, des noeuds ayant la même forme sont supposées d'appartenir à une même communauté. (B) La matrice d'adjacence du graphe avec un ordre aléatoire. (C) Des noeuds sont regroupés dans des communautés densément connectées. (D) La matrice d'adjacence du même graphe réordonnée pour faire émerger la structure communautaire.

différentes fonctions d'objectifs (Fortunato and Hric, 2016). Pourtant, puisque la détection de structure communautaire est un problème mal-défini[3], ce n'est pas facile de pouvoir déterminer des méthodes appropriées dans des différents contextes.

Dans cette thèse, nous envisageons d'examiner les performances de plusieurs méthodes sur de multiples aspects de qualité par une approche empirique afin de montrer leurs performances dans des cas réels. Dans cette vision, nous sommes exposés à plusieurs défis scientifiques dont les plus marquants sont cités ci-après.

0.2.2 Les défis majeurs

- A l'heure actuelle, comme il n'y a pas de consensus sur la définition de la notion de structure communautaire elle-même, cela conduit à de multiples confusions dans l'évaluation ainsi que dans l'utilisation des techniques de détection des communautés dans des cas applicatifs concrets. A moins que le contexte ne soit bien précisé, un seul benchmark d'évaluation ne suffirait pas à démontrer plusieurs aspects des structures communautaires. Bien déterminer des métriques clefs à examiner n'est pas une tâche triviale.

[3]Ce qui veut dire qu'il n'y a ni objectif clair, ni processus formel pour trouver des solutions, ni solution optimale.

- La détection de structures communautaires est un domaine qui évolue rapidement depuis ces dernières années. De nombreux efforts ont été consacrés à résumer le développement de ce sujet dans le littérature (Fortunato, 2010), (Coscia, Giannotti, and Pedreschi, 2011), (Orman, Labatut, and Cherifi, 2012), (Yang, Algesheimer, and Tessone, 2016), (Agreste et al., 2017). Pourtant, la plupart d'entre eux se focalisent soit sur des aspects théoriques afin d'exposer et d'expliquer les différents comportements des méthodes, soit sur des contextes bien déterminés en utilisant des modèles de graphes artificiels avec des structures communautaires bien connues. L'étude pragmatique des méthodes sur des réseaux réels est peu connue et moins investie. Une évaluation par une approche empirique nécessite un grand nombre de traitements, de mesures, d'analyses sur des jeux de données variés et potentiellement grande échelle.

- Plusieurs méthodes visent à trouver des partitions optimisant des fonctions objectifs, ce qui est souvent un problème de type NP-difficile. Par conséquent, les mécanismes employés sont souvent heuristiques et non-déterministes. Les résultats trouvés par une méthode sur un graphe donné peuvent être très différents d'un calcul à l'autre. Il est donc important de déterminer des critères qui caractérisent bien les comportements des méthodes.

0.2.3 Méthodes de détection

Nous allons présenter ici les principales approches et méthodes qui ont été proposées dans l'état de l'art. Bien que la liste prenne en compte les méthodes les plus répandues, elle est pourtant non exhaustive. Nous rappelons que notre objectif final n'est pas de résumer les récents développements, mais de proposer des expériences pouvant aider les analystes à identifier les méthodes qui leur conviennent.

Approche séparative

L'idée principale des méthodes de cette approche est d'essayer de scinder le graphe en question en plusieurs communautés en supprimant progressivement les liens reliant des communautés distinctes. Basée sur le concept que des sommets dans la même communauté sont plus densément connectés, les méthodes de cette approche identifient des liens inter-communautaires et les retirent un à un pour faire apparaître des composantes connexes du graphe qui constituent des candidats pour des communautés du graphe. On peut citer quelques méthodes bien connues dans cette famille telles que celle de (Girvan and Newman, 2002) basée sur la centralité d'intermédiarité ou celle de (Radicchi et al., 2004) basée sur le clustering d'arêtes.

Approche optimisation de modularité

La modularité est une fonction qui mesure la qualité de structure communautaire d'un graphe (Newman and Girvan, 2004). Elle est souvent utilisée comme une fonction objectif dans plusieurs méthodes. Le principe de ces méthodes est de chercher des partitions qui maximisent la différence entre la fraction de liens intra-communautaires et une fraction attendue si les liens avaient été distribués de manière aléatoire. Dans cette famille, on analyse l'algorithme glouton (Clauset, Newman, and Moore, 2004), une méthode muti-échelle communément appelée *Louvain* (Blondel et al., 2008) et une approche spectrale utilisant une matrice de modularité (Newman, 2006).

Approche utilisant des processus dynamiques

Des méthodes utilisent des processus aléatoires dans les graphes souvent représentés par des marches aléatoires pour estimer des structures communautaires. Le comportement stochastique des marches aléatoires sur un graphe étant fortement lié à la structure du graphe, elles aident à identifier des sous-graphes densément connectés. Quelques méthodes populaires de cette approche peuvent être citées telles ques *Walktrap* (Pons and Latapy, 2005), *Infomap* (Rosvall and Bergstrom, 2008).

Approche statistique

Ces dernières années, la communauté scientifique s'intéresse aux méthodes qui essaient de reconstruire des paramètres latents d'un modèle génératif basé sur la distribution de liens dans un graphe observé. Les méthodes dans cette approche considèrent que la probabilité que deux sommets dans un graphe soient connectés dépend des communautés auxquelles ils appartiennent. Ensuite, on calcule la vraisemblance qu'un graphe soit généré par un ensemble de paramètres pour déduire la structure communautaire. Parmi les variantes de cette approche proposées, on analyse les méthodes basées sur des modèles stochastiques comme *Infomod* de (Rosvall and Bergstrom, 2007) ou (DC)SBM de (Riolo et al., 2017) ainsi qu'une méthode basée sur la signification statistique des communautés appelée *Oslom* de (Lancichinetti et al., 2011).

Autres approches

Il existe de nombreuses autres approches pour la détection de structures communautaires. On peut citer des méthodes basées sur les mécanismes de verre de spins (Reichardt and Bornholdt, 2006), sur un processus simulant la propagation de l'information (Raghavan, Albert, and Kumara, 2007), (Xie and Szymanski, 2012) ou celles qui utilisent une approche hybride employant des informations globales et locales de connectivité afin d'identifier des communautés dans un graphe (Meo et al., 2014). Les méthodes étudiées dans cette thèse sont résumées dans Tableau 1.

Approche	**Référence**	**Label**
Séparative	(Girvan and Newman, 2002)	GN
	(Radicchi et al., 2004) $(g = 3)$	RCCLP-3
	(Radicchi et al., 2004) $(g = 4)$	RCCLP-4
Optimisation de modularité	(Clauset, Newman, and Moore, 2004)	CNM
	(Blondel et al., 2008)	Louvain
	(Newman, 2006)	SN
Processus dynamiques	(Pons and Latapy, 2005)	Walktrap
	(Rosvall, Axelsson, and Bergstrom, 2009)	Infomap
Inférence statistique	(Rosvall and Bergstrom, 2007)	Infomod
	(Lancichinetti et al., 2011)	Oslom
	(Riolo et al., 2017)	(DC)SBM
Autres méthodes	(Reichardt and Bornholdt, 2006)	RB
	(Raghavan, Albert, and Kumara, 2007)	LPA
	(Xie and Szymanski, 2012)	SLPA
	(Meo et al., 2014)	Conclude

TABLE 1: Un résumé des méthodes de détection de structures communautaires incluses dans nos études.

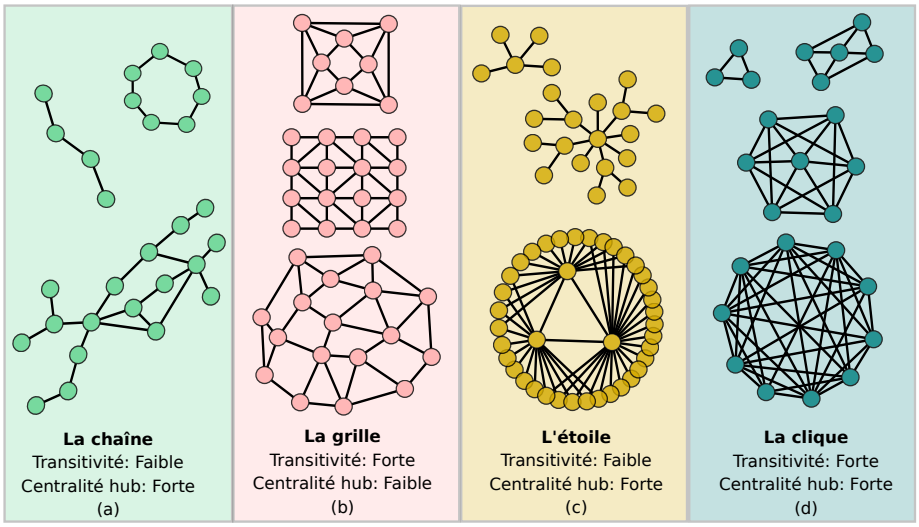


FIGURE 2: Une illustration des modèles topologiques d'interactions: (a) La chaîne, (b) La grille, (c) L'étoile, (d) La clique

0.3 La caractérisation de structure communautaire

Étant donné que le concept d'une *"bonne"* structure communautaire est lié au contexte dans lequel un graphe est étudié, il est important de discerner différentes aspects structurels qui existent au sein des graphes. Nous allons présenter dans cette section une caractérisation des communautés structurelles identifiables dans des réseaux empiriques. La caractérisation présentée par la suite vise à répondre aux questions: *"À quoi ressemblent les structures communautaires dans des réseaux réels?"* et *"Y-a-t'il des différences majeures entre les structures communautaires dans réseaux réels appartenant à des domaines différents?"*.

Nous proposons une caractérisation basée sur des indices topologiques quantifiant la présence de certains motifs d'interactions entre des sommets dans une communauté. En réalité, puisque plusieurs métriques mesurant de différentes notions de qualité structurelle des communautés sont corrélées (Yang and Leskovec, 2013), il est possible de choisir quelques métriques représentatives afin de caractériser des communautés structurelles. Le choix des métriques qui conviennent le mieux pour distinguer des communautés peut être varié selon le contexte, mais d'une manière générique, nous nous intéressons aux métriques dont la combinaison nous fournit des informations sur des topologies d'interaction entre sommets. Une analyse profonde de correlation entre de différentes métriques nous permet d'identifier des modèles d'interactions principales entre les sommets d'une communauté grâce aux deux propriétés structurelles appelant *la transitivité* et *la centralité hub* (Dao, Bothorel, and Lenca, 2018a). La transitivité (communément connue par son nom *"clustering coefficient"* en anglais) quantifie la probabilité que deux voisins d'un sommet soit connectés. Un bon score de transitivité implique une profusion de structures triangulaires dans la communauté. Tandis que la centralité hub mesure la présence des sommets fortement connectés au sein d'une communauté. Un bon score de centralité hub signifie une forte structure centralisée. Une combinaison des scores de transitivité et de centralité hub nous conduit à des topologies illustrées sur la Figure 2. Comme nous pouvons constater, ces deux mesures sont complémentaires et apportent une information topologique sur des communautés.

Si l'on représente une communauté par un point dans un système des coordonnées déterminé par les deux dimensions que sont la transitivité et la centralité hub, la

position de ce point peut nous aider à identifier la topologie de la communauté. En plus, cette description nous permet d'associer les familles topologiques à des modèles génératifs de graphes comme illustrée sur la Figure 3. Basé sur l'idée que des réseaux réels sont construits par un certain mécanisme (ou processus), cette technique de représentation agit d'une manière intuitive pour associer un réseau et ses structures des communautés aux différents modèles génératifs de graphes.

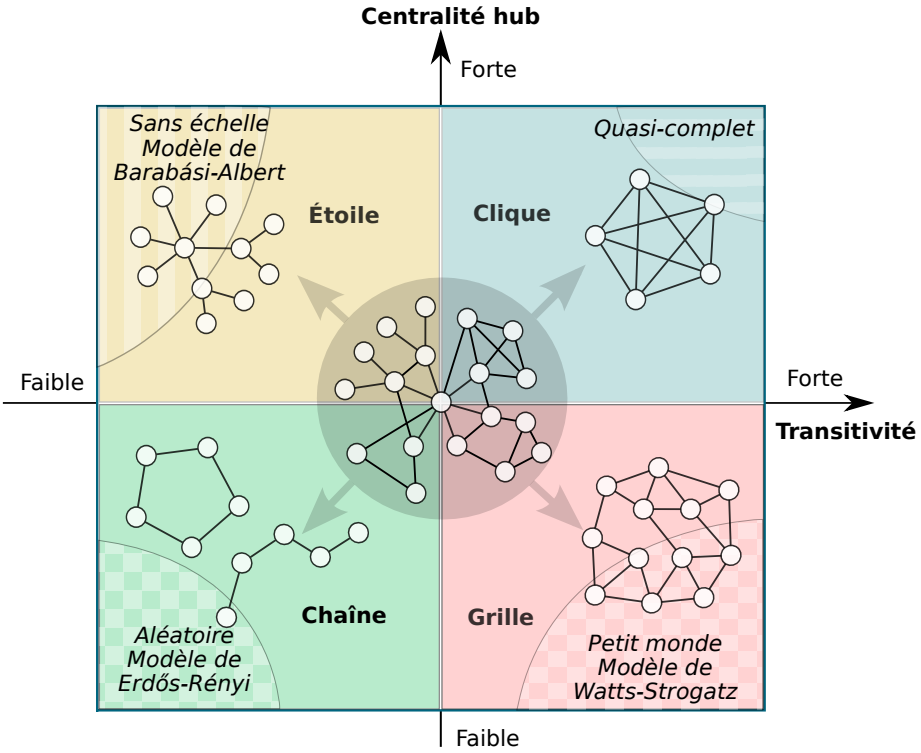


FIGURE 3: Des modèles génératifs de graphe associés aux structures topologiques

Pour illustrer comment des structures des communautés calculées dans des réseaux réels se situent dans un tel espace à deux dimensions, nous utilisons un corpus contenant une centaine de réseaux comme illustré dans le Tableau 2. L'objectif de notre expérience est de montrer comment la caractérisation par des familles topologiques peut nous renseigner sur les différences des communautés issues de réseaux provenant de différents domaines d'études, comme par exemple les réseaux biologiques, les réseaux sociaux, etc. La Figure 4 démontre les distributions des communautés découvertes dans chaque famille de réseaux. Comme l'on peut le constater, les structures des communautés dans ces domaines sont bien distinctes. Par exemple, dans les réseaux de communication et les réseaux sociaux, la plupart des communautés ont de très forts hubs et il y a peu de connexions triangulaires. Ce modèle d'interaction est très proche avec le modèle d'attachement préférentielle (Barabási and Albert, 1999). Tandis que sur les réseaux d'information, il existe une haute fréquence de groupes dont les sommets sont connectés de manière très compacte comme ceux de modèle petit monde (Watts and Strogatz, 1998). On constate bien une diversité structurelle de motifs d'interactions dans les réseaux réels que des modèles de graphes théoriques n'arrivent pas forcement à imiter. Cette diversité implique également une nécessité de construire des modèles génératifs décrivant mieux des graphes ayant des structures communautaires hétérogènes.

Domaine	Nb.	Sommets	Liens	Exemple
Biologie	7	1860	10763	Protein, levure
Communication	9	39595	195032	Email, forums
Information	25	38358	159812	Citation, Amazon
Sociale	37	6888	49666	Facebook, Youtube
Technologie	19	18431	48494	Internet, P2P
Divers	11	4298	49033	Ecologie, synthetique
Total	108	1.99M	9.08M	

TABLE 2: Un résumé des réseaux analysés. **Nb.**: Le nombre de réseaux concernés, **Sommets**: Nombre moyen de sommets dans les réseaux, **Liens**: Nombre moyen de liens dans les réseaux, **Total**: Le nombre total de tous les réseaux et leurs sommets et liens. Source: `http://networkrepository.com` (Rossi and Ahmed, 2015), `http://konect.uni-koblenz.de` (Jerome, 2013), `http://snap.stanford.edu` (Leskovec and Krevl, 2014)

.

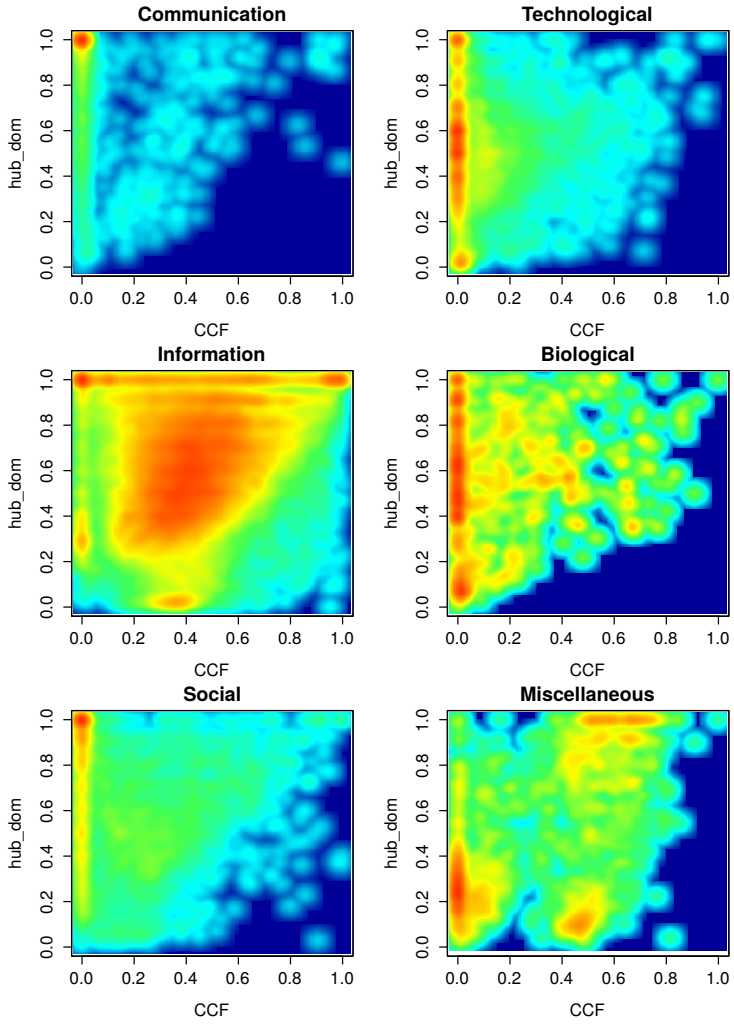


FIGURE 4: Les distributions des communautés identifiées dans des réseaux de différents domaines. Sur l'abscisse et sur l'ordonnée de chaque sous-figure se trouve la transitivité (CCF) et la centralité hub (hub_dom) respectivement. De haut en bas, de gauche à droite (a) Communication, (b) Technologie, (c) Information, (d) Biologie, (e) Sociale, (f) Divers.

0.4 L'évaluation de structure communautaire

Une des techniques pour évaluer la performance d'une méthode de détection de structure communautaire sur un graphe est de comparer la structure qu'elle a trouvée avec une vérité terrain. Si la structure découverte par la méthode correspond bien à la vérité terrain, nous concluons que la méthode fonctionne bien. À l'inverse, si les deux structures ne correspondent pas, nous parlons d'une mauvaise performance. En réalité, la vérité terrain n'existe pas dans une application de type non supervisée, autrement dit, nous ne disposons pas de structure communautaire à identifier[4]. Souvent, des métadonnées provenant des identifiants des sommets sont utilisées en tant que vérités terrains. Cette utilisation conduit souvent à des conclusions non pertinentes sur la performance de détection (Peel, Larremore, and Clauset, 2017).

À travers plusieurs expériences, nous constatons que ces communautés métadonnées fournies avec certains réseaux réels ne sont pas structurellement bonnes. Cela veut dire également qu'en appliquant des méthodes de détection de structure communautaire, nous pouvons trouver de meilleures structures communautaires.

Graphe	**N**	**E**	$\widehat{\mathbf{k}}$	$\bar{\boldsymbol{\alpha}}$	**CCF**	**Communautés métadonnées**
zachary	34	78	4.6	-2.2	0.26	Séparation du club
football	115	613	10.7	-9.1	0.41	Ligues de champions
polblog	1222	16714	27.4	-3.7	0.23	Partis politiques
youtube	39841	224235	11.3	-2.8	0.06	Groupes d'abonnement
livejournal	84438	1521988	36.1	-2.4	0.77	Groupes d'abonnement
dblp	317080	1049866	6.6	-3.3	0.31	Lieus de publication
amazon	334863	925872	5.5	-3.6	0.21	Catégories de produits

TABLE 3: Une description de réseaux réels avec des communautés métadonnées. **N** - nombre de sommets, **E** - nombre de liens, $\widehat{\mathbf{k}}$ - degré moyen, $\bar{\boldsymbol{\alpha}}$ - l'exposant estimé de la séquence de degré selon une loi de puissance, CCF - Coefficient de clustering. Source: http://www-personal.umich.edu/~mejn/netdata/ et http://snap.stanford.edu/data/

À titre exemple, nous évaluons des communautés structurelles identifiées par les méthodes présentées dans la Section 0.2 sur les quelques réseaux réels avec des communautés identifiées par des métadonnées (Tableau 3). Nous considérons quelques mesures de qualité structurelle dans notre analyse:

- *La densité*: mesure la fraction entre le nombre de liens existants dans une communauté et le nombre de liens maximal que l'on peut construire entre ses sommets.
- *La compacité*: suggère qu'une bonne communauté doit avoir une forte densité et un faible diamètre pour que les sommets soit facilement accessibles l'un avec l'autre (Creusefond, Largillier, and Peyronnet, 2015).
- *Le coefficient de clustering*: mesure la fraction entre le nombre de triangles existants et le nombre de triangles maximal que l'on peut construire.

[4]Dans certain contexte, sous réserve d'une hypothèse que le graphe en question soit créé par un modèle théorique dont la structure communautaire est formellement définie, nous pouvons parler d'une vérité terrain déterminée par le modèle.

- *La modularité de communauté*: mesure la différence entre la fraction de liens dans une communauté et cette fraction attendue si les liens avaient été distribués aléatoirement.
- *L'embeddedness*: valorise l'idée que les voisins d'un sommet dans une communauté devrait mieux appartenir à cette communauté.
- *La separabilité*: est basée sur le concept qu'une bonne communauté doit être bien separée (faiblement connectée) des autres communautés du graphe.

Méthode	**Sep**	**Emb**	**Den**	**Com**	**CCF**	**Q**
CNM	6.18	1.46	2.79	1.79	0.99	2.71
Louvain	**11.01**	**1.50**	2.68	**6.02**	0.94	**12.67**
Infomap	2.24	1.26	3.34	0.96	0.90	0.75
Walktrap	1.87	1.19	**3.35**	0.78	0.93	0.65
Oslom	1.69	1.10	1.29	1.21	1.05	0.83
LPA	2.72	1.40	1.84	1.15	1.11	1.06
SLPA	5.34	1.39	2.54	1.19	1.03	0.84
Conclude	1.42	1.13	2.52	0.72	**1.33**	0.63
Ratio moyen	4.06	1.30	2.54	1.73	1.03	2.52

TABLE 4: Ratio de qualité entre des communautés structurelles et des communautés métadonnées. **Sep** - *La separabilité*, **Emb** - *L'embeddedness*, **Den** - *La densité*, **Com** - *La compacité*, **CCF** - *Le coefficient de clustering*, **Q** - *La modularité de communauté*. Le meilleur ratio de chaque qualité est mis en gras.

Nous mesurons les scores de qualité définis par ces derniers métriques sur les communautés structurelles découvertes par les différentes méthodes sur les réseaux présentés dans le Tableau 3 et les comparons avec ceux des communautés métadonnées. Le Tableau 4 présente les ratios entre les scores des communautés structurelles et des communautés définies par des métadonnées. Nous pouvons facilement remarquer qu'il y a des améliorations significatives de toutes les qualités analysées sur les communautés découvertes par les méthodes de détection (une ratio > 1 signifie une amelioration de qualité). Ce résultat est aussi vérifié dans les graphes artificiels générés par le modèle de LFR (Lancichinetti, Fortunato, and Radicchi, 2008) dont les communautés métadonnées sont souvent considérées comme les meilleures solutions (Dao, Bothorel, and Lenca, 2017a). Dans le cas des réseaux réels, ce résultat est explicable. Puisque les communautés métadonnées ne sont pas construits basées sur des informations structurelles mais seulement sur des attributs des sommets, par exemple des catégories de produits sur Amazon, il est peu probable que leurs sommets soient systématiquement densément connectés. Sur des réseaux LFR, les communautés vérité terrain sont générées en assurant une ratio entre le nombre de liens intra-communautés et inter-communautés (mixing parameter en anglais). Dans cette manière de configuration, il est toujours possible de diviser une communauté métadonnées pour augmenter la densité, la compacité ou au contraire fusionner plusieurs communautés pour augmenter la modularité, etc.

Notre évaluation a montré que la plupart de méthodes arrivent à trouver des partitions ayant de meilleures structures par rapport aux métadonnées. De plus, nous apportons des preuves pour démontrer l'inadéquation de l'utilisation des métadonnées en tant que vérité terrain dans l'évaluation de la performance des méthodes

de détection. Nos analyses ont indiqué également que, étant donnée une fonction de qualité, nous pouvons toujours identifier une méthode qui expose une meilleure performance que d'autres (Dao, Bothorel, and Lenca, 2017a).

0.5 La comparaison des méthodes de détection

Dans cette section, nous nous concentrons sur des analyses comparatives entre différentes méthodes de détection de structures communautaires. Les méthodes concernées ont été introduites dans le Tableau 1 de la Section 0.2. Nous considérons plusieurs aspects: le temps d'exécution empirique en fonction de la taille des réseaux, la distribution des tailles des communautés, la similarité entre partitions, etc (Dao, Bothorel, and Lenca, 2018b). Nous soulignons ci-dessous quelques analyses importantes.

0.5.1 Temps de calculs

Le temps d'exécution est un facteur important à considérer lors du choix d'une algorithme surtout pour des applications temps réel. Des méthodes de détection de structures communautaires sont souvent proposées avec des estimations de complexité, ces estimations fournissent pourtant peu d'information sur le temps de calcul réel qui peut fortement varié entre deux méthodes de même complexité. Nous mesurons les durées de temps de calcul des méthodes dans le Tableau 1 pour détecter des communautés sur les réseaux présentés dans le Tableau 2. Ensuite, nous estimons le temps requis par chaque méthode en fonction de la taille du réseau en utilisant une méthode de régression locale (Cleveland, 1979).

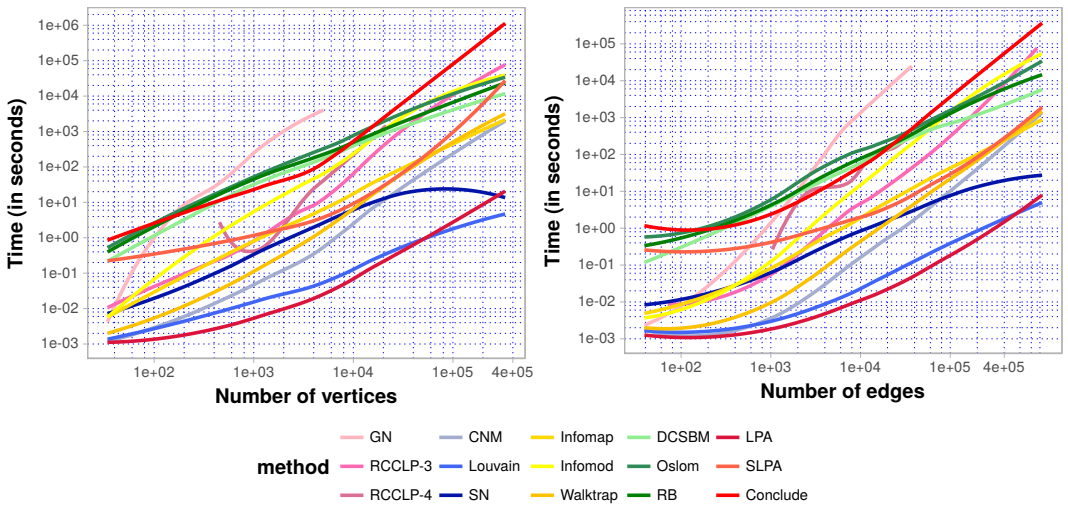


FIGURE 5: Une estimation de temps d'exécution en fonction de taille de réseaux (nombre de sommets et nombre de liens).

La Figure 5 illustre le temps estimé[5] requis par chaque méthode en fonction du nombre de sommets (à gauche) et du nombre de liens (à droite). Nous pouvons facilement constater que, entre la méthode la plus lente et la méthode la plus rapide, pour un graphe de même taille, la consommation en temps peut varier d'une

[5]Temps estimé basé sur les implémentations de bibliothèque *igraph* dans la plupart de cas et provenant des auteurs dans d'autres cas. Les paramètres par défaut des implémentations sont utilisés.

manière très sévère. Il est donc crucial de considérer le temps de calculs pratique pour le choix d'une méthode de détection, surtout pour des applications temps réel. Notre analyse fournit une prédiction fiable et informative du temps consommé en fonction de taille des données pour des analystes ayant besoin de déployer des méthodes de détection des communautés.

0.5.2 Distribution de taille de communauté

Lors de la décomposition d'un graphe en plusieurs sous-graphes, on s'intéresse à savoir combien de communautés sont produites et quelles sont les tailles de ces communautés[6]. Cette question est équivalente à la question de combien de clusters sont identifiés et quelles sont leurs tailles dans un problème de clustering traditionnel. Dans une toute première idée de la détection des communautés, il est important qu'une méthode puisse proposer d'une manière automatique le nombre de communautés dans un graphe. Puisque les méthodes ont des stratégies différentes pour identifier automatiquement les structures communautaires d'un graphe et fournissent souvent des uniques répartitions, on peut les distinguer par une analyse sur les nombres de communautés identifiées. Cependant, nous nous rendons compte que deux méthodes produisant un nombre équivalent de communautés peuvent répartir des sommets d'un graphe par des manières très distinctes. Par conséquent, nous nous intéressons à analyser la distribution de taille de communauté qui est directement liée au nombre de communautés caractérisant les méthodes de détection. De plus, une analyse compréhensive en distribution de taille de communauté d'une méthode est informative car elle prédit l'information sur d'autres qualités de structure communautaire en question comme la modularité, la densité, la conductance, etc qui sont corrélées avec les tailles des communautés.

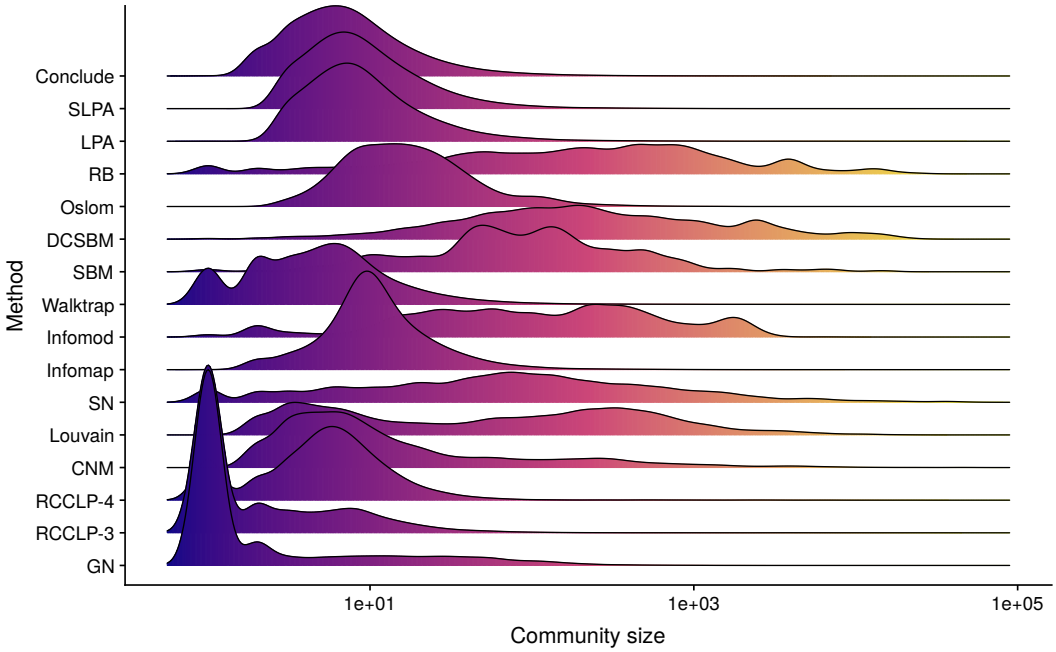


FIGURE 6: Les distributions des tailles de communautés détectées par chaque méthode de détection de structure communautaire.

[6]Lors qu'on dit *taille d'une communauté*, il s'agit du nombre de sommets contenant dans cette communauté parfois appelé *volume de communauté*.

Concrètement, la Figure 6 montre la distribution des tailles des communautés découvertes par toutes méthodes présentées dans la Section 0.2 sur les graphes résumés dans le Tableau 2. Les fonctions de densité ont été estimées à partir des fonctions de masse associées en utilisant un estimateur par noyau de type Gaussien. Cette démonstration affiche plusieurs stratégies de division parmi les méthodes analysées. Nous montrons qu'une classification des méthodes se basant sur cet aspect nous permet d'exposer des différences ou des similarités qu'une évaluation par une technique traditionnelle (comme l'information mutuelle ou l'indice de Rand) n'arrive pas à faire. En effet, il est possible de définir une fonction de similarité entre deux méthodes pour comparer les distributions qu'elles produisent. Deux méthodes sont considérées similaires si elles détectent des communautés de tailles comparables. À partir de cette idée, nous définissons la similarité de deux méthodes comme étant l'aire commune sous les courbes de deux distributions associées (Dao, Bothorel, and Lenca, 2018c).

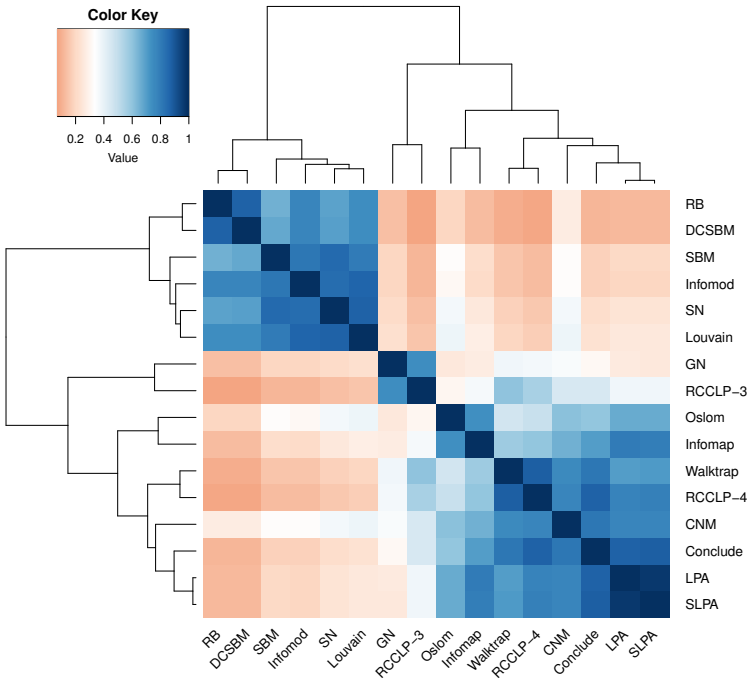


FIGURE 7: La similarité entre les méthodes étudiées en terme de taille de communautés qu'elles produisent.

La Figure 7 montre les estimations de scores de similarité par cette dernière définition. On distingue nettement trois à quatre stratégies de répartition parmi les méthodes étudiées représentant une différence fondamentale dans la manière qu'elles considèrent les structures communautaires des graphes dans notre jeu de données. Un rapport plus détaillé sur cette analyse peut se trouver dans (Dao, Bothorel, and Lenca, 2018b).

0.5.3 Autres analyses

Nous avons réalisé de nombreuses analyses afin de montrer différents aspects de structure communautaire qui font la nuance entre les méthodes de détection. Bien qu'une liste limitée de mesures ne peut pas tout démontrer, nous essayons de cerner les parties les plus fondamentales. Concrètement, nous avons analysé la similarité entre ces derniers méthodes par une approche traditionnelle en utilisant des

métriques de validation telles que l'information mutuelle (normalisée et ajustée) (Vinh, Epps, and Bailey, 2010), ou l'indice de Rand (ajusté ou non). Nous avons présenté également une nouvelle *mesure de co-performance* quantifiant la similarité entre des méthodes en termes de capacité à repérer des communautés montrant une certaine qualité. Plusieurs métriques des qualités ont été étudiées telles que la *modularité de Newman-Girvan*, la *modularité densité*, la *Z-modularité*, la *Significance*, la *Surprise* (Dao, Bothorel, and Lenca, 2018b).

0.6 Conclusions et discussions

Le choix d'une méthode de détection de structure communautaire est un problème très ouvert dont une solution claire n'existe pas à moins que le contexte et la motivation d'analyse soient très bien déterminés. Il faut préciser que l'interprétation du résultat d'une répartition est aussi difficile et discutable que la détection elle-même. Dans cet état d'esprit, nous avons réalisé de nombreuses analyses afin d'aider des utilisateurs à interpreter, caractériser différentes types de structures et évaluer différentes méthodes. Même quand la détection de structure communautaire est un domaine qui évolue rapidement ces dernières et même si la notion de structure communautaire est perçue de manière très variable, les approches proposées dans cette thèse restent valides puisqu'elles servent qu'à démontrer des informations structurelles. Ces informations peuvent aussi assister des utilisateurs potentiels à réduire le nombre d'analyses nécessaires à réaliser afin de traiter uniquement des méthodes dont les perspectives leur conviennent.

Les études réalisées dans cette thèse ne permettent cependant pas de répondre directement à quelle méthode utiliser dans quel contexte, ce qui reste malgré tout une tâche ambitieuse. Pourtant, de récents développements ont éclairci de plus en plus d'aspects de structure communautaire qui peuvent être considérés (Schaub et al., 2017). Dans ces travaux, le problème de detection de structure communautaire se décompose en quatre perspectives: minimization de taille de coupe (min cut en anglais), problème de clustering basé sur la densité de liens, regroupement stochastique des sommets ou identification des groupes dynamiques. Cette décomposition du problème est fondamentale pour des analystes voulant analyser leurs réseaux car elle précise de différents objectifs dont les solutions optimales sont très distinctes. Autrement dit, un analyste qui cherche à minimiser la taille de coupe d'une répartition ne va pas avoir besoin de prendre en compte des méthodes stochastiques ou dynamiques par exemple. Dans le cas où il réfléchit entre deux méthodes de même perspective, la tâche de choisir une méthode optimale peut revenir à un problème de type NP-difficile. À ce moment-là des analyses empiriques comme celles introduites ci-dessus deviennent significatives. Puisque la plupart des méthodes proposées dans le problème de détection de structure communautaire appartiennent à une perspective de clustering, nous avons focalisé nos analyses sur cette direction.

Même quand une vérité terrain n'existe pas, ce qui est généralement le cas, nous avons obtenu plusieurs constats significatifs à travers des analyses guidant à des choix appropriés. Par exemple, pour une application temps réels, l'utilisation de la méthode *Louvain* ou des variantes de *LPA* seraient favorables grâce à leur scalabilité. Si le perspective est de chercher des groupes de tailles homogènes ayant une forte transitivité à l'intérieur, *Infomap* est une bonne méthode à employer. Afin d'identifier des groupes de sommets qui sont stochastiquement similaires (en terme de connexion intérieure et extérieure de communautaire), l'approche utilisant des modèles de blocs stochastiques pourrait être considéré en premier lieu.

Finalement, l'analyse et l'interprétation de la structure d'un réseau exige une connaissance et de l'expertise sur le domaine en question. La détection des communautés est seulement une étape dans une suite d'analyses séquentielles qui facilite des travaux qui la suivent. Par conséquent, l'évaluation de la performance d'une partition ne peut pas explicitement être considérée sans prendre en compte l'objectif ultime et les différentes modélisations réalisées au cours des autres étapes. La fin de cette thèse est plutôt une ouverture de nouveaux problèmes à considérer et à résoudre afin de démystifier la détection des communautés et en faciliter la prise en main à des analystes, qu'ils soient décideurs ou chercheurs, spécialistes en analyse de réseau ou non, avec des problématiques algorithmiques ou économiques, sociales, etc.

Chapter 1

Introduction

The study of *complex systems* can be considered in a brief locution: *interaction learning*. If we take a look, our world is full of interactions such that everyone can tell inexhaustibly hundreds of examples and the only limit that exists is, unluckily, our imagination. Think about the complexity of social relations between friends, families, professionals; communications between any single molecule with the others in every single living thing; tonnes of data exchanging between computers and servers though millions of navigating packages in an exploding Internet system, etc. There is too much information that we can study from these interactions in order to understand and to explain the functionality of real world systems. This abundancy is, in fact, at the same time an opportunity and a great challenge.

In a methodological reductionism point of view, if it is complicated to understand the behaviors of components of real-world systems, it must be exponentially more challenging to comprehend phenomena produced from their complex organized interactions. However, ignoring the intrinsic complexity of single individuals, emerge fascinating collective patterns that could not be explained separately by examining systems at individual levels. Therefore, there is a necessity to find appropriate scientific tools that could help to understand complex systems. And from that demand, with an appealing philosophical approach, network science becomes naturally an eligible and legitimate solution for discovering real-world complex interactions.

The study of complex systems in a modern network science approach can be briefly summarized into three principle mainstreams: *network discovering*, *network modeling* and *processes on networks*:

- In network discovering, people study different algorithms and methods to understand network structures in different levels: *microscopic*, *mesoscopic* and *macroscopic* structures (Reichardt, Alamino, and Saad, 2011). Exploring the microscopic level of a network consists in studying properties of nodes through their interaction rules with the others, such as measuring centrality, transitivity, reciprocity, etc. (Newman, 2010). On the other end, the macroscopic structure of a network discloses information in a global view resulting from microscopic rules regulated by nodes, such as: average degree, diameter, network spectrum, etc. It is also possible to discover a network in an intermediate level, i.e. mesoscopic level constituted by groups of nodes, large enough so that collective properties can be reasonably discoursed and small enough so that there can be a representative constituent member for each one (Porter, Onnela, and Mucha, 2009).

- In network modeling, researchers are interested in representing real-world systems through networks characterized by different statistical rules. There is a close relation between network modeling and network discovering, such that

quantifying different properties of observed networks allows to better develop theoretical models. Indeed, it is not difficult to create a network model but creating models that cover well various real-world phenomena is very challenging. The most notable and widely studied network models in the literature of network science that could be mentioned are: *Erdős-Rényi* model (Erdős and Rényi, 1959) commonly known as *random* graphs, *Watts-Strogatz* model (Watts and Strogatz, 1998) commonly known as *small world* graphs and *Barabási-Albert* model commonly known as *scale-free* graphs (Barabási and Albert, 1999). According to a specific context, some models could be preferable than the other in describing complex networks.

- Studying processes on networks is a very appealing domain in network science recently thanks to the availability of sophisticated analysis tools as well as novel techniques that help to collect more efficiently dynamics network data. Since the world is not static, systems inside it also expose different dynamic mechanisms. Many researches aim to explain real life phenomena in social science, biology, information and technology, etc. controlled by different interaction rules on associated networks. Prominent work that could be found on this axis consists in epidemics, resilience on networks, dynamical systems (Masuda and Lambiotte, 2016), etc.

1.1 Context and problems

In this dissertation, we invite readers to be interested in the discovery of mesoscopic structure of networks, widely known as *community structure* in the literature of network science. There are several reasons why one might want to decompose a network into smaller groups of vertices. In accordance with the availability of information as well as the final objective, one could possibly consider different techniques. For instance, community detection with attributes (Yang, McAuley, and Leskovec, 2013), (Bothorel et al., 2015) could be used if information about nodes and/or edges are available or community search (Sozio and Gionis, 2010) when only communities of a portion of nodes need to be queried. In a traditional way, when graph structure is the only available information, community detection is referred to as using algorithms to divide the vertices of a given graph into several groups according to the distribution of edges in the graph (Newman, 2010).

The notion of community could be considered for different aspects in real life and each technique to discover communities has its own attractiveness. In this thesis, we are interested in studying different community detection methods and the characterizations of associated community structures. However, community notions defined by the principle techniques as presented above are not directly comparable since they process different kind of information and have different objectives. Therefore, we restrict ourself in a context where the only available information is the structure of networks. Hence, and from now on, community detection is implicitly understood as stated by the definition of Newman and many others authors (Danon et al., 2005), (Porter, Onnela, and Mucha, 2009), (Fortunato, 2010), meaning structural information characterized by the distribution of edges in a network.

Although showing a high similitude with traditional unsupervised data clustering, community detection methods have just been becoming prosperous in the last

two decades remarked by the invention of modularity (Newman and Girvan, 2004)[1] and the availability of a large volume of networks thanks to the development of Internet and notably the richness of social platforms. Since then, a numerous number of detection algorithms with various approaches have been proposed (Fortunato and Hric, 2016) to resolve this problem, each one with its own mechanism and sometimes with different objective functions [2]. However, this multiplicity of choices also leads to a confusion in deciding which method to choose to automatically discover community structures of a given network as there is no standard choice. Specifically when there is still no consensus on a closed-form expression of community structure. Indeed, some recent researches indicate that no algorithm can globally perform better than all of the others in a general No Free Lunch theorem (Peel, Larremore, and Clauset, 2017). It implies that in some specific contexts, some methods will be better than the others. For that reason, we are interested in investigating different state-of-the-art and well-known community detection methods in order to answer some following questions:

1. What do real world communities in networks look like and how to describe them?
2. How much community detection algorithms are good in detecting community structures?
3. How the structures of communities in different kinds of networks are seen by community detection algorithms?
4. How can we help practitioners to choose an appropriate community detection method corresponding to different criteria?

1.2 Challenges

As community detection is a quite new problem in network science, it has been drawing huge attention in recent years and presents several challenges to our work. Some principle ones that are worth mentioning:

- The researches in finding new community detection methods and evaluation metrics are very active. There are hundreds of algorithms presented each year in conferences and scientific journals making it very challenging to be able to include as many as possible novel representative methods in our experimental study.
- There is no consensus on the formulation of community detection problem [3]. Indeed, community detection is sometimes *decomposed* into many sub-problems such as: vector partitioning problems (Newman, 2006), optimization problems (Duch and Arenas, 2005), (Brandes et al., 2008) or inverse problems (Karrer and Newman, 2011), (Peel, Larremore, and Clauset, 2017) etc. Hence, comparing community detection methods becomes comparing their sub-problems, which is not straightforward.

[1] No one has contributed to the development of the domain of community detection as much as Newman does. Therefore, similarly to many other scientific publications of the same subject, a huge number of references in this thesis are connected to his work.

[2] We provoke a small difference between objective and objective function here, as two methods may have the same objective (finding a community structure) but use different objective functions.

[3] Community detection is widely considered as an ill-defined problem (Fortunato and Hric, 2016), meanings *"it does not have clear goals, solution paths or expected solution"* (Arifin et al., 2017)

- Surveys in the literature normally address theoretical aspects by analytical arguments (Fortunato, 2010), (Coscia, Giannotti, and Pedreschi, 2011) to deduce the properties that are regulated by different detection mechanisms. Empirical work exists (Orman, Labatut, and Cherifi, 2012), (Agreste et al., 2017) but requires a huge number of processing, experiments and analyses, especially when many standards in representing and modeling communities exist.

- Finally, understanding community structures in networks is not a destination, but a long journey of discovery. The more we understand them, the more we need to step back to in order to reevaluate the appropriateness of our objectives and pursue suitable ones.

1.3 Contributions

From the research questions that have been introduced recently in the context of this thesis, after a three-year-long period, approximately 20 thousands experiments on more than a hundred of networks using 16 state-of-the-art and well-known community detection methods were conducted. These methods allow us to discover more than 1.35 million communities, which were evaluated by more than 40 different quality metrics. The outcome of our study gives rise to the following principle contributions:

1. We presented a novel descriptive approach to describe ground-truth community structures in some real world large scale networks (such as DBLP network reflecting the co-authorship relations of scientists or Amazon network illustrating co-purchase products). Our method allows to classify real world community structures that could be classified in 6 different classes characterized by 6 interaction archetypes. This characterization also helps to demonstrate that ground-truth communities in real world networks are not structurally good, and leads to the next contribution (Section 4.2).

2. From the notice that real world communities are not structurally good, we are interested in quantifying how different community detection methods could improve different types of quality of communities on networks. This quantification gives a quick view on how good methods are in discovering communities and which method one should chose in order to extract a given structural quality (Section 4.3).

3. We characterized the structures of communities detected by different methods on a large number of networks using a low-dimensional space. Our study uncovers that networks across different categories including communication, technological, information, biological and social networks show different community structures, which can be described by popular network models in the literature. This discovery could open a possibility to design new models that adapt better networks in different contexts (Section 5.1).

4. We provide a practical study of real computation time of different community detection methods in function of network size. Although theoretical time play an important role in evaluating the scalability of a method, our study provide concretely a good prediction of necessary time which is not always in agreement with theoretical estimates (Section 6.1).

5. We invent a new approach to estimate the similarity of different community detection methods based on the estimated distribution of community sizes that they detect. Our findings show that there are three different partition strategies. According to an expected distribution of community sizes, one could based on our analysis to choose a detection method or target appropriate alternative methods when the favorite methods are not eligible (Section 6.2).

6. Finally, we demonstrated a new quality evaluation paradigm that helps to identify groups of methods that expose similar performance in terms of some qualities. We present an coefficient called co-performance index, that reveals a prediction about how the outcome of a method could help us to guess about the outcome of the others. This contribution helps network practitioners to quickly identify equivalent methods for a given expect quality (Section 6.3 and 6.4).

1.4 Instructional outline

The next parts of this dissertation is organized in the following way:

- Chapter 2 is an introduction about complex networks and graphs. In this chapter, we present briefly many state-of-the-art studies of complex systems in different domains, which draw huge attention in the network science community. Then, some fundamental notions of graphs are presented as a prerequisite part in studying networks as well as community detection. They are some very important statistical measures in networks such as: degree distribution, local clustering and node centrality. Beside, generative network models are also presented in the line with networks since these models reflect the way that networks are thought. Readers who are not familiar with the literature of complex network or network science are encouraged to have a look on this chapter. On the contrary, readers can skip this chapter and navigate directly to Chapter 3 dedicated to community detection.

- Since we are interested in comparing community detection methods, in Chapter 3, we introduce essential notions of community detection as well as some major challenges. Then, an introduction of different state-of-the-art and widely-used methods studied in this thesis will follow. These methods are classified according to different theoretical approaches in Section 3.2 in order to highlight theoretical distinctions between the associating mechanisms that community structures are explored. Readers who are familiar with community detection methods in the literature could proceed directly to a summary of these methods presented in Section 3.3 after Section 3.1.

- Chapter 4 presents our contributions on the evaluation and the characterization of meta-data community structures in many wide-known real-world networks. We introduce in this chapter a representation of communities using a descriptive approach that helps to unveil interesting structural information of community structures. By using this approach, we find that communities in real-world networks are usually not structurally good and are not always eligible to be employed as *"ground-truth"* in the problem of community detection. Then, we demonstrate a quantification showing that almost all popular detection methods can find significantly better clusters in function of many common

quality scores of meta-data communities or even planted communities in synthetic networks.

- If Chapter 4 focuses on structural information of meta-data communities, in Chapter 5, we investigate different kinds of topological communities detected by community detection methods. The studies in this chapter is based on the premise that community structures in different network categories can be distinguishable. By consequence, we employ some popular community detection methods as a tool to discover node interaction patterns in many types of networks. It turns out these patterns are very discernible from one network category to another and they could be matched to some popular network models in the literature. Analyses in this chapter help us to characterize community structure and connect well-studied network categories with well-known network models.

- In the previous chapter, we focused on modular structures inside networks and used community detection methods as a discovery tool to characterize them. In Chapter 6, we demonstrate several comparisons of community detection methods using different quality criteria. These comparisons expose various traits that help to profile community detection methods and provide several sources of information that could assist network practitioners to decide best methods for particular cases. We address several critical issues of the problem of community detection such as: practical time computation, number of communities, community size distribution.

- Finally, Chapter 7 is dedicated for discussion in regards to the study of community detection. Also, three-year is still a short period in order to discover comprehensively all of interesting aspects of this domain and we are definitely aware of some existing limitations. From that, the dissertation is closed by our perspectives of potential future work.

Chapter 2

Complex network and graph

Interest in using networks to resolve complex problems dates back to the 18th century with the *seven bridges of Königsberg puzzle*. In fact, back to 1736 Leonhard Euler resolved this puzzle by employing nodes and edges to illustrate connections and hence placed a foundation for graph theory. However, until the 20th century, due to the computer revolution and the rapid development of the Internet, huge amounts of data and computational resources have began to be available and join in the network playground which engender an emancipation of graphs from just a theoretical domain to pervade widely computer science. During the past decades, we have witnessed more than ever before, an evolution in the study of network science[1] in order to discover the structures of colossal networks. Since networks are considered as a natural language to represent interactions in complex systems, they are not restricted to only represent bridges and landmasses but have been expanding in many fields of study such as biology, social, information, communication, etc. in order to leverage powerful mathematical tools for discovering their mechanisms and functionalities.

Our study focuses on many facets of evaluating community detection techniques on networks, hence many compulsory and common state-of-the-art concepts will be repeatedly mentioned and employed ubiquitously in the later chapters. Consequently, every theoretical point one needs to know for a better comprehension of the following contents of this thesis are introduced in this chapter. We present in Section 2.1 a simplified global picture of complex system science perceived from a network science approach, this introduction allows to locate and formulate the problem of network analysis in general or specifically the problem of community detection from a more global perspective. From that, Section 2.2 establishes mathematical notations that help to demonstrate different definitions and measurements in networks such as essential statistical properties of graphs, generative network models. Community structure, which is the core of our study and some indispensable related keys are presented in Section 3.1, followed by community detection techniques in Section 3.2.

2.1 Complex systems

Although the problem of community detection was born independently in different domains as fundamental objectives, recently in a more global picture of science revolution, it has been studied widely in the complexity science community due to

[1]The distinction between *network* and *graph* is quite subtle in the literature. In this document, they are used interchangeably in many cases according to commonly usages in the literature. Note that some authors prefer to designate graphs to abstract mathematical objects and networks to actual systems. In this way, we say *network science* but not *graph science*, *graph theory* but not *network theory* and *hypergraph* but not *hypernetwork*.

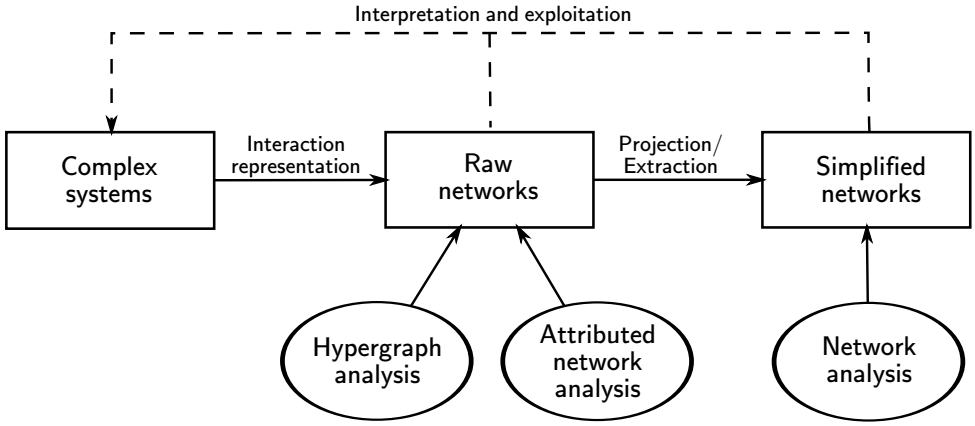


FIGURE 2.1: A summarized learning cycle using network analysis tools to extract and comprehend complex system information. The process is illustrated in a network science viewpoint.

the development of computer calculation capacity. It is hence important to contextualize the study of network analysis techniques a systematic relation with complex systems in order to avoid any further ambivalence. Emerged as a interdisciplinary domain, there is generally no officially accepted definitions in the world of complexity and most problems must be resolved in a domain-specific context. Since the generalization of complex science is far from the context of this thesis as long as cutting edge achievements of the domain, the utilization of *complexity-related* terms in this dissertation will be unambiguously abused in a thinking-network way. Figure 2.1 depicts a global picture that elucidates the intervention of network analysis tools in the process to study different collective behaviors of complex systems. It also helps to perceive whether a perspective could be theoretically attained or not using different network analysis techniques on the two stages shown in the figure. The following clarifications assist to decipher different blocks in the schema which serves as a context introduction.

Definition 2.1.1 *A complex system is a group composed of multiple entities which interact with each other. From these relationships, a collective behavior arises that can not be explained by the properties of the individual components.*[2]

It is also indicated that: "Complex is different from complicated in the sense that a system comprising a large number of entities is not necessarily complex". On the other hand, a simple and small system where interactions engender collective behaviors could be considered complex. Another definition of complex systems given by the Complex System Society:

Definition 2.1.2 *Complex systems are systems where the collective behavior of their parts entails emergence of properties that can hardly, if not at all, be inferred from properties of the parts.*[3]

[2]The definition given by Institut d'Études des Systèmes Complexes de Toulouse: https://xsys.fr/en/welcome/

[3]The definition given by the Complex System Society: https://cssociety.org/about-us/what-are-cs

The common point between the two definitions is the justification of the system's collective behavior or properties that can only be inferred or exploited from the interactions of its parts but not the individuals themselves. This makes the principle difference with the traditional reductionism whose approaches endeavor to explain phenomena in terms of their elementary constituents. A plethora of real world systems can be cited as complex systems such as predator-prey ecosystems, protein-protein interaction, human economics, telecommunication infrastructure, social and communication systems. A complex system can contain in itself different complex systems which could even be more complicated, or can be hosted in other systems.

Since the study of complex systems spans a wide range of disciplines, the methodologies employed and favorite tools in each field are also discernible from one to another. Network is probably the one of the most widely used tool to decrypt complex systems. As illustrated in Figure 2.1, information of interest in complex systems are extracted and presented in form of raw networks (Papadopoulos et al., 2012) which can be seen as their projections in a specific data structure. In raw networks, many contextual information about system's entities and their connection are available for sophisticated analysis tools. In real world as well as human-created complex systems, there are most of the time several types of entities, which raise an immense challenge for even the most modern and sophisticated tool for an exhaustive analysis. So that, according to a concrete purpose, suitable information need to be gathered from the studied system to allow ensuing processes to be realized faithfully. That is one of the reason why information extraction play a very fundamental role in the process of comprehending complex systems.

In the schema presented in Figure 2.1, from the left hand side to the right hand side, the amount of processable information decrease in exchange for a flexibility in using network analysis tools. In other words, simplified networks allow a wide spectrum of methods and techniques for system exploitation. On the other hand, in hypergraph and attributed network analysis tools, the constraints are much more severe and tools or analysis processes are usually designed for a specific context or domain due to the high complexity of objects of study. Particularly, in community discovering, the majority of methods are not applicable to hypergraphs or attributed graphs unless a preprocessing integration into structural information are had been done to obtain compatible processing data. For a clearer idea about the terminology used, we illustrate a specimen of a raw network using the context of Social Media depicted in Figure 2.2. The network contains three types of entities including user, published content and comment as well as many information related to these entities and their relations. It can be presented as a *tri-partite* graphs; hypergraphs where each hyperlink connects users, contents and comments; or attributed graphs where users are described by birth place, birthday, gender, career, etc.

The utilization of networks analysis methods to understand real world complex systems have been emerging with an enormous speed in several domains. Specifically, many efforts have been given to investigate the structures of communication, technological, information, biological, social systems (Newman, 2010). However, procedural tasks in the learning cycle could be different from one domain to another. Imagine the processes of collecting and integrating interactions of social systems could be totally different with those of biological systems as well as the set of available techniques do not allow researchers to realize the same action. In the scope of thesis, domain-related challenges are not focused although the great influences of their alternative solutions on the final result. Our study is based on a prime hypothesis that relevant information are satisfactorily synthesized for succeeding analysis and we are not judging domain-related techniques employed to collect data

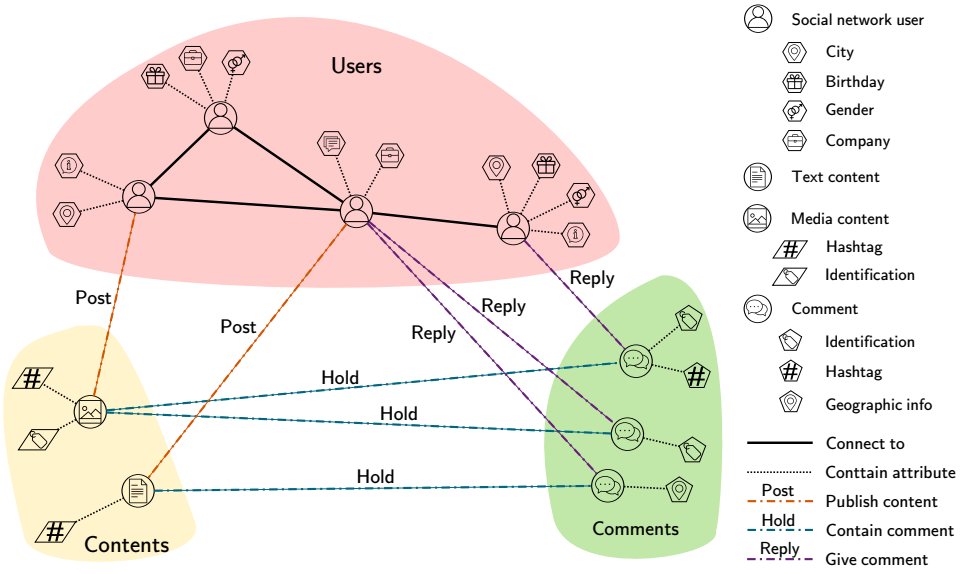


FIGURE 2.2: A visual representation of a fraction of a social network. Only a small set of exploitable information from most contemporary social media are illustrated.[4]

and their quality. In fact, all of real-world networks that are used in our analysis are well-known networks and have been widely used in many specialized researches. The quality of these networks is, on the other hand, evaluated using structural measures. The understanding of these underlying network structures is indispensable for the reasoning of appropriate metrics using in each analysis process. Essential structural measures will be presented in Section 2.2, but let us first introduce some highlight researches which have been conducted to understand the specificity of some principle network categories widely studied in the contemporary network science community.

Social and communication networks

Social networks comprise sets of social individuals and their mutual interactions, which are normally represented by networks in order to explain emerging social collective phenomena. In traditional studies, individuals are usually be people called *actors* and their social interactions, such as friendship, called *ties* in the lingo of socialists. Many other types of social interaction have been investigated such as networks of drug consumers or terrorists, movie actors, sexual contact networks, business relations between companies, etc. Although antecedent work of social analysis exists back toward the end of the nineteenth century, Jacob Levy Moreno, an American psychiatrist, is generally considered as the pioneer of the domain. With a presentation of relations by *sociograms*, he promulgated the first concepts of sociometry science to study the dynamics of social interactions in small groups of people (Moreno and Jennings, 1934). Another well discussed example of conventional social study is the case of the affiliation network called *Southern Women Study* of 18 women who participated to 14 different social events, presented by Davis *et al.* The authors inspected the social circles of these women by connecting those who attended at least to a common event and found out two subgroups of tightly connected clusters of

[4]Figure credit: some icons are from https://www.freepik.com/

acquaintances with weak inter-cluster interactions (Davis, Gardner, and Gardner, 1941). However, traditional studies in sociology are heavily contingent upon the methodology of collecting information which was mainly questionnaires, interviews or direct observations. Hence, they are subject to several restraints such as limited survey sample sizes, cognitive biases, etc. It was not until the 1960s, an innovative experiment of Stanley Milgram, a psychologist, was first introduced to study social networks differently (Stanley, 1967) in the well-known *small-world* experiment. The itineraries of 96 tracked packages which were intentionally sent from Omaha, Nebraska to Boston, Massachusetts through many intermediate recipients were analyzed. The author quantified the *geodesic distance* between actors in the social network, which characterizes the average number of maximal intermediate relations between any two arbitrary actors. The uncovering about this unexpectedly short distance has been inspiring many researches afterward and being subject to many further discussions.

Accompanying with the explosion of the Internet and online social network platforms, many innovative approaches leverage the availability of increasingly voluminous and often more reliable data sources. Although many privacy protection regulations are changing day by day research methodologies and accessible information, recent development of social media has unchained major constraints to the expansion of the domain. Nowadays, online databases contain social networks of people across many geographical areas, in a dynamical and evolving representation, with an massive amount of individuals which was, separately, already a big challenge. The example illustrated in Figure 2.2 reveals solely a small aspect of the complexity of contemporary social networks and the interactions of their constituents. Many prominent researches have been well leveraging the wave of technology to exploit collaboration networks between scientists (Newman, 2001b), dynamical aspect of high resolution student interaction networks (Sekara, Stopczynski, and Lehmann, 2016) and large-scale social networks (Leskovec, Kleinberg, and Faloutsos, 2007), social recommendations by detection of community of interest (Brun and Boyer, 2012), privacy-related questions of online social network profiles (Kevin, Jason, and Nicholas, 2008).

Communication networks consist of information exchanging systems between any kind of artefacts or humans such as signal transmission between computers or servers, email or message exchanges, phone calls, etc. They usually have a close relation with social and information aspects in many cases since people principally communicate within their social contacts. As a consequence, there is a similar evolution in communication patterns with that of social interactions along with the intervention of the Internet and collaborative platforms. In fact, the classification of networks into different categories is quite relative and fuzzy. Many networks can be classified into one category or another according to the aspect that we are interested in. Some popular studies focused on discovering communication networks of information exchange include: structure discovery in e-mail traffic networks (Eckmann, Moses, and Sergi, 2004), (McCallum, Wang, and Corrada-Emmanuel, 2007), large dynamic graph approach for fraud detection in telecommunication (Cortes, Pregibon, and Volinsky, 2001), strength ties in mobile communication networks (Onnela et al., 2007), human communication capacity and interaction strategies (Miritello et al., 2013).

Information networks

Information networks consist of interconnected systems of information contained items, which are mostly man-made. Some representative examples of information networks include the World Wide Web (WWW), networks of connected blogs or journals, the web of citations between scientific papers, peer to peer networks, etc. Information networks also have a close relation with social and communication networks. In fact, depending on the aspect that is considered, some networks could be classified at the same time to be in either of these categories.

The structure of the Web attracts enormous attentions in the scientific community over the last decades due to the availability of automatic computer programs such as web crawler. Specifically, the Web is widely considered as a network in which the vertices are web pages and the edges are the hyperlinks contained in these web pages which redirect users from the actual page to another. A web crawler receives an arbitrary initial page as source and scans all of its contents to find its connected pages represented in the form of *Uniform Resource Locator* - widely known as *URL* - to discover the Web throughout a breath first search process. However, given a web page, one is only able to discover a part of the Web since not every web pages are reachable from a single source. Moreover, many web pages are also technically invisible to some web crawlers. By consequence, the picture of the Web network is generally an assembly of many small fragments discovered independently from many parts of it. The study of the whole structure of the Web is challenging and time consuming due to its highly dynamical structure, a colossal size[5] and the unreachability of many web pages. Several notable discoveries to comprehend the structure of the Web could be found in the literature (Kleinberg et al., 1999), (Albert, Jeong, and Barabási, 1999), (Andrei et al., 2000), (Bosch, Bogers, and Kunder, 2016).

Other well-studied types of information network include the citation network between academic papers. In fact, each paper normally refers to many other published papers in its bibliography part to indicate the sources of reference. A network of citation constructed from these connections between papers could help to reveal interesting information about the picture of the research collaboration between authors or the relationship between different scientific disciplines as well as their evolution during time (Rosvall and Bergstrom, 2008), (Newman, 2001b). Besides, recommender networks of products on many commercial platforms have been also studied. Such that, along with the development of online shopping behavior, a wide variety of algorithms have been developed to attract consumers by recommending them relevant products that they are likely to buy. The relation between the efficiency of these collaborative filtering algorithms and the network structures began to draw the attention of some researches (Cano et al., 2006), (Su, Sharma, and Goel, 2016) similarly the effect of social connection recommendation on networks (Daly, Geyer, and Millen, 2010).

Technological networks

Technological networks include physical infrastructure networks constructed to facilitate or enable some specific demands. The Internet is probably the most well known and celebrated technological network, which links different information systems or devices together by electrical or optical cables, wireless for fast data connections. Besides the Internet, some other types of networks could be classified into

[5] It is estimated by `http://www.worldwidewebsize.com/` that there is around 47 billions reachable static pages on the WWW between *May 2016* to *May 2018*.

technological networks such as: transportation networks, distribution networks, telephone infrastructure networks (Newman, 2001b). Since technological networks are man-made systems and are regulated under different artificial mechanisms, their structural characteristics are very distinguishable from those of the other types of networks. Several researches have been conducted to understand complex properties of technological systems, notably the topology of the Internet in many structural levels including router-level, subnet-level, domain-level, autonomous system-level (Faloutsos, Faloutsos, and Faloutsos, 1999), (Pastor-Satorras and Vespignani, 2004), (Magoni and Pansiot, 2001). The telephone network topology is also studied using tomographic methods inspired by medical imaging (Rabbat et al., 2005), (Treichler et al., 2004). Besides, many statistical properties of technological networks whose topologies are highly impacted by geographical and demographic conditions such as electric power networks, airline traffic networks, road way networks, railway networks, etc. are well investigated (Amaral et al., 2000), (Porta, Crucitti, and Latora, 2006), (Sen et al., 2003), (Watts and Strogatz, 1998). In fact, the understanding of these networks could reveal valuable information for many real life issues such as vehicle traffic controlling, highway infrastructure development, efficient delivery and distribution problems and many other economical and management problems.

Biological networks

Networks are widely used in biology to represent interactions between biological elements as a very natural way. Biological networks cover a very wide scale: from macro networks such as interactions of different species in an ecosystem known as food webs to micro networks such as biochemical reactions between substances within cells. The mechanisms that many biological networks are determined are quite discernible from those of the other types of networks since they are usually dependent to available experimental techniques in biology. For instance, in a metabolic network where each node represents a chemical substance called *metabolite* and each edge represents a reaction, its is quite complicated in many cases to determine exactly the exhaustive participating components of every reactions. In protein-protein interaction networks or neural networks, the processes for determining whether an interaction exists between two proteins or two neurons respectively are usually time consuming and expensive. Hence, in many cases, analysis in biological networks are often conducted on small systems or just on a local fraction of the whole picture. There are still many challenges in constructing a full map of knowledge in many biological organisms.

The study of brain functionality using networks to represent neuron interactions by neuroscientists is probably the most common in this group. A full comprehension of human neural network requires an enormous number of experiments and many parts of the human brain stay mysterious up to the present time. In parallel, several researches have been focused on the structure of smaller neural networks such as the one of the nematode *C. Elegans* - a type of soil worm (Green et al., 2011), (Hizanidis et al., 2016) where the entire neural network has been successfully mapped (White et al., 1986). Besides, protein-protein interaction networks also received a great attention in biochemical biology. For example, some methodologies and experimental methods to translate protein interaction data into network presentations for understanding cellular processes are introduced in (Nils, 2014); the robustness of the *Saccharomyces cerevisiae* yeast's proteome against removal of proteins with different levels of centrality within its protein network in (Jeong et al., 2001); the dynamics of protein complexes that reveals previously unknown modules (Lichtenberg et al.,

2005), etc. At a smaller scale, structural properties of some metabolic reaction networks have been also studied to understand key aspects of cellular functionality and robustness. For instance, the metabolite hierarchical modularity organization as well as cellular functionality and gene regulation in *E. coli* intestinal bacterium are inspected using metabolite pathway structures (Stelling et al., 2002), (Ravasz et al., 2002), (Wunderlich and Mirny, 2006). Although an exhaustive list of current research trends in biology using network approaches requires undoubtedly a larger scale of investigation and domain knowledge, a few examples should suffice to expose a remarkable presence of network science in this area.

2.2 Preliminary definitions

In this section, some preliminary theoretical tools and concepts for the analysis and description of networks are introduced. Since networks are naturally represented by graphs, most analysis of real world networks in the literature leverage several methods and algorithms developed in graph theory to explain their structures, functionalities and relating phenomena. By consequence, essential notions of graph theory that help to understand network characteristics, analysis processes especially community detection are presented in this part. These concepts help to understand what do networks look like on a global scale, how do they evolve over time, whether they are robust against external stimulations and how do they change under perturbations, etc.

Several extensive researches using statistical tools have discovered many fascinating characteristics of real world networks such as *small-world* phenomenon (Watts and Strogatz, 1998) as also known alternatively as six degree of separation in an earlier version (Stanley, 1967), the power-law degree distributions, heterogeneous structure (Estrada, 2010), modular structure (Newman, 2006), self similarity (Chaoming, Shlomo, and Hernán A., 2005), etc . Not only in static networks, statistical properties also help to disclose many interesting properties in dynamic networks such as shrinking diameters, densification power-law, phase transition, etc.

Then, some traditional data clustering approaches will also be mentioned subsequently as they are involved in the partition process of some community detection algorithms. In fact, one popular approach in community detection consists in transforming network data into adapted forms represented by pairwise proximity distance between individuals and then performing conventional clustering techniques. It is hence requisite to expound the mechanisms of data clustering, at least in a perspective from where they will be adapted in community detection context. However, it is not supposed to be a comprehensive introduction for neither of the above contents. More complete details could be found in (Newman, 2010), (Estrada, Ernesto, 2011) for introduction of networks, in (Cormen et al., 2009) for graph theory and in (Hastie, Tibshirani, and Friedman, 2009a) for clustering techniques.

2.2.1 Graph

A network composing of individuals and their interactions can be represented by a *graph* - a type of data structure that allows several techniques to discover the network under question. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set $\mathcal{V}$ of nodes (or vertices - vertex in singular) representing individuals of the associated network and a set $\mathcal{E}$ of edges (or links) representing interactions between pairs of individuals. The number of nodes and edges in graph $\mathcal{G}$ are denoted by $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$ respectively. When

two nodes i and $j \in \mathcal{E}$ of a graph are connected by an edge $e_{ij} = (i,j) \in \mathcal{E}$, we can refer to them as *neighboring nodes* or sometimes *incident nodes* of edge e_{ij}.

The edges of a graph can be optionally described by a weight function $w(i,j) : \mathcal{E} \rightarrow \mathbb{R}^+ : i,j \in \mathcal{V}$ which allows to quantify the interactions between its nodes. In case where $\exists i,j \in \mathcal{V} : w(i,j) \notin \{0,1\}$, we call the graph *weighted graph* and denote it by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. The weight of an edge $(i,j) \in \mathcal{E}$ between two nodes i and j can be alternatively denoted as $w_{ij} : w(i,j) > 0$, which implies that a null weight indicates a nonexistent edge $w_{ij} = 0 \Leftrightarrow (i,j) \notin \mathcal{E}$. In the case that only the existence of edges in considered, we call the graph is *unweighted graph* or *binary graph*, which literally means $w(i,j) = 0$ if $(i,j) \notin \mathcal{E}$ and $w(i,j) = 1$ if $(i,j) \in \mathcal{E}$. We simply omit the weight function and use $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to denote unweighted graphs. Besides, when no further information is indicated, a graph is considered *undirected*, which means its edges are symmetrical and there is no specific order in the connection between two nodes: that is $(i,j) = (j,i)$. On the opposite, the graph is called *directed* if $(i,j) \neq (j,i)$. In this case, $(i,j) \in \mathcal{E}$ indicates an edge whose direction is from node i called *source* to node j called *target*. In a directed graph, edges are visually depicted by arrows from source nodes to target nodes while edges in a undirected graph are simply represented by links connects their extremities. Normally, without further indication, networks that we analyze in this work are represented by undirected and unweighted graphs.

Most of the time, graphs have at most one single edge between any pair of vertices. In some cases, it is also possible that there are more than one edge between the same pair of vertices called *multi-edges* and/or there are edges that connect vertices to themselves called *self-edges* or *self-loops*. Graphs that contain multi-edges are called *multi-graphs* and possibly have also self-edges. When a graph have no multi-edge nor self-edge, we call it a *simple graph*. Depending the context or analysis purpose, a multi-graph could be simplified by a simple graph by removing self-loops and presenting multi-edges by a weight function. Most of community detection methods in the literature are designed to work with simple graphs. Figure 2.3 illustrates an example of a simple network which is modeled as a simple graph 2.3(a) or a multi-graph 2.3(b).

A very important property of nodes in graphs, which is repeatedly discussed in many network analysis contexts, is node *degree*. The degree $d(i)$ of a node $i \in \mathcal{V}$ in graph $\mathcal{G}$ is defined as the number of connections that it has in the graph. In other words, it is the number of neighbors that node i possesses in graph $\mathcal{G}$. The degree distribution of nodes in a graph is a principle property that is usually studied to understand its global structure. Besides, for weighted graphs, we also define a *node weight* $w(i)$ of node $i \in \mathcal{V}$ to be the sum of the weights of its incident edges. Since $\forall i,j \in \mathcal{V} : (i,j) \notin \mathcal{E}, w_{ij} = 0$, we can write for all node i in $\mathcal{V}$:

$$w(i) = \sum_{j \in \mathcal{V}} w_{ij} \tag{2.1}$$

Graph representation

There are many ways to represent a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ mathematically such as: a collection of adjacency lists, an edge list or an adjacency matrix. Each type of presentation has its own advantages according to the task that one needs to conduct on the graph. While the adjacency list representation provides an efficient way to represent sparse graphs; the edge list representation is appropriate for processing

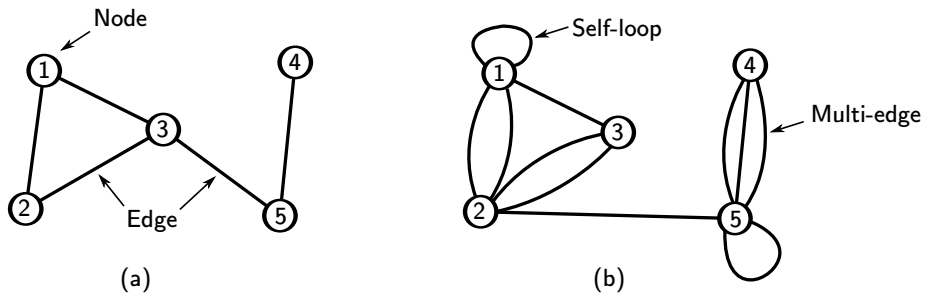


FIGURE 2.3: An example of a simple network represented by: (a) A simple graph, (b) A multi-graph with multi-edges and self-loops.

dynamic graphs whose edges are added or removed regularly; and the adjacency matrix representation facilitates algebraic calculations on graphs.

If we consider an undirected graph with n vertices which are indexed each one by a unique label, for example, from $1..n$ as the graph illustrated in Figure 2.3(a). We can present this graph of 5 vertices and 5 edges by an **adjacency list**: $\{(1:2,3),(2:1,3),(3:1,2,5),(4:5),(5:3,4)\}$ where $(k:i,j)$ indicates that node i and j are adjacent to node k in graph $\mathcal{G}$. The adjacency list representation, as indicated by its name, itemizes every neighbors of each node in the graph one by one. It works like a list of pointers in computer programming that contains addresses to all neighboring nodes. Hence it facilitates navigating procedures and is a preferred representation in the implementation of many algorithms in graph theory such as Prim algorithm for searching minimum spanning tree (Prim, 1957) or Dijkstra algorithm for searching the shortest path problem (Dijkstra, 1959), etc. The same graph in Figure 2.3(a) could be also represented by an **edge list** $\{(i,j)\}$: $\{(1,2),(1,3),(2,3),(3,5),(4,5)\}$. Edge lists and adjacency lists are often used to store networks in computers since they require much less memory than adjacency matrices, especially in sparse graphs where the number of edges m is much less than the number of possible edges n^2. Besides, edge weights could also be stored easily in by adding $w(k,i), w(k,j)$ in the adjacency list of k or simply a third element $w(i,j)$ in the edge list representation. However, for mathematical manipulation purposes, it is advantageous to use the **adjacency matrix**, where nodes are presented in rows and columns, and the elements of the matrix represent the weights of associated edges:

$$\mathrm{A}_{ij} = \begin{cases} w_{ij} & \text{if (i,j)} \in \mathcal{E} \\ 0 & \text{if (i,j)} \notin \mathcal{E} \end{cases}$$

When $\mathcal{G}$ is an unweighted graph, the w_{ij} elements in the adjacency matrix is then replaced by 1. For instance, the adjacency matrix A of the graph $\mathcal{G}$ presented in Figure 2.3(a) is:

$$\mathrm{A} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Simple unweighted and undirected graphs corresponds to symmetrical binary adjacency matrices whose diagonals elements are all zero. When a network is directed, its associated graph is not anymore symmetrical since $a_{ij} \neq a_{ji}$. Multi-graph adjacency matrices are, on the other hand, filled by integer values corresponding to

number of edges between pairs of nodes and non-null diagonal values corresponding to self-loops. Finally, in weighted graphs, these values are non negative real numbers which quantify interactions between nodes reflected by weight functions.

Graph connectivity

In order to understand about graph connectivity, it is necessary to understand the notion of *walk* and *path* in graphs. In simple words, a walk can be understood as a way of getting from one node to another node in a graph. It consists of a finite sequence of edges beginning at one node and finishing at the other, in which two consecutive edges are always adjacent or identical. A path is a special case of walks, where no node appears more than once in the edges sequence, which means edges must not be identical. For example, in Figure 2.3(a) $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 4$ is a path from node 1 to node 4 throughout a sequence of edges $(1,2) \rightarrow (2,3) \rightarrow (3,5) \rightarrow (5,4)$. There may be several paths between two nodes in a graph and the length of shortest paths among them are called *geodesic distance* or just *distance* for short. The average geodesic distance between two generic nodes in a graph is sometimes called *characteristic path length* L (Watts and Strogatz, 1998) to describe a dimension of graphs. Note d_{ij} be the distance between i and j, it can be written:

$$L = \frac{1}{n(n-1)} \sum_{i,j,i \neq j} d_{ij} \tag{2.2}$$

The notion of geodesic distance between nodes in networks is sometimes used as a function of proximity (or similarity) to determine input data for traditional clustering methods. However, this utilization provokes essential inappropriatenesses for network clustering problems. As real world networks are commonly sparse and have small diameters, nodes which are geodetically close sometimes should be considered to belong to different clusters. Inversely, geodetically distant nodes could, in some cases, affiliate to the same cluster. More advanced methods which have been developed to determine node closeness for network clustering will be discussed in Section 3.1.

An important notion to be mentioned in the analysis of networks is *component distribution*. It is an important feature to consider in order to get some insight into the network's global structure. In undirected graphs, we refer to a *connected component* as a subgraph where exists at least a path between any two arbitrary nodes. It means that, in a connected component, every node is reachable from any other node through at least one path. The component distribution reveals informative details about network connectivity, scarcity, vulnerability, etc. against some certain internal or external disturbance. Nevertheless, on the perspective of analyzing community structure in networks, without losing the generality, only connected networks are examined. Actually, in networks consisting of several connected components, the node clustering problem always has an evident solution in which each connected component is considered as a separated community. This seems to be a reasonable solution since it is somewhat arbitrary to group nodes from disconnected parts of a network into the same community. Hence, the community detection problem in a disconnected network can be effectuated independently in each connected component[6]. Hence, a pre-calculation of connected components is often required to make the network compatible with detection algorithms. Finding all connected components only

[6] Actually, in many community detection algorithms, it is implied that the input network must only have one connected component, otherwise a converged solution is not obtainable.

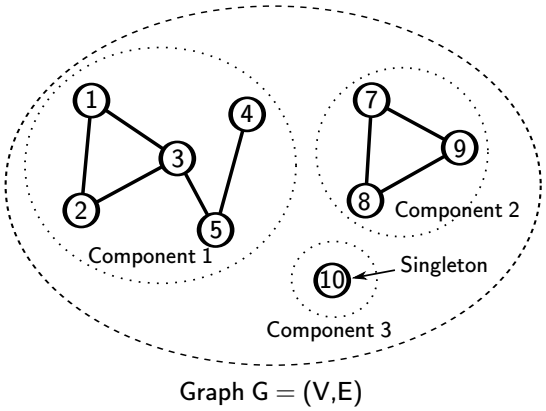


FIGURE 2.4: An example of a graph consisting of 3 connected components

requires linear time in terms of the graph size by using a breath first search or a depth first search $\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|)$. Besides, *singletons* representing nodes without any connection in their networks are also ignored in the community detection problem due to its triviality. Figure 2.4 illustrates the concepts of connected components and singletons in graphs. From later on, without any further mention, networks under our analysis include one and only one connected component, and every node has at least one edge connecting it to the other nodes in its network. These constraints also ensure important preliminary requirements for the correct functionality of some community detection algorithms.

Random walks

We introduce in this part an important stochastic process in graph theory called *random walk process*, which is a diffusion process that helps to understand the concepts of some community detection algorithms that make use of it. Basically, a random walk is a random sequence of nodes selected by a *random walker* based on a specific probability function similarly to *Markov chains* on a directed graph. That is, given a graph and a departing node, the walker choose stochastically a neighbor of its current node and move to this node. Then, the process continues through several time steps and creates a sequence of nodes called *random walk*. The probability that the walker chooses a neighbor for its next step does not depend on the past steps but only on its current position on the graph. This property is referred as the memorilessness of the process.

We define the transition probability of a random walker going from node i to node j in an undirected and unweighted graph, based on the above notations:

$$p_{ij} = \frac{w_{ij}}{w_i} = \frac{w_{ij}}{\sum_{k \in \mathcal{V}} w_{ik}}. \tag{2.3}$$

The possibility of going from a node i to another node is the portion of its weight to that node. For a connected graph with no singleton, $d_i > 0 : \forall i \in \mathcal{V}$, p_{ij} is finite and receives a value $0 \leq p_{ij} \leq 1$. The *transition matrix* (or stochastic matrix) $\mathbf{P}$ that characterizes the random walk process is then written:

$$\mathrm{P}_{ij} = \begin{cases} p_{ij} & \text{if (i,j)} \in \mathcal{E}, \\ 0 & \text{if (i,j)} \notin \mathcal{E}. \end{cases} \tag{2.4}$$

We can write the transition matrix $\mathbf{P}$ in function of adjacency matrix $\mathbf{A}$ and diagonal matrix $\mathbf{D}$ where $D_{ii} = w_i = d_i$ as following:

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}. \tag{2.5}$$

An important property of a random walk process is the probability of the position of the walker after a finite discrete time steps $t = 0, 1, 2, ...$ Given $\rho_i(t)$ representing the probability of the random walker being at node i at time step t, we have $\forall i \in \mathcal{V}, \rho_i(t) \geq 0$ and $\sum_{i\in\mathcal{V}} \rho_i(t) = 1$. This can be interpreted that every node in the graph could be reached if t is sufficiently large and at a certain time step, the walker must be somewhere in the graph. The probability of position of the walker at node j and time $t+1$ depends on the probability of position at time t and the probability of transition to node j as following:

$$\forall j \in \mathcal{V}, \rho_j(t+1) = \sum_{i\in\mathcal{V}} \rho_i(t) p_{ij}. \tag{2.6}$$

It can be written in a matrix form that $\rho(t+1) = \mathbf{P}^T\rho(t)$ where $\mathbf{P}^T$ is the transpose of transition matrix $\mathbf{P}$. A simple recursive construction gives us:

$$\rho(t) = (P^T)^t\rho(0), \tag{2.7}$$

where $\rho(0)$ represents the initial distribution of the random walker's position. The element p_{ij}^t of matrix $(\mathbf{P}^T)^t$ quantifies the probability that, starting at a node i, a random walker reach j in t steps. This probability of transition is *time reversible*, meaning that a random walk reserves its stochastic properties in two direction towards from i to j and backwards from j to i. Mathematically, it can be written that:

$$\forall i, j \in \mathcal{V}, \frac{P_{ij}^t}{w(j)} = \frac{P_{ji}^t}{w(i)}. \tag{2.8}$$

The probability of a random walker to go between two nodes depends only on their weights in weighted graphs or their degrees in unweighted graphs. This can be equivalently written in a matrix form:

$$\mathbf{P}^t\mathbf{D}^{-1} = \mathbf{D}^{-1}(\mathbf{P}^t)^T. \tag{2.9}$$

The vector of probability distribution of position of the random walker, denoted as $\rho(t)$ is proven to be *stationary* (or *steady-state*) when $t \to \infty$ in connected and aperiodic graphs, independently of the initial probability distribution of position. In this stationary state, the probability being found at a node is proportionate to the weight ratio between the node to the whole graph (László, 1993), (Pons and Latapy, 2005):

$$\forall i \in \mathcal{V}, \lim_{t\to\infty}\rho_i(t) = \frac{w(i)}{\sum_{k\in\mathcal{V}} w_{ik}} = \pi(i) \tag{2.10}$$

Many approaches of community detection have been inspired from the random walk process's properties. Although based on different ways of formulation, many of them search community structures by exploiting the properties of random walks such as time reversible, stationary state or regularity of walking patterns in different conceptualizations. Pons *et al.* define a similarity function between nodes in networks by using small steps random walks (Pons and Latapy, 2005). Then they use a

traditional hierarchical clustering method to effectuate clustering on their networks (Joe H. Ward, 1963) to minimize the average distance between nodes and their communities. Another popular community detection method exploiting random walks in an information theoretical approach that has been proposed by Rosvall *et al.* The authors describe nodes by encoded binary digits of different lengths in a way to maximize the compression rate of random walks which are represented by sequences of adjacency nodes (Rosvall and Bergstrom, 2008), (Rosvall, Axelsson, and Bergstrom, 2009). An efficient compression will encode nodes in a dense sub-graph by the same binary header, hence discloses potential candidates for communities. Many other approaches that exploit that random walk processes to detect dense structures in networks could be found in (Dongen, 2000), (Francois et al., 2004), (Haijun and Reinhard, 2004). However, these methods struggle with large scale networks due to high complexity of calculation time (in the order of n^3 where n is the number of nodes). The high complexity make them become less popular for many real world applications. For instant, an algorithm whose time complexity is in an order of $\mathcal{O}(n^2)$ takes hours for a fast personal computer to complete the calculation for a network of a million nodes, which is not quite affordable for many real-time applications; and the ones in an order of $\mathcal{O}(n^3)$ requires several years to solve large-scale networks of a million of nodes, which is quite an unreasonable time for most of the cases.

2.2.2 Statistical measures

Some concepts and statistical measures are indispensable to understand the functionality and the mechanism of different learning methods on networks. They consist in basis elements of a whole structure which are omnipresent in network analysis processes. Some essential concepts, which directly concern important community detection techniques in the literature are introduced below.

Degree distribution

As presented in the previous section, the degree d_i (sometimes denoted as k_i) of node i in graph $\mathcal{G}$ signifies the number of edges that it shares with other nodes in $\mathcal{G}$. It can be interpreted as the number of connections or the number of neighbors of a node in its network. Mathematically, the degree of a node in an undirected graph equals the number of non null values of the corresponding row (or column) in the adjacency matrix. This calculation could be simplified to be the sum of row i or column i in the adjacency matrix of the unweighted graph.

$$d_i = \sum_{j=1}^{n} A_{ij} = \sum_{j=1}^{n} A_{ji}. \tag{2.11}$$

In directed networks, one could distinguish incoming degree d_i^{in} and outcoming degree d_i^{out} of a node representing the number of links pointing to node i and the number of links leaving from node i respectively. In fact, the way that nodes connect to each other differs from one network to another and it is important to understand connection property of networks. Normally, we characterize the connectivity of a network by its degree distribution $p(d_i = k)$ representing the probability of node i having k neighbors in the network. The degree distribution reveals whether nodes connect in a homogeneous or heterogeneous way and quantitatively how edges are expanded over a network.

One the most early pioneer theoretical study in the contemporary network science about the degree distribution of random networks could probably be attributed to (Erdős and Rényi, 1959) and (Gilbert, 1959). More recently, Barabási *et al.* introduced an appealing empirical analysis of many real-world networks and disclosed a frequently observed property of *scale-free* degree distribution according to which the authors consider to be a consequence of self-organizing phenomena in the development of networks (Barabási and Albert, 1999). The finding eventually attracts many attentions in the research community. A detail technical discussion about popular probability functions, including *power-laws*, *Pareto distributions* (Pareto, 1964), *Zipf's law* (Zipf, 1949) which are frequently found in degree distributions of many real world networks, is reviewed meticulously in (Newman, 2005) and (Clauset, Rohilla Shalizi, and Newman, 2009).

In the network science community, many efforts have been given in the recent years to model network degree distribution in order to characterize the nature of real world systems and to describe mechanisms that are responsible for their formation. While bell-shaped distribution such as Gaussian gained a great success to describe many random processes in nature, they are not very compatible to explain phenomena in network connectivity patterns. Imagine the number of links point to a web site, the number of citations to a scientific paper, the number of interactions of a protein or a gene, etc. they are not distributed massively around their average values in general such as the distribution of noise in signals or height of humans. The common point that characterizes their connectivity is the profusion of connections in some few individuals in the expense of a small number of connections for the majority of individuals (Faloutsos, Faloutsos, and Faloutsos, 1999), (Barabási and Albert, 1999). In many real world networks, if we plot the degree distribution by a histogram, it will be highly *right-skewed*, which means the bulk of the distribution is found mainly in the small values range and there are only a small number of high degree nodes exhibiting in a *long tail* on the right of the histogram. It is to say, there is a high dynamic in the probability variation from small degrees to large degrees. Interestingly, if we illustrate the histogram in a logarithm scale in the horizontal and vertical axes, the distribution becomes quite a straight line. This means the distribution p_k that node i has k neighbors can be estimated to follow a linear relation in a logarithm scale, which allows us to write:

$$log(p_k) = -\alpha log(k) + c, \tag{2.12}$$

where α and c are constants. If we take an exponential of both sides, this relation can be written equivalently:

$$p_k = Ck^{-\alpha}, \tag{2.13}$$

where $C = e^c$ such that, $\sum_{k=1}^{\infty} p_k = 1$. The degree sequence in this case is said to follow a *power-law* distribution since the probability of a node having k neighbors degrades polynomially in function of k. The α constant is often call power-law exponent or power law coefficient, which is estimated to vary normally between $2 \leq \alpha \leq 3$ in many real world networks but sometimes networks with $\alpha > 1$ or $\alpha < 4$ are found (Dorogovtsev and Mendes, 2002), (Newman, 2003). Some authors use the term *heavy-tailed* or *fat-tailed* distributions to generalize these particular degree distribution decaying polynomially instead of exponentially as the degree $k \to \infty$ and hence having unbounded variances (Mary, Leman, and Christos, 2011).

Figure 2.5 illustrates this idea of degree distribution which characterizes a large number of real world networks. Concretely, Figure 2.5(a) depicts the degree probability distribution in Amazon network containing products that are frequently bought

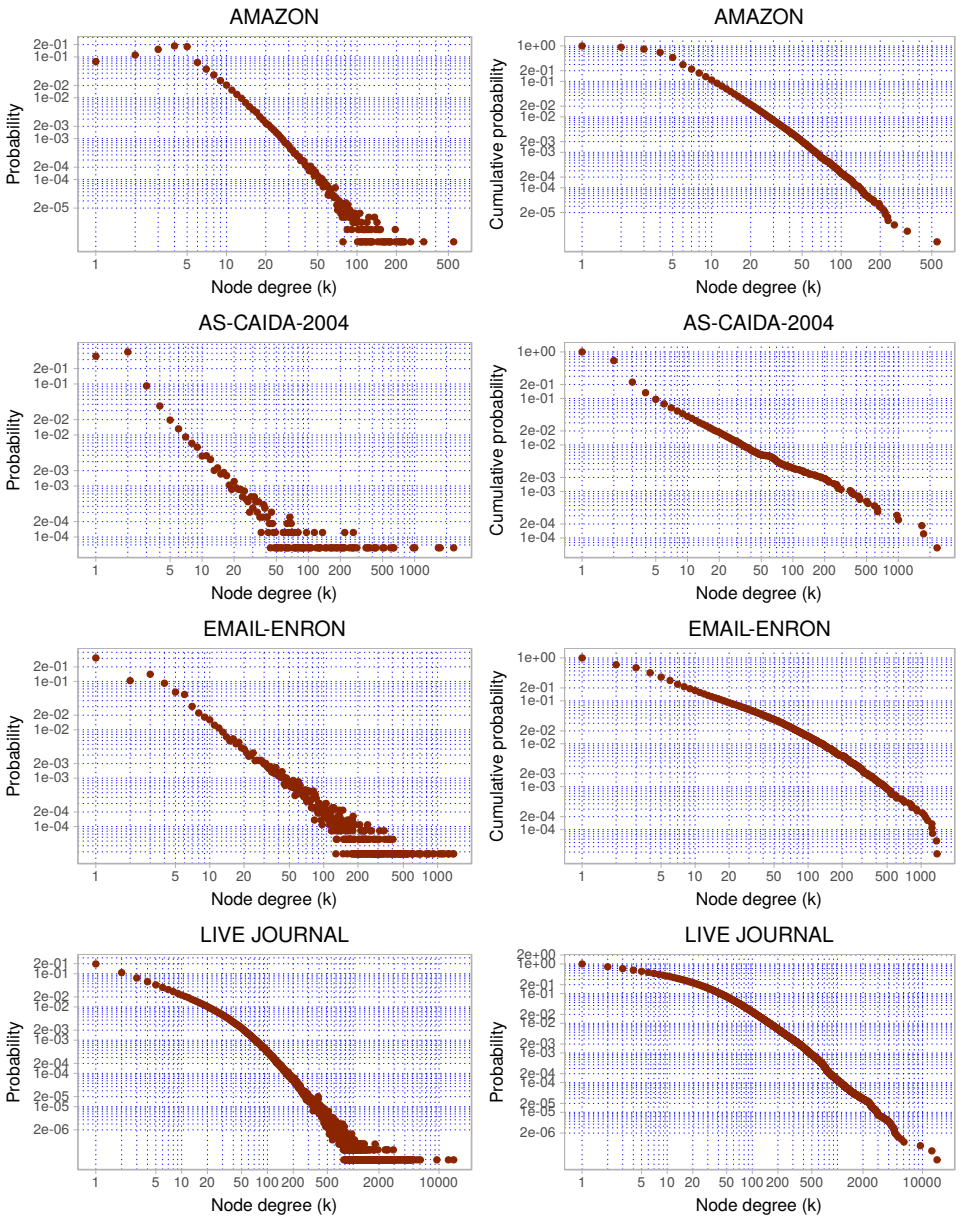


FIGURE 2.5: Degree distribution (left column) and degree cumulative distribution (right column) of some popular networks. From top to bottom: (a,b) Network of products that have been bought on a same cart in Amazon online commercial platform, (c,d) The network of connection between CAIDA Autonomous Systems recored in 2004, (e,f) Exchanges of around half million emails between different users in Enron company, (e,f) Live journal online friendship blogging community where users form groups and connect to each other.

together on Amazon[7] commercial website. The distribution shows are straight line in a logarithm scale with several dwindled samples on the right-hand side corresponding to high degree nodes (all distribution functions on the left column). This noisy phenomenon is due to the small number of large-degree nodes which creates a fluctuation in the distribution on the tail. The estimation of power-law fitting could be erroneous, especially on small networks. Using *logarithmic binning* technique with incremental intervals could palliate this problem and reduce statistical errors, however unlikely to be used due to the context dependent that requires further inspections for adapted intervals. Instead, the *cumulative distribution function* is employed to delineate the degree sequence. Such that, the probability P_k that a node has k or more connections can be calculate:

$$P_k = \sum_{k}^{\infty} p_k = \frac{C}{\alpha - 1} k^{-(\alpha-1)}, \tag{2.14}$$

where the exponent $\alpha > 1$. Hence, the cumulative distribution function P_k also follows a power law, but with a smaller exponent $\alpha - 1$. One could easily deduce the original coefficient by estimating the slope of P_k, which is 1 unit shallower. The cumulative distribution of the above mentioned networks are illustrated in the right column of Figure 2.5 with a much smoother quality. Nevertheless, one could undoubtedly recognize that the distributions do not follow the power-law on the whole ranges of degree. In many cases, they are just well fitted for high-degree enough nodes $k > k_{min}$ such as shown in Figure 2.5(h) of the *Live Journal* network where the law is only becoming well suited from $k_{min} \approx 100$. Sometimes, the distribution reassembles power law over a range of smaller degrees and decays faster for higher degrees as shown in Figure 2.5(f). This is considered as an exponential cutoff by some authors (González, Hidalgo, and Barabási, 2008), (Clauset, Rohilla Shalizi, and Newman, 2009) who model the distribution by adding an exponential term:

$$p_k = C e^{\frac{-k}{K}} k^{-\alpha}, \tag{2.15}$$

where $e^{\frac{-k}{K}}$ is the exponential cutoff. Studying the degree distribution of a network reveals a lot of insight about it. The power law distribution implies that there are a lot of *hubs* or *connectors* that attract a large fraction of nodes in their neighborhood whether the majority of nodes just have a few neighbors. This characteristic engenders many interesting properties that will be discussed more in the later sections.

Local clustering

Another important structural feature of networks that are widely studied, especially in the context of social networks, is *clustering coefficient*. It reflect the likelihood of occurring an edge between two incident nodes of any arbitrary node in a network. In a friendship network for example, a high clustering coefficient means people who have a mutual friend is more likely to be friends than two randomly chosen people. It is not clear when and where the concept was first used to study networks, neither why the clustering coefficient are named in such a way since it is not directly related to network clustering problem. However, according to (Newman, 2010), the term was probably first proposed by (Watts and Strogatz, 1998), (Watts, 1999) but similar concepts have been used before to analyze a node property in networks called *structural hole* (Burt, 1992) reflecting the opposite idea, which is the missing links between

[7] https://www.amazon.com/

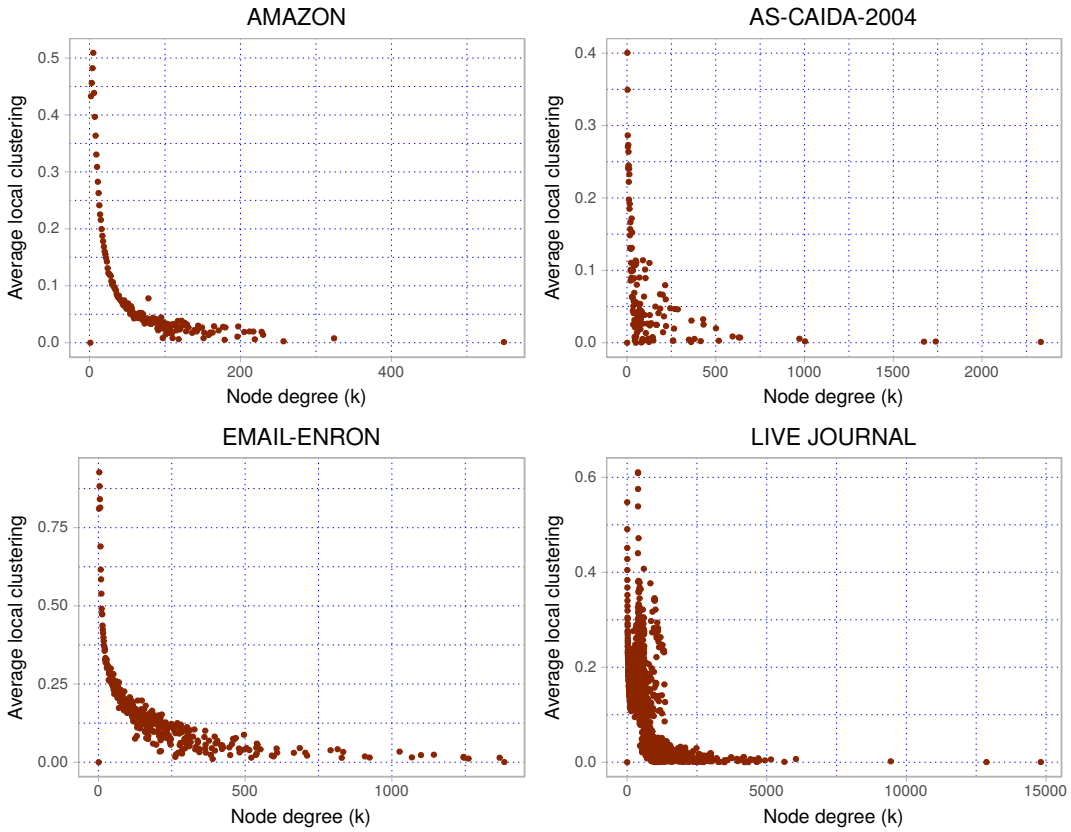


FIGURE 2.6: The average local clustering coefficient of the above presented networks. From top to bottom, left to right: (a) Network of products that have been bought on a same cart in Amazon online commercial platform, (b) The network of connection between CAIDA Autonomous Systems recored in 2004, (c) Exchanges of around half million emails between different users in Enron company, (d) Live journal online friendship blogging community where users form groups and connect to each other.

neighbors of a node. Since the first appearance, there are many derived formulas to demonstrate different versions of clustering coefficient (Barrat et al., 2004), (Marcus, 2008), which are sometimes very distinguishable. Nevertheless, the most well-known and used version of local clustering C_i of a node i is expressed as (Watts and Strogatz, 1998):

$$C_i = \frac{\text{Number of triangles connected to node } i}{\text{Number of pairs of neighbors of node } i} \tag{2.16}$$

It quantifies the fraction between the number of connected pairs of neighbors of node i and the total number of pairs, meaning the average possibility that two friends of i are also friends of one another. In case that a node have zero or only one neighbor, C_i can be defined as 0 or 1 according to the specific context and by definition $0 \leq C_i \leq 1$.

Figure 2.6 illustrate the average local clustering coefficient of nodes in the 4 previously mentioned networks in function of their node degrees in the networks. As we can easily observe, the general trend of local clustering is quite clear: the higher the degree of a node, the lower the possibility that its neighbors are connected. In fact, according to the context, different local values will be expected. For example, to evaluate whether a node is a high *connector* of *information flow* in a network, a

low local clustering coefficient would be expected since a low value means most of its neighbors must go through it in order to to propagate information between each other. On the other hand, in the context of traffic circulation where a node represents a city and an edge represents a road, a high local clustering coefficient of a city implies a good signal since there are several alternative itineraries between its surroundings. The local clustering concept shows a close relation with betweenness centrality (Burt, 1992), which will be introduced in the next section, but much more simpler to calculate. Therefore it is sometimes used to replace the betweenness measure to reduce time complexity, especially in large-scale networks. Some researches show that the local clustering of nodes in function of their degrees in some networks such as the Internet, words co-occurrence networks, urban streets system could be estimated to follow a scaling law distribution: $C(k) \approx k^{-0.75}$ (Vázquez, Pastor-Satorras, and Vespignani, 2002) or $C(k) \approx k^{-1}$ (Erzsébet and Albert-László, 2003) and between $C(k) \approx k^{-1.26}$ and $k^{-0.5}$ (Porta, Crucitti, and Latora, 2006).

In some cases, it is preferable to analyze local clustering on a network level. It can be calculated as following (Watts and Strogatz, 1998):

$$C(\mathcal{G}) = \frac{1}{n} \sum_{i \in \mathcal{G}} C_i \tag{2.17}$$

In many real-world networks, this coefficient is often found in the range between 0.1 and 0.9. However, as can be seen in the previous section about degree distribution, there are normally an enormous number of small degree nodes in real-world networks. Further more, it can been seen in Figure 2.6 that there are many differences between the local clustering coefficient of small degree and large degree nodes. The coefficient given by Equation (2.17) will be highly dominated by low connected nodes. Hence, another definition of global clustering (Barrat and Weigt, 2000), (Barrat et al., 2004) are sometimes preferable:

$$C_{BW}(\mathcal{G}) = \frac{\text{Number of closed paths of length two}}{\text{Number of paths of length two}} \tag{2.18}$$

This coefficient is usually used to evaluate characteristics of small-world networks and community structure in networks, which will be discussed in more details under the name **C**lustering **C**oe**f**ficient (*CCF*) in later sessions.

Node centrality

Social influence is becoming a very fashionable subject of discussion in recent years from static network to streaming network contexts (Matthew, 2008), (Flaviano et al., 2016), (Subbian, Aggarwal, and Srivastava, 2016) and are penetrating a multidisciplinary playground gathering economics, viral marketing, management, etc. The study of social structures, information diffusion, customer behaviors are closely related using network approaches under the assumption that people tend to follow the behaviors of their friends. Among available techniques, the analyzing of node centrality in networks contributes a principle role in these fields of study to evaluate the impact of each individual to different collective phenomena. Depending on the context, there are several variations of centrality metrics that can be used to reflect the desired concept. The most commonly discussed centrality metrics in the literature that could be cited consist in *degree centrality* or simply degree (presented in

Section 2.2.1) *closeness centrality* (Sabidussi, 1966), *Katz centrality* (Katz, 1953), *eigenvector centrality* (Newman, 2008) and *betweenness centrality* (Freeman, 1977; Newman and Girvan, 2004).

Specifically, the degree centrality of a vertex is simply the number of edges attached to it, it is often a highly effective indicator to evaluate the influence of a node in its network. For instance, a person who has many friends in social networks probably gains more social influence than a low connected person. Eigenvector centrality is, on the other hand, a little bit more sophisticated than degree centrality since it takes into account the associated centrality of neighboring nodes of the node under consideration. Taking the famous Pagerank algorithm (Page et al., 1998) of Google search engine as an example, its very first version considers a website popularity based on not only the number of its hyperlinks from other websites but also on the their qualities. Imagine the notoriety of a website would not increase in the same way if it receives a citation from a famous review and from a personal blog. Katz centrality derives degree centrality in a slightly different way, it weights the influences of other nodes to the centrality of the node of interest in function of their geodesic distances. A direct neighbor will contribute an amount of β to Katz centrality but a neighbor of a neighbor will contribute only β^2. In this way, longer walks will be heavily penalized in the calculation of Katz centrality.

The closeness centrality and betweenness centrality are both based on the concept of path in network. While closeness centrality reflects the notion of geodesic distance, i.e. shortest path between two vertices, the higher the closeness centrality score of a vertex, the lower the average geodesic distance from it to the other vertices of the graph. This notion is close to the characteristic path length of networks described by Equation (2.2) in the previous section, but in a node-scale instead of network-scale. Betweenness centrality of a vertex i measures the fraction of shortest paths between every pair of vertices in the network that traverse i. It represents somehow an influence in the sense of information flow between individuals based on the hypothesis that information flows along geodesic path and every vertex is equally seen as information source. Besides, local clustering presented in the previous section can also considered as a kind of node centrality in networks, which reflects the *transitive* central notion.

2.2.3 Generative network models

The previous sections introduced a summarized picture of some essential characteristics of real-world networks that have been discovered in the contemporary network science community. The understanding of these notions is the first step, if not the most important stages in the analysis of networks. It helps to construct appropriate exploratory processes, to choose most suitable metrics for concrete cases, to interpret obtained results and to get insight into hidden information in networks. Due to the explosion of technological advances in the age of information, a plethora of data are becoming available for researching different learning methods. However, real world networks contain uncontrolled properties for the evaluation of the performance of different methods, and hence restrict independent experiments. Many network models and benchmarks are invented under some specific hypotheses to circumvent this obstacle and allow analysts to ensure testing conditions. The most well-known models that are worth to be mentioned are *Erdős-Rényi (ER) model* to create random networks, *Watts-Strogatz (WS) model* to create small world networks

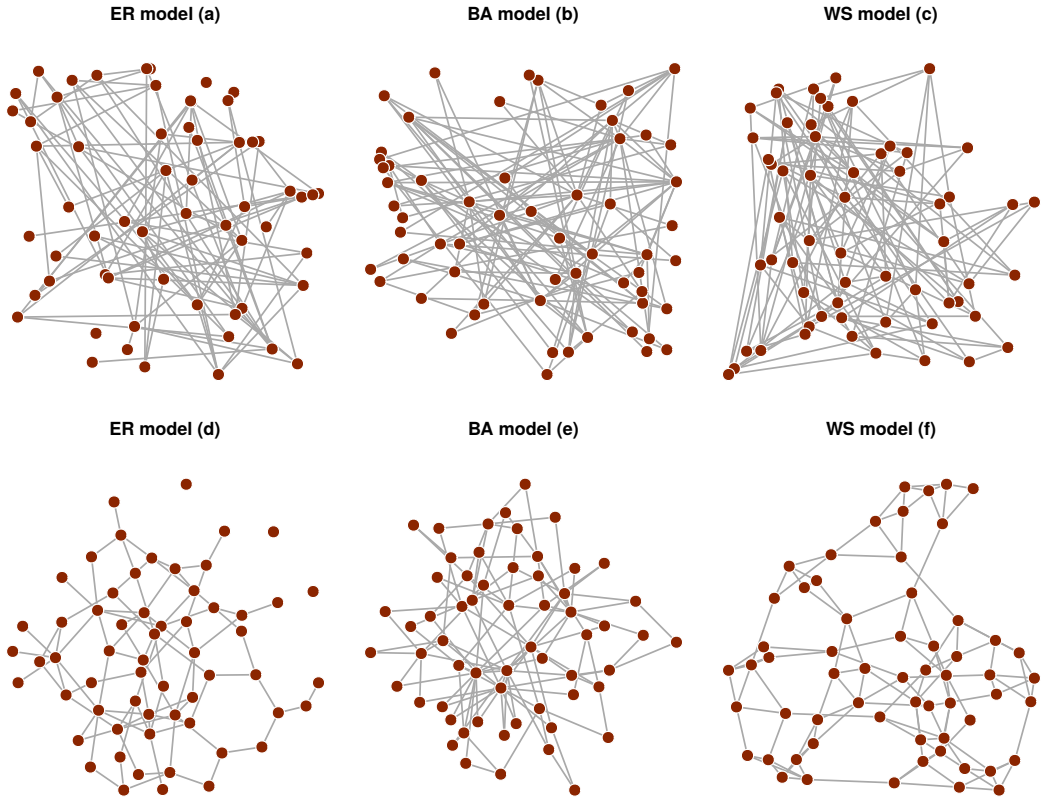


FIGURE 2.7: Some network instances created by different network models with $n = 60$ nodes each one. Networks are illustrated in a random layout in the first row and in Fruchterman-Reingold layout (Fruchterman and Reingold, 1991) in the second row. Figure (a) and (d) in the first column represent a random network created by Erdős-Rényi model with a probability $p = 0.05$, that two nodes are connected; (b) and (e) in the second column represents a *scale free* network created by Barabási-Albert model where each new node connect to two other nodes when it comes to the network, hence the average degree $\langle k \rangle = 4$; (c) and (f) in the third column represents a *small world* network created by Watts-Strogatz model where each node is connected to $\langle k \rangle = 4$ closest neighbors and the rewiring probability that a connection are moved randomly is $p = 0.1$

and *Barabási-Albert (BA) model* to create scale-free networks. These generative network models produce networks with some expected structural properties to understand the behaviors of real world networks and different analysis methods. In the context of community detection, sometimes one needs additional information about modular structure of networks under consideration as a reference to evaluate his or her method. In this occasion intervened the *Girvan-Newman (GN)* benchmark and the *Lancichinetti–Fortunato–Radicchi (LFR)* benchmark which produces networks accompanied with associated *ground truth* community structure to response to this demand.

Figure 2.7(a-c) illustrates some networks produced by these models in two different layouts. With similar network configurations (number of nodes, average degree), it is difficult to distinguish the differences of network's structure in a random layout, even when the degree distribution in these networks and the responsible

mechanisms for their creation are quite discernible. The same networks are illustrated in a different layout in Figure 2.7(d-f), it is easier to interpret the fundamental organizing principles of network connectivity behavior. In a random network, there is a regularity in the connectivity between nodes since edges are constructed randomly between each pair of nodes. In scale free networks, there are several hubs (node connectors) with high degrees that reach a large portion of nodes in their networks. In small world networks, nodes create compact and tight-knit connections in their local neighborhoods with a small fraction of remote connections that helps to reduce significantly network's diameter with comparison to a random network. More details about these generative network models will be presented in Section 5.1.5 in order to describe different topologies of community structure in networks.

It would require a much more detailed synthesis to address all important notions of complex network and graph theory relating to the problem of community detection. We presented in this chapter some of the most important and compulsory ones, which are well presented and utilized in community detection algorithms. In the following chapter, we are introducing more technical contents relating to community structures as well as well-known and state-of-the-art detection methods that are analyzed in this thesis.

Chapter 3

Community structure and detection methods

In this chapter, Section 3.1 is dedicated to a brief introduction of some essential notions of community detection. It is followed by Section 3.2 being a technical introduction of some highlight community detection methods that will be analyzed in the next chapters. Readers who are familiar with community detection methods in the literature can skip this section and go directly to a brief summary presented in Section 3.3.

3.1 Community structure and challenges

Due to different natural or artificial mechanisms that regulate the complex connectivity of nodes in networks, their organizations are generally not random nor regular but dissimulate highly inhomogeneity and some special patterns. These mechanisms provoke some typical properties of real world networks that are not exposed in random networks such as: power law degree distribution, small characteristic path length and recently widely studied *community structure*. In fact, in many networks, nodes are not connected to each other equally with an invariant probability, but they have a tendency to connect more frequently with some specific ones.

For example, in social networks peoples are often connected to their friends or acquaintances, their geographical neighbors at home or at work, more than an arbitrary people that they met. This preference connectivity phenomena give rise to the occurrence of groups of densely connected nodes in networks called *communities* (Wasserman, 1994), (Girvan and Newman, 2002). On the Internet, websites are more likely to refer each other in the same topic through hyper-links. For instance, a cooking blog might contain more connections to other cooking blogs, forums, magazines than to political on-line newspapers or scientific discovery channels. In protein-protein interaction (PPI) networks, proteins interact very frequently if they belong to the same functional blocks, i.e. proteins having similar biological functions, which are expected to be involved in the similar processes. The detection of these groups in PPI networks are important for the prediction of cancer and metastasis. However, such groups of content-similar blogs, functional biological molecules or real-life friends in social networks are not that explicit. However, if we are able to construct networks to describe complex systems in a way that there are much more connections between nodes insides *real modules*, then the identification of these modules could be solved through the detection of dense subgraphs.

3.1.1 Problem identification

In network science, *community detection*, sometimes called *graph clustering*[1] is one of fundamental challenges to discover the structure of networks in a mesoscopic level. However, it is an ill-defined problem such that there exists no universal definition or closed form formula of what kind of objects one should be looking for (Fortunato and Hric, 2016), and consequently there is ambiguity on what should be used as a golden standard to assess the quality of a community and the performance of a detection algorithm.

The most frequently found definition of community in network science literature is derived from the mechanism of connection preference. It implies that *a community is a group of nodes (a subgraph) in a graph where there must be many edges (denser) connecting them together than edges connecting the community with the rest of the graph* (Radicchi et al., 2004), (Fortunato, 2010). Newman defines a community as a *"group of vertices with a higher-than-average density of edges connecting them"* (Newman, 2006). Depending on the context, a community may be called a *cluster*, a *module*, a *class* or a *modular group*. This definition is the most basic that sets the fundamental requirement for most of its derivative definitions. Many different variations of community could be found in (Wasserman, 1994), for instance $LS-set$, which is a set of nodes in a network such that each of its proper subsets has more ties to its complement within the set than outside; or $k-core$, which is a subgraph in which each node is adjacent to at least a minimum number k of the other nodes in the subgraph. However, in recent developments of community detection algorithms, there is no consensus of the quantity of edges in reality that could be considered as *"many"*, communities are just algorithmically defined, i.e. they are final products of the algorithm without any precise a priori definition (Fortunato, 2010).

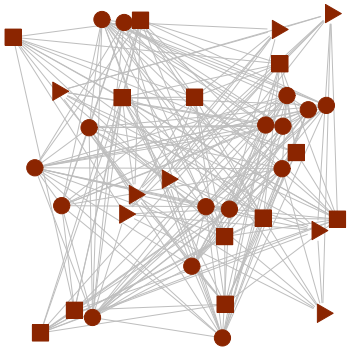
Community detection definition

Given a network that could be presented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a community detection algorithm task is to find a partition $\mathcal{P} = \{C_1, C_2, ..., C_p\}$ of nodes $\mathcal{V}$ in order to satisfy the basis condition of community structure stated above. This means that there must be much more edges inside communities than edges between communities. The quality of detected communities is often evaluated through a *quality function*[2] $\mathcal{Q}$, which quantifies the fitness of discovered groups in function of a specific aspect. Many quality functions exist in the literature, but there is still not a consensus on which is the best one. Nevertheless, the most commonly used quality function is the *modularity* (Newman and Girvan, 2004) whose formula will be first introduced in Section 3.2.2.

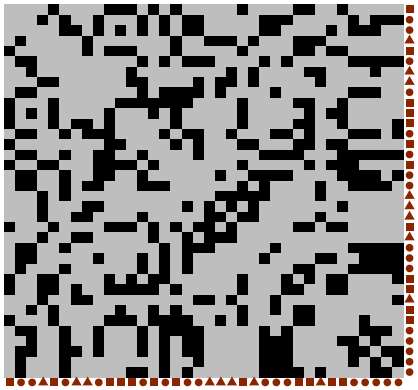
When $\forall i \neq j, C_i \cap C_j = \emptyset$, we say that the communities are disjointed, otherwise they are called overlapped. Normally, community detection methods attribute at least one community for each node of the network, which means $\bigcup_{i=1}^{p} C_i = \mathcal{V}$. There are recently some community discovering methods that try to reformulate the problem by identifying only local communities for a subset of nodes in networks or just

[1] The concept of graph clustering might refer to two different meanings existing in the literature. The first one implies a categorization of many graphs into different sets within which graphs share a common similar feature. The second one relates to the problem of partitioning nodes of a graph into densely connected groups. Here we means graph clustering in the latter case.

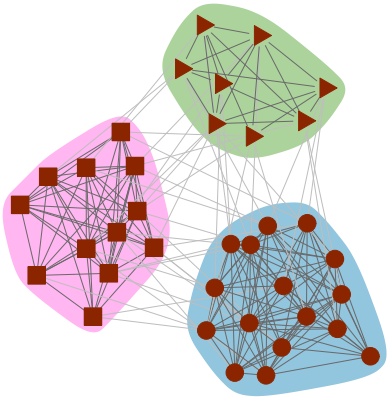
[2] Quality function can be sometimes called goodness function, objective function, fitness function or benefit function according to the context. While goodness/quality function are often used for the evaluation of detected communities, objective/fitness function are related to an estimation or an optimization process.



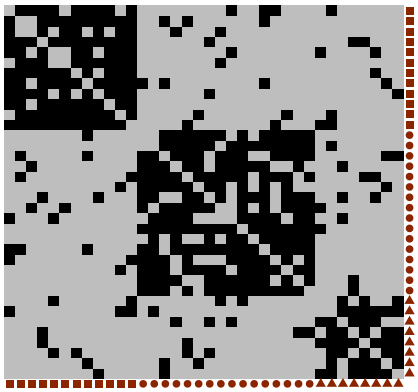
(A) Random layout



(B) Random order



(C) Force-directed layout (Fruchterman and Reingold, 1991)



(D) Ordered by group affiliation

FIGURE 3.1: (a) A network of 36 nodes, 217 edges with hidden densely connected nodes. Nodes with the same shape are supposed to be connected more frequently than two randomly chosen nodes. (b) The associated adjacency matrix of the network when its nodes are numerated randomly without knowing its clustering structure. Black pixels represent non-null values and gray pixels represent null values. (c) The network are plotted to clarify a clustering structure whose nodes are grouped in 3 clusters (communities) and are illustrated in dark-background colors, *intra-community* edges are drawn more bold than *inter-community* edges. (d) The reordered adjacency matrix of the network whose indexes are enumerated in a way that nodes having the same shape are placed next to each other.

significant communities that have good qualities. However, the methods employing this approach are not focused on this thesis.

In order to illustrate the task of community detection in networks, we create a network with a strong community structure embedded inside and later use an algorithm to find it. Figure 3.1(a) illustrates an artificial network whose nodes are visualized in a random layout as we do not take into consideration the community structure embedded. By this representation, we do not have a lot of information about how nodes interact. It is difficult to recognize that we actually configure on purpose to make nodes having the same shape (triangle, circle, square) more connected to each other than between nodes of different shapes. Figure 3.1(b) on the right-hand side represents the associated adjacency matrix of the network with a random numeration where black pixels indicate edges between associated rows and columns. Once again, the adjacency matrix is not directly interpretable and does not convey connection patterns of nodes in the network.

Figure 3.1(c) presents the same network as shown in Figure 3.1(a) but we applied an community detection algorithm and it turns out that there are probably three different densely connected groups which are highlighted in three different colored background. On the right-hand side, the columns and rows of the adjacency matrix are reshuffled in such an order that nodes belonging to the same group are placed next to each other. Since there are more edges inside each group called *internal edges* than edges that across between two groups called *external edges*, the adjacency matrix expose a diagonal block form. This block form shows the connection likelihood between nodes in the network and is sometimes chosen as visualization technique to demonstrate community structure. Finally, the role of a community detection algorithm can be globally resumed as to find an arrangement of nodes in a adjacency matrix to make emerge a diagonal block form.

However there are more constraints in reality, which are sometimes not explicitly expressed, than that appeared in the announcement of the problem. If one only look for a partition of graph that maximize the number of internal edges and minimize the number of external edges, then the graph itself can be considered as a big community and there is none external connection. Another solution is to let the node having the smallest degree into one community, and all other nodes into another community. This solution could also maximize the ratio between external and internal edges. However, these monotonous solutions seem not be a seductive one for most (if not to say all) analysts who consider using a method to detect communities. In fact, it is preferable to cluster a network into at least 2 relatively similar size communities or more[3] (Newman, 2010). It means that somehow, the relative size of communities with respect to the network is important without having explicitly been announced. Besides, there are many other criteria that could be mentioned such as community complete mutuality, reachability, vertex degree distribution and the comparison of internal versus external cohesion (Wasserman, 1994), (Fortunato, 2010). There exists a subtle compromise between adding new vertices as well as their edges into a community and conserving the common property that defines the group. In fact, different community detection methods usually have different ways to divide a network into multiple subsets of nodes. There are many reasons that could lead to these contentions between detection methods:

[3]Community detection is identified in the research community as the search for *natural groups* in networks without a given number of clusters. When the number and the size of clusters are specified, the problem is often referred as *graph partitioning* or *graph bisection* for a division into only two clusters.

- Different algorithms may have different notions of community meaning that what an algorithm finds in a network may strongly depend on the assumptions it makes about community structure.
- When two algorithms define the same concept of community, it may also mathematically and algorithmically be formalized in different ways (the same objective but different objective functions) and hence lead us to different results.
- Even when two algorithms have exactly the same objective function, the algorithmic mechanism they employ to find communities also decides what they are going to find, especially in heuristic searching approaches.
- Initial configuration in also another important factor that affects the final result of an algorithm, many community detection methods are not deterministic.
- Each method may include a consideration between obtaining optimized results in its sense and providing a high-performance method (in terms of calculation time, memory consumption, etc.). This trade-off may be considered differently across the methods.
- Some algorithms are variable in function of input data and will prove more or less performant on some kinds of inputs than on others.
- Variations due to implementation factors could also impact the final result of an algorithm.
- Finally, in some algorithms, there are tie-break situations where the algorithm have to chose randomly without any factor related their final objectives. It may also affect heavily the result that one would get if the tie-break problems have been resolved in a different way.

A computational challenge

Searching for a good partition among all possible ways to divide a network into different parts is computationally complex, even for a small network in a graph bisection or graph partitioning problem - the simplest scenarios of community detection. In fact, an exhaustive search for every partition in a network is prohibitively expensive in terms of computation time. There is $\binom{n}{n_1} = \binom{n}{n_2} = \frac{n!}{n_1!n_2!}$ possible divisions of a graph of n vertices into two groups of n_1 and n_2 vertices, given that $n_1 + n_2 = n$. By applying Stirling's approximative formula $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$, we have approximately:

$$\frac{n!}{n_1!n_2!} \approx \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{\sqrt{2\pi n_1}\left(\frac{n_1}{e}\right)^{n_1}\sqrt{2\pi n_2}\left(\frac{n_2}{e}\right)^{n_2}} = \frac{n^{n+1/2}}{\sqrt{2\pi}n_1^{n_1+1/2}n_2^{n_2+1/2}} \tag{3.1}$$

When one needs to have two equal size communities, this number becomes $\frac{2^{n+1}}{\sqrt{2\pi n}}$. It means for a very small network such as the Zachary karate network (Zachary, 1977) consisting of 34 nodes, one have approximately in the order of billions of possible ways to divide 34 nodes into two equal-sized groups of 17 nodes. The amount of time needed for this simple case of graph bisection for an exhaustive check grows exponentially with the size of the network. In fact, finding optimal solutions for many objective functions in community detection is at least a **NP**-hard problem. In

other words, finding an optimal solution takes a non-polynomial time and even verifying whether a given solution is an optimal one according to an objective function could also require a non-polynomial time.

One of the earliest solution to graph partitioning is the *Kernghan-Lin* algorithm (Kernighan and Lin, 1970), which was inspired from the problem of distributing electronic circuits onto boards in minimizing the number of inter-board conductive wires with a constraint on the limit number of elements on each board. The algorithm greedily optimizes an objective function R representing the difference between the number of intra-board wires and the number of inter-board wires. Given an initial partition, equal-sized subsets of elements are swapped between two boards to obtain a maximal increase of R. The partition with the largest value of R is chosen after a limited number of swaps. The performance of the algorithm depends heavily on the initial configuration of the partition and can be pretty poor if one does not have additional information on the attribution of nodes (Fortunato, 2010). Moreover, the runtime complexity of the algorithm is $\mathcal{O}(n^3)$, which is not scalable for large networks. To make it even more critical, finding an exact partition to minimize the number of edges traversing clusters could be solved in $\mathcal{O}(n^{c^2})$ time, where c denotes the number of clusters (Goldschmidt and Hochbaum, 1988). Clearly, these approaches are not quite scalable for the problem of community detection. Since then, a plethora of methods have been proposed to resolve the problem. Many *heuristic algorithms* have been invented to estimate relatively good solutions for some different objective functions related to the quality of communities. These *approximation algorithms* are commonly *non-deterministic*, which means they deliver different solutions for the same input network, with different initial conditions or parameters. Among them, some notable and widely used techniques will be presented in Section 3.2. In many cases, a tolerant bound condition related to the goodness of the solution could be implicitly or explicitly required in order to determine the stopping condition of the algorithms. Regulating stopping condition in approximation algorithms can be viewed as compromising between getting an optimal quality and reducing computational complexity.

3.1.2 Vertex similarity

In traditional data clustering, a widely used approach to cluster different individuals into an unknown numbers of groups is based on the notion of *similarity*[4]. This measure is normally a function of different features selected meticulously to best characterize individuals in a way that reflects how we want data points to be distinguished. In graph clustering, it is also a natural approach to assume that a good community consists of vertices which are similar to each other. Equivalently with traditional data clustering, one can compute the similarity between each pair of vertices in a graph with respect to some proclivity properties based on local, global characteristics or both. Depending on similarity function, vertices could be similar even they are not connected in their network. Pairwise similarity scores are then used to attribute nodes into communities using conventional clustering methods such as hierarchical clustering, partitional clustering.

The simplest technique to define vertex similarity is probably to calculate the number of common neighbors that two vertices have. The more neighbors two vertices share, the more similar they are. In an undirected and unweighted network,

[4]Depending on the context, different authors may reflect this notion through an inverse notion of *dissimilarity* when the differences between individuals need to be highlighted or a notion of *proximity* or *distance* when the spatial aspect is emphasized.

the distance $d_n(i,j)$ measuring the number of common neighbors between node i and node j can be calculated as:

$$d_n(i,j) = \sum_k A_{ik}A_{kj} \tag{3.2}$$

However, this simple definition of similarity seems not be a good measure of similarity, especially for small-degree nodes which occupy a large portion of networks. Additionally, the quality of this measure also depends on the connectivity of the network in question. For example, two people having five mutual friends seems not be considered as very *"similar"*, however two cities sharing five common highways could be very similar in the functionality of a transportation system. In fact, simply counting the number of common neighbors will neglect the relative information about node connectivity. Having one common neighbor with a one-degree node would not be comparable to having one common neighbor with a one hundred-degree node.

An alternative solution that takes into account the number of total degree of each node into consideration is the *Jaccard distance*:

$$d_{Jaccard}(i,j) = \frac{\sum_k A_{ik}A_{kj}}{\sum_k A_{ik} + \sum_k A_{kj}} \tag{3.3}$$

Another metric that could palliate the problem by normalizing the number of common neighbors is the *cosine similarity*:

$$d_{cosine}(i,j) = \frac{A_{i\cdot} \cdot A_{j\cdot}}{|A_{i\cdot}||A_{j\cdot}|} = \frac{\sum_k A_{ik}A_{kj}}{\sqrt{\sum_k A_{ik}^2}\sqrt{\sum_k A_{kj}^2}} \tag{3.4}$$

Where $A_{i\cdot}$ is the ith row of the associated adjacency matrix and $\cdot$ is the dot product of two vectors. In an unweighted and simple network, $A_{ij}^2 = A_{ij}$ since A_{ij} contains only binary values, $\sum_k A_{ik}^2 = d_i$, where d_i is the degree of node i, the cosine similarity can be written as:

$$d_{cosine}(i,j) = \frac{\sum_k A_{ik}A_{kj}}{\sqrt{d_i d_j}} = \frac{d_n(i,j)}{\sqrt{d_i d_j}} \tag{3.5}$$

Simply said, the distance between two nodes is normalized by the geometric mean of their degrees. Another way to normalize the distance $d_n(i,j)$ is to compare this amount with the expected value of number of common neighbors that the nodes would take if nodes choose their neighbors randomly. In this way, we have the *Pearson similarity*[5] describing a notion of *structural equivalence* between two nodes in their network, which can be written as:

$$d_{Pearson}(i,j) = \frac{\sum_k (A_{ik} - \langle A_i \rangle)(A_{jk} - \langle A_j \rangle)}{\sqrt{\sum_k (A_{ik} - \langle A_i \rangle)^2}\sqrt{\sum_k (A_{jk} - \langle A_j \rangle)^2}} \tag{3.6}$$

Where $\langle A_i \rangle = \frac{1}{n}\sum_k A_{ik}$ is the expected degree of node i in the network. The Pearson distance measure between two node i, j varies in the range $-1 \leq d_{Pearson}(i,j) \leq 1$. A positive value means two nodes have more common neighbors than expected and a negative value indicates that they have fewer common neighbors than expected. The Pearson coefficient is widely used as a measure of similarity. Another widely used similarity metric in this group is proposed by (Leicht, Holme, and Newman,

[5] This similarity function is often referred as *Pearson correlation coefficient*

2006), who propose that two vertices are similar if their immediate neighbors in the networks are also similar.

A traditional technique to define a similarity function between pairs of vertices in to embed vertices into an multi-dimensional Euclidean space. It means that each vertex is assigned to a position described by a coordinate system. Then one could use different well-known norms to deduce the *distance* of these vertices. Supposing that two vertices i and j are embedded in an Euclidean space with different coordinates $I = (i_1, i_2, ..., i_f)$ and $J = (j_1, j_2, ..., j_f)$, one could use one of many different L_m norms to calculate the distance between two vertices:

- Manhattan distance

$$d_{Manhattan}(i,j) = \sum_{k=1}^{f} |i_k - j_k| \tag{3.7}$$

- Euclidean distance

$$d_{Euclidean}(i,j) = \left(\sum_{k=1}^{f} (i_k - j_k)^2 \right)^{\frac{1}{2}} \tag{3.8}$$

- Chebyshev distance

$$d_{Chebyshev}(i,j) = \lim_{m \to \infty} \left(\sum_{k=1}^{f} (i_k - j_k)^m \right)^{\frac{1}{m}} = \max_{1 \leq k \leq f} |i_k - j_k| \tag{3.9}$$

- Canberra distance

$$d_{Canberra}(i,j) = \sum_{k=1}^{f} \frac{|i_k - j_k|}{|i_k| + |j_k|} \tag{3.10}$$

These distances are often used in traditional data clustering to determine the similarity of different data points that need to be clustered. In the context of graph clustering, nodes in networks could be described by several features often called *attributes*. For example, in social networks where users are represented by nodes, they could be described by additional information such as: name, age, occupation, geographical position, etc. as depicted in Figure 2.2. A simple and straightforward approach to embed nodes into a normed vector space is to associate each node's attribute (as called feature) to a dimension in the Euclidean space in case that they are described by quantitative variables. In this way, each dimension in the space describe a *physical* property that characterize and distinguish nodes among them.

In order to calculate other measures of structural equivalence of nodes using these normed distances in a vector space, one could embed each node i into a n dimensional space by assigning a value of w_{ij} to the j-th dimension. The value w_{ij} is the weight of edges between node i and node j. In this way, each node is associated to a dimension and the position of a node in a dimension equals to the strength of its connection to the node associated to that dimension[6]. The normed distances previously presented could be rewritten as following:

[6]The position of a node in its associated dimension are 0 for simple graphs or can be defined as the number of self-loops

- Manhattan distance

$$d_{Manhattan}(i,j) = \sum_{k=1}^{n} |A_{ik} - A_{jk}| \tag{3.11}$$

- Euclidean distance

$$d_{Euclidean}(i,j) = \left(\sum_{k=1}^{n} (A_{ik} - A_{jk})^2 \right)^{\frac{1}{2}} \tag{3.12}$$

- Chebyshev distance

$$d_{Chebyshev}(i,j) = \lim_{m \to \infty} \left(\sum_{k=1}^{n} (A_{ik} - A_{jk})^m \right)^{\frac{1}{m}} = \max_{1 \leq k \leq n} |A_{ik} - A_{jk}| \tag{3.13}$$

- Canberra distance

$$d_{Canberra}(i,j) = \sum_{k=1}^{n} \frac{|A_{ik} - A_{jk}|}{|A_{ik}| + |A_{jk}|} \tag{3.14}$$

Besides, there are some other ways to define vertex similarity using the notion of *paths*, *geodesic distance* and *walks* between vertices. These metrics reflect the *dynamic* aspect of flows when we consider a network as a transporting medium where a vertex represents an individual and an edge represents a communication channel. This is a quite popular context when analyzing dynamical systems such as: information propagation on social networks, vehicle circulation in transport systems, communication between biological modules in living things, etc. One of the earliest way to define vertex similarity using this approach is to count the number of edge-dependent (or vertex-independent) paths[7] between them. This similarity function is inspired from the max-flow min-cut theorem (Elias, Feinstein, and Shannon, 1956) where each independent path between two vertices is considered to be a channel with limited capacity to convey a flow between a source and a destination. However, this similarity notion does not take into account the *length* of each path between vertices. In other words, a direct connection between two vertices are considered equally with a long path having the same extremities, which is sometimes an inappropriate way to model the problem. Consequently, some measures weight paths between pairs of distant vertices to decrease the influence of long paths to the similarity measure, i.e. vertices which are connected by many short paths will be more similar to each other than vertices which are connected by distant paths (Estrada, Higham, and Hatano, 2009). For instance, one could use an exponential amount of α^l to penalize paths of length l between two vertices (Katz, 1953), where $0 < \alpha < 1$ has a small value. The pairwise similarity values can be represented in a matrix form:

$$W = \sum_{l=0}^{\infty} (\alpha A)^l = [I - \alpha A]^{-1} \tag{3.15}$$

Some probabilistic approaches propose to measure vertex pairwise distances based on the probability of a *random walker* to move between two points in a limited number of steps (Harel and Koren, 2001), (Nadler et al., 2006). Among the metrics in

[7] Two paths are called edge-independent (or vertex-independent) if they do not share any common edge (or vertex) in their way.

this group, the most well-known measure distance in the context of community detection could probably attributed to (Pons and Latapy, 2005), where the authors use short-step (usually from 3 to 5 steps) random walks to define a distance between every reachable pair of vertices in networks[8] in order to detect communities in very large networks. Some other popular notions of dynamic distance is *hitting time* and *commute time* (Fouss et al., 2007), (Von Luxburg, Radl, and Hein, 2014), which represents the average number of steps required for a random walker, starting from a vertex, to reach the other vertex for the first time (hitting) and to come back to the starting vertex (commute). Further information about the utilization of stochastic processes to infer pairwise distance will be presented in later parts.

It is worth mentioning that even a plethora of vertex similarity measures exists, each one reflects a different notion of similarity which could be totally different from one to another. In reality, sometimes vertex similarity measures are used to determine, for instance, similar items for a given product, similar user profiles on entertainment platforms, etc. Community detection is just one of possible applications, which employs vertex similarity functions to group vertices into different groups. However, not every similarity metric could be well fitted in this context since in community detection, sometimes very *close* vertices are not necessarily expected to belong to a same cluster. Hence, the choice of a similarity function is very subtle to the performance of the detection method and this is still a very open subject in the research community to determine appropriate metrics and no consensus function are widely known to the best of our knowledge.

3.2 Community detection methods

We present in this section some popular community detection methods that have been widely used and discussed in the literature. Note that in recent years, there are a large number of innovative methods which are proposed to solve either generic or specific cases. However, an empirical and exhaustive analysis of all methods would be impractical if not to say unrealizable. In the best of our knowledge, we introduce the most important and representative methods among several approaches for the community detection task.

In fact, there are many possible theoretical taxonomies for community detection methods depending on the final objective of each categorization. For instance, one could classify methods according to differences in searching mechanisms, objective functions, assumptions about the structure to be found, expected qualities, hypothesis models, or even theoretical model employed, etc. Moreover, many methods are not just some simple algorithms to resolve a specific problem but instead are combinations of many different approaches in order to leverage as much as possible algorithmic power provided from each one, which makes the problem more tricky. There is not a consensus on how different methods are similar and how they can be classified into different families whose functionality can be resumed in some simple words. (Porter, Onnela, and Mucha, 2009) uses centrality based, local techniques, *modularity* optimization[9], spectral clustering to describe communities in networks. (Fortunato, 2010), (Fortunato and Hric, 2016) group community detection

[8] In fact, input network must be modified to satisfy some additional probabilistic conditions in order that the transition probability between two arbitrary vertices can be determined. Some preprocessing steps are applied on input network to assure the functionality of the method.

[9] This notion we be explained according in the community detection methods and further information could be found in Apendix A.1

methods into traditional data clustering methods, divisive algorithms, modularity-based methods, spectral algorithms, dynamic algorithms, statistical inferences based methods. (Coscia, Giannotti, and Pedreschi, 2011) summaries community discovering into feature distance based, internal density, bridge detection, diffusion process, closeness based, structural pattern based, link clustering, meta clustering. In a context of Social Media, (Papadopoulos et al., 2011) compares methods in substructure detection, vertex clustering, community quality optimization, divisive and model-based. (Bohlin et al., 2014) aggregates different approaches into three principle classes: null models, block models and flow models[10]. (Schaub et al., 2017) classifies methods into four perspectives: cut based, clustering internal density based, stochastic equivalent based and dynamical based showing four different facets of community structure.

In the following section, commonly used community detection methods are introduced according different theoretical approaches including traditional data clustering, removal based, modularity based, spectral partitioning, dynamic process based and statistical inference based. The order in which some approaches are organized may relate to some historical reasons since some methods were invented to circumvent some issues of their predecessors, to encompass an unsolved obstacle or to improve performance. Although every theoretical taxonomy can be questionable, this categorization is expected support the empirical analysis in the next chapters to answer how theoretical and conceptual closeness could engender quality closeness in practice.

3.2.1 Traditional detection methods

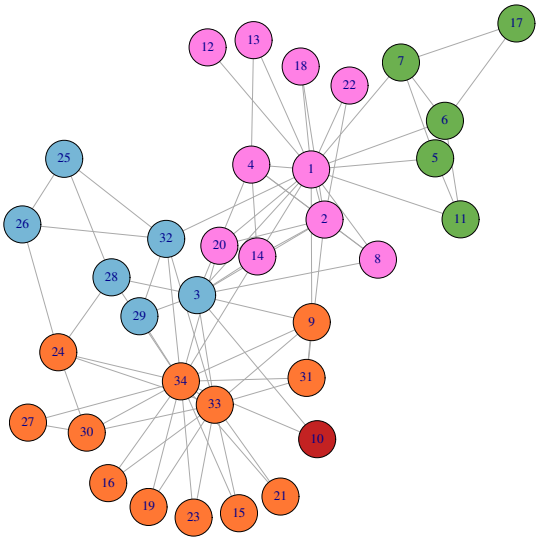
Before describing popular community detection methods, we introduce some preliminary concepts of earlier traditional methods for detecting communities. One will see in fact a smooth and gradual evolution of different concepts to resolve the problem of community detection.

In the context of social network analysis, it seems to be a very natural approach to leverage the vertex similarity information (introduced in Section 3.1.2) to detect classes of closely related individuals. Among different techniques that can employ vertex similarity to detect these kinds of groups, *hierarchical clustering* (Joe H. Ward, 1963), (Hastie, Tibshirani, and Friedman, 2009b) is probably one of the most commonly used. Depending on the sequential order that vertices are considered, the mechanism of hierarchical clustering can be separated into two groups of *agglomerative methods* and *divisive methods*.

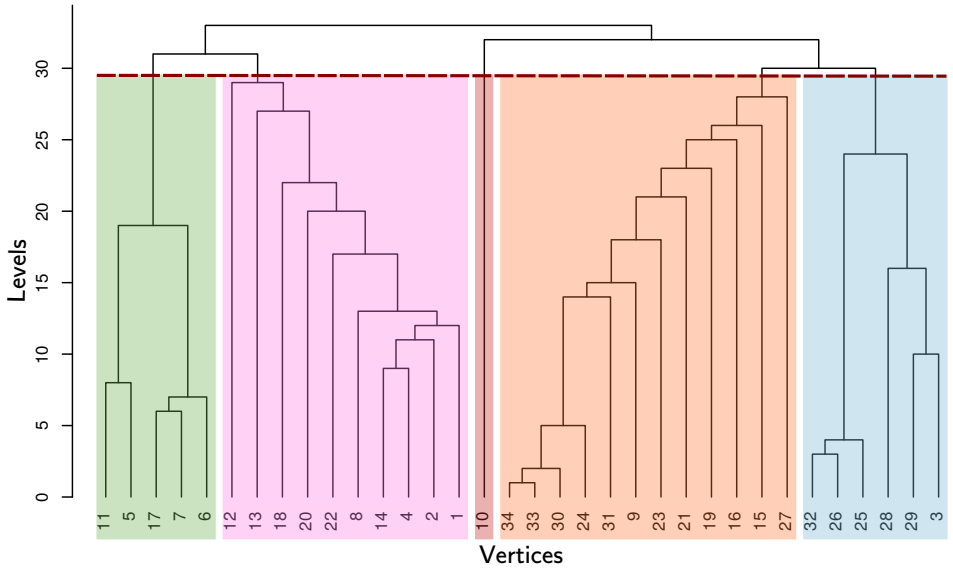
Agglomerative methods

Methods of this group consider an initial configuration of a network where each vertex belongs to its own community, i.e. there are as many communities as vertices at the beginning. Then, based on a predefined pairwise similarity function (as mentioned in Section 3.1.2), the most similar pair of vertices is aggregated into the same community. Since two communities are merged in this step, the number of communities is decreased by one and the process is iterated until all vertices are grouped together. However, it is worth mentioning that from the second iteration, one needs to define another similarity function between groups of vertices in order to decide which cluster to merge in the next steps. Widely used conventional

[10] These models can be corresponded to the methods presented in Section 3.2.3, 3.2.5 and 3.2.6



(A) The Zachary network where 34 vertices represents 34 members of the karate network and each edge represents a social tie. The network is divided in 5 classes drawn in different colors corresponding to the 30-th level of the dendrogram below.



(B) A hierarchical clustering on the Zachary network. From top to bottom of the dendrogram, there are 34 levels representing 34 different partitions. One cluster is divided into two parts when going down a level and respectively, two clusters are aggregated into one bigger cluster when going up one level. Low-level clusters are nested into higher-level clusters. The 5 groups of nodes in Figure (A) are framed into corresponding colored boxes.

FIGURE 3.2: A hierarchical clustering in the Zachary network

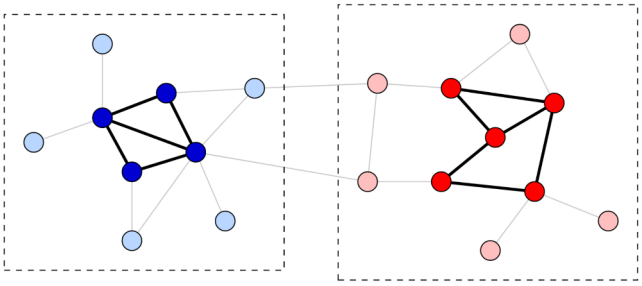


FIGURE 3.3: A typical problem of agglomerative clustering methods where highly similar nodes (connected by bold edges) are merged without their peripheral nodes, even intuitively nodes in each dashed box are supposed to belong to the same community. Reprinted figure from (Newman and Girvan, 2004) with permission RNP/18/JUN/005057 ©2004 by American Physical Society.

methods answering this call consist in *single linkage clustering*, *average linkage clustering* and *complete linkage clustering*, which calculate the distance between two clusters based on the closest distance, the average of all distances and the largest distance of every pair between two clusters respectively. In bioinformatics, another method called *neighbor-joining method* (Saitou and Nei, 1987) is used to locate a new central vertex representing each couple of vertices after their aggregation. The central vertices are designated to represent their groups of vertices in each iteration. The result of a hierarchical clustering, hence agglomerative clustering, is often delineated in form of a *dendrogram* as illustrated in Figure 3.2(B) especially by sociologists. It is a special form of *tree* with nested nodes or groups of nodes in different levels. Each level corresponds to an iteration in the process described above. Notwithstanding the fact that dendrogram contains a lot of information of a network structure, it is only convenient for small networks and is rarely used to evaluate community detection performance.

Divisive methods

Agglomerative methods shows an essential default when applied to the context of community detection due to its mechanism. In fact, it merges very high similar vertices in the early iterations and normally detects efficiently the cores of communities but not the periphery parts who are loosely connected. An illustrative example of this problem is depicted in Figure 3.3. Actually, after the cores of the two clusters have been identified, peripheral vertices continue to be allocated. However, as cores are reduced into single nodes, they are susceptible to be merged again before the allocation of distant peripheral vertices. Finally, peripheral vertices are aggregated in a quite random way without knowing their core. Moreover, the mechanism of taking only the closest element without considering the trade-off with other elements conducts to a large quantity of errors, especially in real world networks where low degree nodes are omnipresent.

Methods using divisive approach employ a reverse process to discover hierarchical clusters in networks. Instead of accumulating clusters from local areas in networks, they explore the network of interest from a global view with the presence of all vertices and edges. The whole network is recurrently cut into smaller parts. Equivalently, a divisive process can be literally translated as a top-down procedure in the dendrogram presented in Figure 3.2b. At the beginning, least similar connected couples of vertices are identified and the corresponding edges are removed

until the network are disconnected. Then the process is repeated and disconnected sub graphs become smaller and smaller until every node are separated. In other words, divisive methods search for inter-community edges based on the hypothesis that they connect low similar vertices, which is not always the case.

A common point between these two classes of methods is that they are very sensitive to the definition of similarity function in use. The same hierarchical clustering mechanism can provide totally different partitions if one modifies slightly the assumption of similarity. Another reason why hierarchical clustering methods are not directly used alone is due to the fact that they all provide groups of different partitions. Hence, there must be additional quality measures to evaluate different hierarchical levels discovered. Such a combination will be presented in detail throughout particular methods in the following sections.

3.2.2 Centrality removal based approach

Girvan-Newman's method

Being a member of the divisive family, this method, on the contrary to agglomerative methods, aims to find communities by removing edges progressively to disconnect tightly-knit groups of vertices. Firstly introduced in a former version (Girvan and Newman, 2002) and then complemented with a more detailed version in (Newman and Girvan, 2004), the method has exploded many research interests in the field of community detection. One of the two elements that make the method much more competitive in solving the problem of community detection with respect to the other homologous methods is the utilization of *edge betweenness*[11] instead of using conventional similarity functions. Specifically, it based on an intuition that if there are communities in a graph who are only loosely connected by a few inter-group edges, then shortest paths between vertices in different communities must go along these few edges, making their edge betweenness centrality higher than those of intra-community edges. Consequently, if one could detect and then remove these *border edges* assumed having high betweenness scores, community structure will be highlighted. The concept is illustrated in Figure 3.4 where inter-community edges are gray. In case when there are several shortest paths going between two vertices, the final contribution of each path for the centrality of its edges are normalized. For example, if there are three shortest paths going between two vertices, each edge on the three paths will be assigned an additional value of $\frac{1}{3}$ to its total centrality score.

The divisive method of Girvan and Newman is presented in Algorithm 1, where $BETWEENNESS(\mathcal{G})$ is the calculation of edge betweenness scores of graph G presented in Algorithm 3. Summarily, in order to determinate communities, the algorithm executes the following steps:

1. Calculate betweenness scores for all edges in the graph.
2. Remove the edge with the highest betweenness score.
3. Recalculate betweenness scores for all edges after the removal
4. Repeat from step 2 until no edges remain.

[11] The betweenness centrality of an edge is slightly different with the betweenness centrality of (Freeman, 1977) presented in Section 2.2.2.

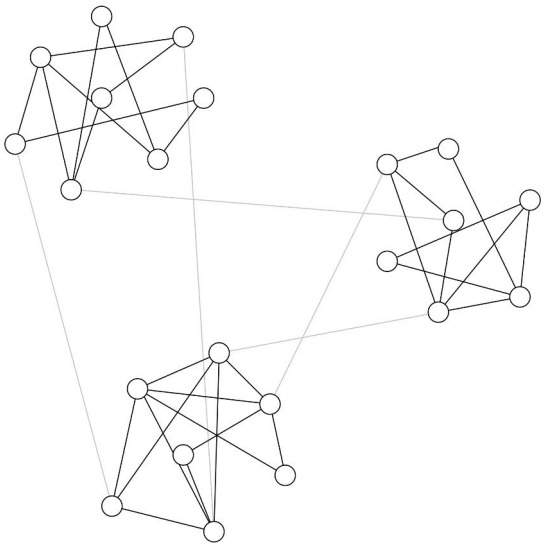


FIGURE 3.4: A graph with community structure. The gray edges that connect different communities has the highest edge betweenness among all edges since all shortest paths between two communities must go through them. Reprinted figure from (Girvan and Newman, 2002) with permission. ©2002 National Academy of Sciences.

Algorithm 1: Girvan-Newman's method

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Output: Edge removal sequence S

$\mathcal{G}_\pi = (\mathcal{V}, \mathcal{E}_\pi) \leftarrow \mathcal{G} = (\mathcal{V}, \mathcal{E})$
$S \leftarrow$ empty list
while $\mathcal{E}_\pi \neq \emptyset$ **do**
  $C_B[u, v] \leftarrow BETWEENNESS(\mathcal{G}_\pi), \forall (u, v) \in \mathcal{E}_\pi$
  $e = (u_m, v_m) \leftarrow max(C_B[u, v])$
  $\mathcal{G}_\pi(\mathcal{V}, \mathcal{E}_\pi) \leftarrow \mathcal{G}_\pi(\mathcal{V}, \mathcal{E}_\pi - \{e\})$
  $APPEND(S, e)$
end
return S

The third step of is the second essential element that make the success of this method that outperforms traditional hierarchal clustering algorithms. In fact, the authors discovered that in some cases, edge betweenness centrality scores may not distributed equally all along every inter-community edge, which makes many of them invisible before the suppression of the most central edge. Hence, it is necessary to recalculate every betweenness score after the removal of any edge. However, this is also the action that produce a very high complexity in the calculation time of the method, which make it less competitive to large-scale graphs, even when a fast method are implemented to calculate betweenness centrality. This method requires $\mathcal{O}(m^2 n)$ time to calculate complete community structure in worst case and is reduced to $\mathcal{O}(n^3)$ on a spare graph, which is quite infeasible for graphs with more than a ten thousand of vertices even when parallel computing is applied. The algorithm used to calculate edge betweenness centrality is described in Appendix A.2.

Since the Algorithm 1 produces a hierarchical structure of communities, which consists in a nested structure of partitions, it is not practical if one need to compare the quality of different partitions or to work with large-scale graphs. The author

also proposed a new quality fitness function called *modularity* to compare the goodness of different partitions. This function has been inspiring an enormous number of work in the scientific community and became a golden standard to compare the quality of community detection algorithms[12]. The modularity is invented based on an intuition that a good partition must contain communities where edges inside communities are more present than one would expect if edges are distributed around vertices in a random way. Specifically, the modularity function compares the difference between the fraction of edges inside communities with the expected fraction of those edges in a *null model* where edges are redistributed in remaining the expected graph's degree sequence. The modularity function is very polyvalent in the context of community detection. It can be used as objective function in optimization process, as quality function in the evaluation of community structure or as a decision function in hierarchical clusterings. The formula of modularity function is can be found with further details in Appendix A.1.

Radicchi *et al.*'s method

We introduce in this part another divisive method proposed by (Radicchi et al., 2004). As presented in the previous section, the method of Girvan and Newman requires repetitive calculations of betweenness centrality, which is a global quantity and expensive to calculate. Hence, Radicchi *et al.* proposed to replace the betweenness centrality by a class of local quantities that is easier to compute. Being local quantities, they also require less time to recalculate after each removal since there is only a small part of the graph is affected.

The method of Radicchi *et al.* exploits the topology of the graph of interest to detect communities. The principle argument of the authors relies on an assumption of a structural property of community structure that *"edges connecting nodes in different communities are included in few or no triangles"*. From this intuition, the proposed method considers the *edge-clustering coefficient*[13], which measures the fraction between the number of triangles to which an edge belong with the maximum number of triangles lied on that edge that could be established. Formally, the edge-clustering coefficient is defined as:

$$C^{(3)}(i,j) = \frac{|\Delta_{ij}|}{min[(d_i - 1), (d_j - 1)]} \tag{3.16}$$

where $|\Delta_{ij}|$ represents the number of triangles constructed based on edge (i, j) and $min[(d_i - 1), (d_j - 1)]$ is the maximal number of possibly constructed from that edge with the neighbors of its extremities i and j. However, the coefficient becomes degraded when an edge does not participate to any triangle. In this case, $C^3(i, j) = 0$ regardless of d_i, d_j and hence it hides the degree information of i and j. Moreover, when $min[(d_i - 1), (d_j - 1)] = 0$, the coefficient is indeterminate, hence it is only applicable for edges whose extremities have degrees higher than 1, which is quite reasonable since nodes connecting to only one neighbor can be naively grouped to the community of their neighbors. A modified version of edge clustering coefficient is hence suggested:

[12] Despite of some critics indicating its defaults in discovering small-size communities and many advanced modifications, the modularity quality function is still very widely used and, in the best of our knowledge, there is currently no quality function that could represent all goodness aspects.

[13] By definition, this topological quantity is very similar to the common node clustering coefficient presented in Section 2.2.2, but instead of quantifying triadic closure by vertex, it measures on edges.

$$\widetilde{C}^{(3)}(i,j) = \frac{|\Delta_{ij}| + 1}{min[(d_i - 1), (d_j - 1)]} \tag{3.17}$$

The edge clustering coefficient is also generalized to higher orders of cycle:

$$\widetilde{C}^{(g)}(i,j) = \frac{z_{ij}^{(g)} + 1}{s_{ij}^{(g)}} \tag{3.18}$$

where $z_{ij}^{(g)}$ represents the number of cyclic structure of order g to which the edge (i,j) belongs, and $s_{ij}^{(g)}$ is the maximum number of cyclic structures of order g that can be built.

The method is represented in Algorithm 2, where $n_{ij}^{(g-3)}$ denotes the number of shortest path of distance $(g-3)$ between i and j. In case $g = 3$ or 4, which is recommended by the authors, $n_{ij}^{(g-3)}$ becomes the Kronecker function $\delta(i,j)$ or the value a_{ij} of the adjacency matrix A respectively. Note that in this method, instead of removing the edge containing the highest betweenness centrality score, the edge having the smallest value of edge clustering coefficient is removed as indicated in line **14** of Algorithm 2. Since the smaller the value of the coefficient, the more possible that the corresponding edge is inter-community. Furthermore, the authors also reinforce their argument about the replacement of betweenness centrality by clustering coefficient by illustrating an anti-correlation of these measures on some networks. In this way of view, removing edges based on these two metrics are statically equivalent.

In terms of complexity, since the clustering coefficients of distant edges from a removed edge are not affected, the calculation from line **5** to line **12** of Algorithm 2 are recomputed only in a small sub-graph. Hence, the time consumption of Radicchi *et al.* method is much inferior to that of the Girvan and Newman. The method takes $\mathcal{O}(m^4/n^2)$ in comparison to $\mathcal{O}(m^2 n)$ of the edge betweenness method, hence can tackle larger graphs.

Algorithm 2: Radicchi *et al.* method

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}), g \in \mathbb{N}, g \geq 3$
Output: Edge removal sequence S

$\mathcal{G}_\pi = (\mathcal{V}, \mathcal{E}_\pi) \leftarrow \mathcal{G} = (\mathcal{V}, \mathcal{E})$
$S \leftarrow$ empty list
while $\mathcal{E}_\pi \neq \emptyset$ **do**
  for $(u, v) \in \mathcal{E}_\pi$ **do**
    $z_{uv}^{(g)} \leftarrow 0$
    for *each* $i \in \mathcal{G}_\pi.Adj[u]$ **do**
      for *each* $j \in \mathcal{G}_\pi.Adj[v]$ **do**
        $z_{uv}^{(g)} \leftarrow z_{uv}^{(g)} + n_{ij}^{(g-3)}$
      end
    end
    $s_{uv}^{(g)} = min[(|\mathcal{G}_\pi.Adj[u]| - 1)(|\mathcal{G}_\pi.Adj[v]| - 1)]$
    $C^{(g)}[u, v] \leftarrow \frac{z_{uv}^{(g)} + 1}{s_{uv}^{(g)}}$
  end
  $e = (u_m, v_m) \leftarrow min(C^{(g)}[u, v])$
  $\mathcal{G}_\pi(\mathcal{V}, \mathcal{E}_\pi) \leftarrow \mathcal{G}_\pi(\mathcal{V}, \mathcal{E}_\pi - \{e\})$
  $APPEND(S, e)$
end
return S

3.2.3 Modularity optimization based approach

Clauset-Newman-Moore's method

Clauset *et al.* proposed a hierarchical agglomeration algorithm for detecting community structure by optimizing greedily the modularity quality function (Clauset, Newman, and Moore, 2004), which is an improved version of (Newman, 2004) in terms of time and memory consumption. Although having a similar discovering process with that of conventional hierarchical clustering, the method shows a good performance in practice thanks to the utilization of modularity. Since it is not compulsory to define a proximity function between a vertex and a community or between two vertices[14], the method is self-consistent to the assumption of community quality.

The main concept of this method is grouping repeatedly communities in a graph together in order to acquire a maximum increase ΔQ of modularity from the agglomerative action. The algorithm begins with an initial partition of a graph where each vertex belongs uniquely to its own community, meaning there are in total as many communities as vertices, and finishes when all vertices are grouped in the same community. Since calculating the gain $\Delta Q_{c_i c_j}$ of merging two communities c_i and c_j as well as finding the pair c_i, c_j with the largest $\Delta Q_{c_i c_j}$ are time-consuming, the authors employ efficient data structures to reduce computational complexity.

First, the modularity function is represented in a convenient form allowing fast updates of changes after each step:

[14] In fact, the modularity function implicitly determines how different communities are *similar* by the gain/loss of its value when they are placed together.

$$Q = \sum_{c_i} (e_{c_i c_i} - a_{c_i}^2), \tag{3.19}$$

where $e_{c_i c_j}$ represents the fraction of edges that join vertices in community c_i to vertices in community c_j:

$$e_{c_i c_j} = \frac{1}{2m} \sum_{uv} A_{uv} \delta(c_u, c_i) \delta(c_v, c_j), \tag{3.20}$$

and a_{c_i} is the fraction of degrees of vertices belonging to community c_i:

$$a_{c_i} = \frac{1}{2m} \sum_{u} k_u \delta(c_u, c_i), \tag{3.21}$$

with k_u is the degree of vertex u. In order to reduce the number of calculations, the changes of modularity are only calculated for communities that are connected since joining two distant communities can not produce any increase in Q. At the initial step, when each vertex belongs to each own community:

$$\Delta Q_{c_i c_j} = \frac{1}{2m} - \frac{k_i k_j}{4m^2}. \tag{3.22}$$

Thus the matrix $\Delta Q_{c_i c_j}$ representing the changes of modularity is sparse, its rows can be represented by balanced binary trees for fast searches and also by max-heaps (one by row) to find the largest value of ΔQ_{c_i}. in constant time. A max-heap H is also used to store the largest elements of rows of the ΔQ matrix. The fast algorithm can be executed:

1. Compute initial values of $\Delta Q_{c_i c_j}$ and a_{c_i} then the max-heap H populated by $max_{c_j}(\Delta Q_{c_i}.)$.
2. Select the largest $\Delta Q_{c_i c_j}$ from H and merge corresponding communities. Update $\Delta Q, H, a_{c_i}$ and Q by $\Delta Q_{c_i c_j}$.
3. Repeat step 2 until all vertices belong to one community.

The changes of modularity gain after each aggregation in step 2 can be calculated easily. By naming c_j the community resulting from merging c_i and c_j, the change of modularity gain when merging the new c_j and c_k is:

$$\Delta Q'_{c_j c_k} = \begin{cases} \Delta Q_{c_i c_k} + \Delta Q_{c_j c_k}, & \text{if } c_i, c_j, c_k \text{ are connected ,} \\ \Delta Q_{c_i c_k} - 2 a_{c_j} a_{c_k}, & \text{if } c_k \text{ is not connected to } c_j, \\ \Delta Q_{c_j c_k} - 2 a_{c_i} a_{c_k}, & \text{if } c_k \text{ is not connected to } c_i. \end{cases} \tag{3.23}$$

The algorithm find communities in $\mathcal{O}(md\log(n))$ time where d is the depth of the dendrogram describing the community structure. In practice, it is often estimated that $d \sim \log(n)$ leading the to a total time complexity of $\mathcal{O}(m\log^2(n))$, which makes the algorithm being the fastest at the time for discovering community structure using an optimization of modularity.

Blondel *et al.*'s method

The method of Clauset-Newman-Moore (*CNM*) sometimes discloses two large communities in detecting community structure of large networks. Also, it sometimes

identify partitions whose modularity are significantly lower than what could be found using some traditional optimization processes (Blondel et al., 2008). Hence, Blondel *et al.* proposed a new method called *Louvain* based on a very similar concept of that of the Clauset-Newman-Moore's method, which can palliate the above problems and can work on very large networks.

This method also try to maximize the modularity function by aggregating iteratively communities together in order to obtain a maximum increase of of modularity. Nevertheless, the process is divided into two iterative phases. Similarly, Blondel *et al.*'s method also initiates the beginning state as a partition of one-vertex communities. In the first phase, every vertex in the graph is considered to be moved to the community of one of its neighbors in order to acquire a maximum improvement of modularity, but in contrast to *CNM*'s method, only if the gain is positive. Another difference of this step with the method of *CNM* is lied on the possibility of removing a vertex from its community after it has been merged provided that there is an interest. This possibility is enable thanks to two mechanisms:

- A vertex that has been visited can be revisited several times,
- The aggregation is not considered merely between two communities, but between a vertex and a community which allow a higher flexibility.

The process is repeated until no further improvement of modularity can be achieved knowing that the modularity change after each moving can be calculated using Equations (3.19), (3.20), (3.21).

After the first phase has been done, the modularity is locally optimized. The algorithm continues to build a multi-graph whose vertices[15] are the communities found in the end of the first phase and edges are edges between those communities[16]. Once the second phase is finished, the first phase are reapplied and the process continues until only one community remains.

In terms of complexity, since the number of vertices to be considered decreases substantially after each construction of meta-graph, the time needed to discover community structure depends essentially on the first pass. For example, the algorithm needs only 6 passes to discover a community structure of an ad-hoc graph of 10000 vertices. However, the computation time of the algorithm is contingent on the order that vertices are analyzed, hence is not deterministic. Although unknown, the time complexity is estimated at $\mathcal{O}(n \log(n))$ which is one of the biggest of its advantages. It can work on graphs up to 100 million nodes and billion of edges.

3.2.4 Spectral partitioning approach

The division of a graph into subgraphs by optimizing a quality of the partition such as number of inter-community edges can be resolved by an approach called spectral partitioning. There are intense work dedicated to study network structure through the spectrum of derivatives of adjacency matrix associating to the graph in the literature of computer science field (Pothen, Simon, and Liou, 1990), (Schaeffer, 2007). The main concept of this approach is to represent a desired quality of partitions using a spectral decomposition of a matrix corresponding to the graph. Then, finding a partition to optimize the quality function is equivalent to optimizing the associated spectral decomposition. The most well-known method using spectral approach for

[15] Sometimes called *meta-vertices* as they represent a groups of vertices.

[16] In this multi-graph, vertices can have self-loops representing intra-community edges in the first step and edges can be weighted according to the number of inter-community edges in the first step

graph partitioning is probably the optimization of partition cut size using the spectrum of *Laplacian matrix*[17]. Even able to find good clusters, spectral bisection methods using the Laplacian matrix is known being not a good solution for discovering community structure in networks (Fortunato, 2010), (Nascimento and Carvalho, 2011) as it requires preliminary assumptions on the cluster sizes (see Appendix A.3).

Several solutions have been proposed to palliate the constraint. Among them, the most popular and well-known solution using spectral approach is possibly attributed to (Newman, 2006). By representing the modularity function in a matrix form, the author demonstrates that the task community structure can be conducted using a spectral decomposition approach in a similar way of classical spectral partitioning using Laplacian matrix. The method is presented in the following part.

Newman's method

The modularity can be used as an objective function in other to search for communities in graphs. Newman adapted the modularity function in order to execute a spectral approach to discover community structure (Newman, 2006). The concept of this method hence also rely on the idea that a good community should have more edges between its nodes than edges connecting it with other communities. The modularity function from Equation (A.1) is rewritten in a matrix form as following:

$$\mathbf{B} = \mathbf{A} - \mathbf{P}, \tag{3.24}$$

where $\mathbf{A}$ is the adjacency matrix and $\mathbf{P}$ is the matrix containing the expected number of edges falling between each pair of vertices. Matrix $\mathbf{B}$ is called *modularity matrix* and its values for a given graph depends on how one chooses the null model expressing by the matrix $\mathbf{P}$. This choice reveals the notion of an equivalent randomized network model in which community structure is considered to be negligible, hence *null*. The standard choice for null model that works very well in the literature mentioned in Equation (A.1) can be rewritten:

$$P_{ij} = \frac{k_i k_j}{2m}, \tag{3.25}$$

where k_i and k_j represents the degree of vertex i and vertex j respectively; m is the number of edges in the graph. By employing the same index vector $\mathbf{s}$ as presented above, the modularity in Equation (A.1) can be reformulated as:

$$Q = \frac{1}{4m}\sum_{ij}[A_{ij} - P_{ij}](s_i s_j + 1) = \frac{1}{4m}\sum_{ij}[A_{ij} - P_{ij}]s_i s_j, \tag{3.26}$$

given that $\sum_{ij} P_{ij} = \sum_{ij} A_{ij} = 2m$ and $\sum_j P_{ij} = k_i$, the modularity matrix is real and symmetric. The familiar form of the spectral clustering approach expressed in Equation (A.10) can be found in this case:

$$Q = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s}, \tag{3.27}$$

where $\mathbf{s} = \sum_{i=1}^{n} a_i \mathbf{v}_i$ with $a_i = \mathbf{v}_i^T\mathbf{s}$. Similarly to the previous processes, the modularity can be represented by a linear combination of normalized eigenvectors $\mathbf{v}_i$ corresponding to eigenvalues λ_i of matrix $\mathbf{B}$:

[17]The Laplacian matrix is very common in the problem of graph partitioning using spectral approach, which aims to optimize objective functions based on an eigen-decomposition of the matrix.

$$Q = \frac{1}{4m} \sum_i a_i^2 \lambda_i. \tag{3.28}$$

The only difference between this case with the previous case is that it consists in a maximization of modularity instead of minimization. Hence, the task of community detection is then translated into choosing $\mathbf{s}$ in such a way to maximize coefficients a_i^2 weighting the largest eigenvalues. If we label eigenvalues in an decreasing order $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$, again the straightforward solution would be choosing $\mathbf{s}$ to be as close to parallel with the *leading eigenvector* $\mathbf{v}_1$ as possible. A good solution could be obtained by assigning:

$$s_i = \begin{cases} \pm 1 \text{ if } v_i^{(1)} \geq 0, \\ \mp 1 \text{ if } v_i^{(1)} < 0. \end{cases} \tag{3.29}$$

The choice of modularity as objective function in the spectral approach inherits a remarkable advantage over the traditional cut size as it does not impose the sizes of clusters. Hence, this method is much more suitable for detecting *natural* groups than the traditional graph partitioning. In order to discover more than two communities, the author suggest several solutions. Among them, the simplest one consisting in a subdivision strategy work nicely in practice. Specifically, after each partitioning one continues to divide repeatedly communities into smaller communities. In function of the modularity contribution ΔQ that the division of a community provides in each iterative step, the process will be stopped if ΔQ is not positive. Other relating solutions to divide networks to more than two communities by using *vector partitioning* to maximize the modularity of Equation (3.28) can be found in the original paper (Newman, 2006). Again, the complexity of the method relies principally on the calculation of the largest eigenvector, hence the multiplication of matrices. Even the modularity matrix is usually not a sparse one, an appropriate decomposition of $\mathbf{B}$ can executes the task $\mathcal{O}(n(n+m))$ time. In practice, one needs $\mathcal{O}(log(n))$ iterative steps corresponding to the depth of the dendrogram describing the community structure, which makes the final time complexity up to $\mathcal{O}(n(n+m)\log(n))$.

3.2.5 Dynamic process based approach

This approach of community detection, instead of using directly topological structure of networks, captures the behaviors of dynamic process models that could occur on the associated real systems to infer meaningful modules. Naturally, methods based on this approach make use of a random walk process describing real world phenomena such as information propagation on networks, and from that extract patterns followed by the random walker to constitute communities.

Rosvall *et al.*'s method

First presented in (Rosvall and Bergstrom, 2008), (Rosvall, Axelsson, and Bergstrom, 2009), the method of Rosvall *et al.* is probably the most well-known method in this category using flow models. Since its first apparency, many improvements and functionalities have been added to the initial method making it become a powerful network analysis tool. Nevertheless, we present here the principle and original concept in which consist the core functionality of the method to resolve the task of community detection. The method is commonly called *Infomap*.

In this method, the task of detecting communities can be translated into the task of finding a two-level encryption coding rule to minimize the total codewords

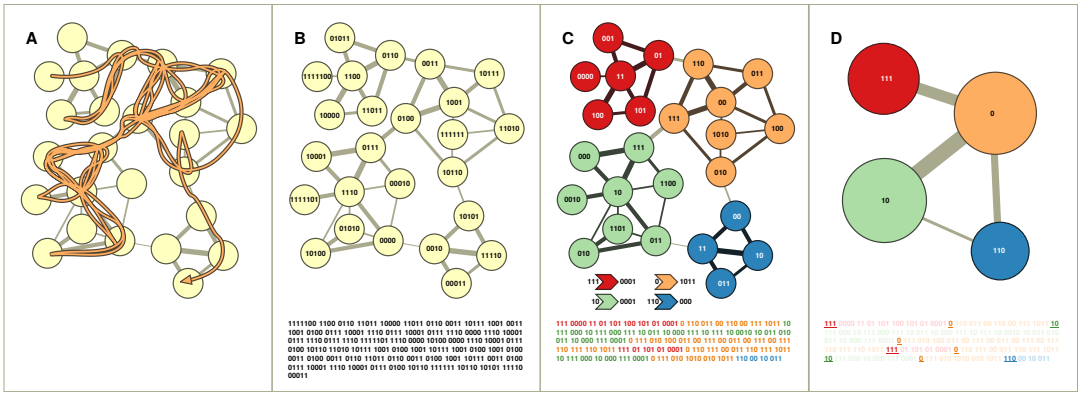


FIGURE 3.5: Communities are detected by optimizing the compression rate of the description of information flows on the network. (A) The description of the trajectory of a random walk on a network highlights network structure. (B) The traditional approach in information theory to to encode vertices using Huffman code needs 314 bits (below the graph) to describe the random walk trajectory. (C) If vertices are encoded using a two-level description (cluster code + vertex code), only 243 bits (below the graph) are needed to describe the same trajectory. (D) Community level description of the graph illustrated in (A) where a vertex are encoded by short-length binary numbers. Reprinted figure from (Rosvall and Bergstrom, 2008) with permission ©2008 National Academy of Sciences.

needed to describe paths of random walks on the graph[18]. An analogous way to interpret this approach is the technique used in cartography. It is not necessary to use unique name for every district and street on over the world making them extremely long, one can reused the same names in geographically distant areas. There can be two streets in two different cities having the same name. If we consider streets being vertices in a graph and cities being communities, we can employ a similar technique to index vertices in a way that the description length needed is minimized. This concept is illustrated in Figure 3.5

From the presented idea, the detection of communities can be done through minimizing a special function called the *Map equation* expressing the average number of bits per step needed to describe an infinite random walk using a two-level encoding corresponding to the partition. By denoting $\mathcal{P} = \{P^1, P^2, ..., P^m\}$ as a partition of graph $\mathcal{G}$, this average description length can be expressed as:

$$L(\mathcal{P}) = q_{\curvearrowright} H(Q) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(P^i). \tag{3.30}$$

The first term of the description $q_{\curvearrowright} H(Q)$ consists in the entropy of the movement between communities and the second term $\sum_{i=1}^{m} p^i_{\circlearrowright} H(P^i)$ is the entropy of the within-community movements. In the equation, $q_{\curvearrowright}$ is the probability that the random walk switches communities at any moment and $p^i_{\circlearrowright}$ is the probability that the random walker stay inside community i at any step. Besides, Q and P^i are the distributions of visit frequencies to the communities in the graph and to the vertices in the community i respectively. The entropy $H(Q)$ and $H(P^i)$ are the lower limits of the averages length of codewords used to label communities and vertices of community i respectively. They are calculated according to Shannon's source coding theorem

[18]The authors then extended the method to a multilevel code length compression to identify hierarchical modular structure (Rosvall and Bergstrom, 2011)

(Shannon, 1948). The shorter the description length, the higher the performance of the corresponding partition $\mathcal{P}$ in compressing code's length.

In order to find a partition that has a low value of description length, a greedy search is first used. Each vertex is assigned to unique community and then communities are chosen to be merged in a way to get the largest decrease in description length (similar to the *CNM* method). This step is repeated until the length can not be reduced anymore. In the second step, a refinement is applied using simulated annealing technique (Kirkpatrick, Gelatt, and Vecchi, 1983) with the initial partition provided by the greedy search. Finally, the partition corresponding to the smallest description length is consider to be the best solution. The time complexity of the method is estimated to be linear at $\mathcal{O}(m)$.

The earlier information-theoretic approach

Additionally, an earlier concept of this approach for the task of community detection had been published by the same authors (Rosvall and Bergstrom, 2007) using an information theoretic model. In this earlier version, the graph of interest in considered as the information that need to be transfered over a limited capacity transmission channel. A *signaler* knows the full structure of the graph and wants to send as much information as possible to a *receiver* through this channel. Therefore, the signaler must encode the graph $\mathcal{G}$ into community structure $\mathcal{C}$ in a way that minimizes the transfered information $L(\mathcal{C})$ and the information loss about the graph at the side of the receiver $L(\mathcal{G}|\mathcal{C})$. The loss means the additional information that is needed to specify exactly the full graph structure after the receiver had already decoded the modular structure information. The method hence minimizes objective function:

$$L = L(\mathcal{C}) + L(\mathcal{G}|\mathcal{C}) \tag{3.31}$$

Finally, the simulated annealing method is also use simulated annealing technique to find the optimized configuration.

Pons-Latapy's method

Another widespread solution of community detection using dynamic process based approach, commonly called *Walktrap*, has been introduced in (Pons and Latapy, 2005). The common point of this method with the method of Rosvall *et al.* is, of course, the exploitation of stochastic process on the network of interest in order to discover community structure. However, instead of calculating the regularity of an infinite walk on the graph, the authors proposed a new measure of vertex proximity based on short step random walks. After defining a distance function representing vertex structural similarity, the problem of community detection is equivalent to traditional data clustering. An agglomerative clustering based on Ward's method (Joe H. Ward, 1963) is applied on the graph in order to find significant communities. It is similar to the one of *CNM*'s method presented in Section 3.2.3. However, instead of maximize the increase of modularity in each agglomerative step, Pons-Latapy's method minimize the increase of average Euclidean distance causing by the fusion of two communities. Finally, the algorithm provides a hierarchical structure of community after n fusions. In order to choose the best partition, the method has the same approach with Girvan-Newman's method and Radicchi's method, meaning the partition corresponding to the highest modularity is taken.

Going into the innovative aspect making the difference of the method, it is argued that if two vertices i and j are in the same community, the probability P_{ij}^t that

a random walker goes from i to j in t steps must be high and the distance between i and j must be small. Moreover, *"two vertices belonging to the same community tend to "see" all other vertices in the same way"*, which means the probability of going from i and j to other vertices in the graphs are similar: $P_{ik}^t \approx P_{jk}^t, \forall k \in \mathcal{V}$ (see Section 2.2.1). From these arguments, a function of distance is defined:

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \frac{(P_{ik}^t - P_{jk}^t)^2}{d(k)}} = ||\mathbf{D}^{-1/2}\mathbf{P}_{i\cdot}^t - \mathbf{D}^{-1/2}\mathbf{P}_{j\cdot}^t||, \tag{3.32}$$

where $d(k)$ is the degree of vertex k, $\mathbf{D}$ is the diagonal degree matrix, $\mathbf{P}_{i\cdot}^t$ is the i-th column of the matrix $\mathbf{P}^t$ representing the probabilities of going between two vertices in t step. Similarly, the distance between two communities C_i and C_j is straightforward:

$$r_{C_iC_j} = \sqrt{\sum_{k=1}^{n} \frac{(P_{C_ik}^t - P_{C_jk}^t)^2}{d(k)}} = ||\mathbf{D}^{-1/2}\mathbf{P}_{C_i\cdot}^t - \mathbf{D}^{-1/2}\mathbf{P}_{C_j\cdot}^t||, \tag{3.33}$$

where $P_{C_ij}^t = \frac{1}{|C_i|}\sum_{i\in C_i} P_{ij}^t$ is the probability of going from community C_i to vertex j in t steps.

In terms of complexity, by using properties of Euclidean distance and considering only adjacent communities for the fusion step, the algorithm's global complexity is estimated at $\mathcal{O}(mn(d+t))$. Given t being the length of the walk often chose at $t \approx \mathcal{O}(\log(n))$, d being the height of partition's dendrogram estimated at $\mathcal{O}(\log(n))$ in balanced trees, the complexity is $\mathcal{O}(mn\log(n))$ and is reduced to $\mathcal{O}(n^2\log(n))$ in sparse graphs.

3.2.6 Statistical inference approach

Statistical inference is also leveraged to discover community structure in networks and usually provides good results. The principled concept of this approach is from the idea that empirical graphs that are observed are the results of some latent models described by sets of parameters. Several methods using this approach employ different variants of stochastic block models to infer the likeliness that a given graph is generated from compatible models, and then suggest the most likely sets of model parameters. The community structure of the graph is then obtained from the configuration of the model that is the most representative for the observed graph. If these methods are applied to graphs that are likely to be generated from the same block model, they can correctly recover the block structure. A meticulous review about different probabilistic models that can be developed to solve the task of community detection is recently published in (Peixoto, 2018). In this empirical analysis, we present one of the most popular variant considered as the standard *Stochastic Block Model (SBM)* and its corrected version *Degree-Corrected Stochastic Block Model (DC-SBM)* adapted for the problem of community detection (Karrer and Newman, 2011). Here, we are introducing only some representative approaches.

Stochastic Block Model

In the standard SBM, graph $\mathcal{G}$ of n vertices and m edges can be generated from a probabilistic generative block model with a hidden group structure c assigning each vertex i to a community c_i, where $c_i \in \{1, ..., k\}$ given that k is the number

of communities. Supposing that edges are placed between any pair of vertices i and j independently and randomly so that vertex degrees follow Poisson distributions. In this block model, the expected value of the adjacency matrix element A_{ij} only depends on the groups to which i and j belong. Denote these groups c_i and c_j for vertex i and j respectively, this expected value of edges between i and j is the $\omega_{c_i c_j}$ element of matrix ω revealing the block structure of the generative model. If $\forall r \neq s \in \{1, ..., k\}, \omega_{rr} > \omega_{rs}$, there could be a latent community structure in graphs generated from the model. Note that in this model, even vertices in group r could have self-edges with an expected value of $\omega_{rr}/2$ and such phenomenon is rare in real world networks, the model used is not really effected when n becomes large. Community structure can be estimated from the probability $P(G|\omega, c)$ of graph $\mathcal{G}$ given the block matrix and the group assignment c as following:

$$P(G|\omega, c) = \prod_{i<j} \frac{(\omega_{c_i c_j})^{A_{ij}}}{A_{ij}!} exp(-\omega_{c_i c_j}) \prod_i \frac{(\frac{1}{2}\omega_{c_i c_i})^{\frac{A_{ii}}{2}}}{(\frac{A_{ii}}{2})!} exp(-\frac{1}{2}\omega_{c_i c_i}), \tag{3.34}$$

given that A and ω are symmetric.

However, this traditional *SBM* does not work well in practice for the task of community detection since it does not take into account the variation of degrees often observed in real networks. In fact, since edges are only distributed among vertices of a graph in function of blocks to which they belong regardless of their real degrees, similar degree vertices tend to be grouped into the same blocks creating unrealistic communities. Consequently, a corrected version of the stochastic block model called DC-SBM is proposed (Karrer and Newman, 2011) (Riolo et al., 2017), taking into consideration of node degree by introducing parameters θ_i that control the expected degree of vertices. In this version, the expected value of A_{ij} becomes $\theta_i \theta_j \omega_{c_i c_j}$. The probability of the observed graph in Equation (3.34) can be rewritten:

$$P(G|\omega, c) = \prod_{i<j} \frac{(\theta_i \theta_j \omega_{c_i c_j})^{A_{ij}}}{A_{ij}!} exp(-\theta_i \theta_j \omega_{c_i c_j}) \prod_i \frac{(\frac{1}{2}\theta_i^2 \omega_{c_i c_i})^{\frac{A_{ii}}{2}}}{(\frac{A_{ii}}{2})!} exp(-\frac{1}{2}\theta_i^2 \omega_{c_i c_i}), \tag{3.35}$$

The final goal of method is is to maximize the probability that graph $\mathcal{G}$ is created from the partition c with respect to parameters ω_{rs}. Karrer and Newman proposed to solve this optimization problem using a two-step procedure. Firstly, the maximum likelihood values $\hat{\omega}_{rs}$ of the model parameters are calculated in function of the partition parameters c. Then, the most likely community structure corresponding to the partition $\hat{c}$ is defined by maximizing the log-likelihood $\mathcal{L}(G|c, \hat{\omega})$ derived from the above probability. The implementation of the method consists of initializing a partition where vertices are assigned randomly into k communities. Then, vertices are swapped between communities in a similar way of the presented Kernghan-Lin's optimization process (Kernighan and Lin, 1970) in order to acquire an optimal partition. The complexity of the method depends principally on the complexity of the sampling process to create different partition candidates.

Lancichinetti *et al.*'s method

Lancichinetti *et al.* proposed a polyvalent method that can handle multiple type of graphs and which is also claimed to be able to distinguish communities from *pseudo-communities* (Lancichinetti et al., 2011). Reasoning that densely connected groups

of vertices in a graph could be a result of random fluctuations and they can not be considered as non-trivial structures of the graph, the authors then suggested to optimize locally a statistical significance function of presented community structure with respect a global null model. This significance function is equivalent with a fitness measure to evaluate the quality of communities.

The method, also known as *OSLOM* for *Order Statistics Local Optimization Method* makes use of the significance level that a community receives a new vertex with respect to a redistribution of degree outside of it according to a configuration model. Specifically, for community C and vertex i found outside of C in graph $\mathcal{G}$, the probability $P(k_i^{in}|i, C, \mathcal{G})$ that i has k_i^{in} neighbors in C expresses the likelihood that there is a topological relation between i and C. So it vertex i has more edges with the vertices of community C than one would expect, we could consider to include i in C as the relation is unexpectedly strong. The method calculates the cumulative probability $r(k_i^{in})$ of vertex i having the number of internal connections with C equal or larger than k_i^{in}:

$$r(k_i^{in}) = \sum_{j=k_i^{in}}^{k_i} p(j|i, C, \mathcal{G}). \tag{3.36}$$

If the cumulative distribution of the smallest r value is smaller than a given tolerance, the inclusion is considered to be significant, and the corresponding vertex is added to the community. In other way the second smallest r is checked and so on. The algorithm begins with an initial partition, in order to speed up, it can use the result of a fast algorithm. Then it looks for significant communities until no vertex can be incorporated into any community. The mechanism of the method makes open straightforward possibilities to assume overlapping and hierarchical clustering, also it is possible that a vertex is homeless, which means it does not belong to any community. The time complexity of the methods depends on the community structure of the network and is estimated at $\mathcal{O}(n^2)$.

3.2.7 Some other methods

Reichardt *et al.* method using Potts model

Reichardt and Bornholdt demonstrated that the problem of community detection can be reformulated as the problem of finding the ground state of a spin glass model (Reichardt and Bornholdt, 2006). Although this method does not try to optimize the modularity function of graph partitions, there is a close relation between the method's objective function and the modularity function.

At the initial phase of the method, each vertex is assigned by a Potts spin label σ_i indicating the state corresponding to the community that it belongs. The states of vertices are then gradually updated according to a basic principle that vertices belonging to the same classes should be connected and have the same spin state whereas vertices of different classes should be disconnected and have different states. The different of this approach of finding communities from the assumption of modularity is that instead of favoring only intra-community edges and penalizing inter-community edges, the method also favors inter-community non edges and penalizes intra-community non-edges. Consequently, the objective of the method is to optimize the energy contributed by edges and non-edges connecting different spin states. The spin glass energy of a configuration is expressed by the Hamiltonian function:

$$\mathcal{H}(\{\sigma\}) = -\sum_{i<j} J_{ij}\delta(\sigma_i, \sigma_j) = -\sum_{i<j} J(A_{ij} - \gamma p_{ij})\delta(\sigma_i, \sigma_j), \tag{3.37}$$

where $J > 0$ is the coupling strength normally chose at 1. J_{ij} expresses the coupling strength between spins that are both ferromagnetic and anti-ferromagnetic, making an amount of contribution of $(J - J\gamma p_{ij}) > 0$ for an intra-edge and $-J\gamma p_{ij} < 0$ for an intra non-edge. Similarly to the modularity function, p_{ij} represents the expected number of edges connecting i and j in a null model. The flexibility of the method relies on the *tunning parameter* γ, which can be chosen arbitrarily, regulating the inclusion level of the null model into the final quality function. When $\gamma = 1$, the function becomes the traditional modularity function. Following the Hamiltonian objective function, the method makes use of the simulated annealing process (Kirkpatrick, Gelatt, and Vecchi, 1983) to estimate the configuration that has the minimal energy corresponding to the community structure outcome. The theoretical time complexity of the method is estimated at $\mathcal{O}(n^{3.2})$ for sparse graphs.

Raghavan *et al.*'s method based on label propagation

The method (Raghavan, Albert, and Kumara, 2007), proposed to detect communities in large-scale networks, possesses a quite different approach to all the above mentioned methods. Such that it uses solely graph topology in order to identify community structure without having to predefine an objective function for an optimization process nor to assume some prior information about the community structure. Even though, the method shows some desirable qualities such as parameter-free, easy for implementation and fast calculation.

Closely related to message passing paradigms or epidemic spreading, the principled idea of the label propagation method is based on the concept that vertices should probably belong to the community of most of their neighbors. Following this concept, the authors propose a *Label Propagation Algorithm (LPA)* in which each vertex is initiated with a different label denoting the community to which they belong. Then, in each iterative step, each vertex determines its new label regarding to the label that a maximum number of its neighbors have and the process continues until the vertices' labels are stabilized. The order in which vertices are examined are shuffled randomly after each iteration. In this way, labels are first located in the regions where they are initiated, then are propagated across the graph. Groups with densely connected vertices will quickly reach a consensus on a unique label and the algorithm stops when every vertex has the label that the maximum number of their neighbors have. The labels in the final step indicate the outcome community structure of the graph identified by the label propagation process. In terms complexity, this method is one of the most performing as it finishes in linear time $\mathcal{O}(m)$.

Xie-Szymanski's method using speaker-listener label propagation

The label propagation method presented above receives high attention thanks to its simplicity, high performance in terms of calculation time as well as leaving room improvement. Although not being the first who suggests a modified version of LPA, the version of Xie *et al.* is probably the most mentioned in the literature.

The new method introduce a new label update strategy that could reduce around 5 to 10 times the execution time with respect to the traditional *LPA* method (Xie and Szymanski, 2011). The time reduced is thanks to a new indicator stocked in each vertex beside label to represent the sensibility to label changes. A vertex is in fact can

be considered as passive interior, passive boundary or active boundary corresponding to interior vertex, boundary vertex but not subject to label change and boundary vertex whose label could be changed respectively. By this way, only a fraction of vertices are considered when the community membership of the others are well determined. Secondly, the author introduce a new paradigm to generalize the label propagation process. The method considers a vertex that is subject to a label change is a *listener* and their neighbors are *speakers* (Xie and Szymanski, 2012). The method called Speaker-listener Label Propagation Algorithm (*SLPA*) also known as $GANXis$ creates a memory for each vertex to contain the information that they receive. Similarity to the previous method, in each step, the listener incorporates solely the most common among the labels that it receives from the speakers. However, when a vertex transmits signals to its neighbors as speaker, a label is chosen randomly from the memory with a probability proportional to its frequency. This flexibility allows vertices to contain more information about their affiliates to communities and reduces unbalances of boundary vertices. The process are repeated after a certain number of iterations and the labels having the highest frequency in each vertex are used to assign community structure. Moreover, the approach of this method makes it straightforward to determined communities under an overlapping assumption. Again, the time complexity of the method is in linear in function of graph size $\mathcal{O}(m)$.

De Meo et al.'s method using a hybrid local-global approach

Two principle mainstreams presented in the previous sections to detect community structure in network consist in:

- Optimizing the modularity function reflecting the structural difference of a graph with an expected null model's configuration,
- Employing dynamic processes on a graph in an theoretical approach to estimate modular structure.

De Meo *et al.* propose a method called *CONCLUDE* (for COmplex Network CLUster DEtection) combining the two previous approaches to exploit community structure in networks, according to which the authors believe to be able to leverage local and global information of graphs in the detection process (Meo et al., 2014). Specifically, the proposed method uses random and non-backtracking walks of finite length to define a new proximity function between every pair of vertices in the graph of interest. This approach can be considered to be in the same family with the one used by Pons *et al.*'s method introduced in Section 3.2.5 representing the exploitation of global information described by the authors. However, the authors suggest to calculate the distance between two vertices i and j differently:

$$d_{ij} = 1 - \left(\sum_{k \in \mathcal{V}} \frac{[L^{\kappa}(e_{ik}) - L^{\kappa}(e_{jk})]^2}{d_k} \right)^{\frac{1}{2}}, \tag{3.38}$$

where $L^{\kappa}(e_{ik})$ is a measure of edge centrality of edge $e(i,k)$ estimating its role in a message transmitting paradigm to spread information, d_k is the degree of vertex k.

After distances of vertices has been defined, De Meo *et al.* reuse the multi-level local modularity optimization strategy published by Blondel *et al.* (presented in Section 3.2.3). However, instead of revising direct neighbors of each vertex in each each loop of the swapping step to improve the modularity, the defined distances are used to split and merge verices with communities. However, the objective function stays

the modularity in the original version. In terms of time complexity, the method requires only $\mathcal{O}(m+n)$ time.

3.3 A summary of presented community detection methods

3.3.1 Edge removal

Edge betweenness (*GN*) of (Newman and Girvan, 2004) detects communities by removing edges progressively according to their betweenness centrality scores. This method is based on the intuition that dense zones in a graph are loosely connected by a few edges that contribute a high inclusion in the shortest paths between every pair of nodes. Removing these edges would reveals densely connected communities.
Edge clustering coefficient (*RCCLP*) of (Radicchi et al., 2004) suggests to replace the edge betweenness centrality of Girvan-Newman's method by edge clustering coefficient, which requires less computation time and hence reduces the algorithm complexity. In this thesis, we analyze two configurations of this method corresponding to triangular ($g = 3$ denoted by *RCCLP-3*) and quadrangular ($g = 4$ denoted by *RCCLP-4*) versions.

3.3.2 Modularity optimization

Greedy optimization (*CNM*) of (Clauset, Newman, and Moore, 2004) greedily maximizes the modularity function Q by aggregating iteratively connected communities which induce a maximum increase or smallest decrease in modularity ΔQ.
Louvain method of (Blondel et al., 2008) adopts two-step agglomerative process similar to that of the greedy optimization method. However, in each iteration of the first step, it allows nodes to move between communities until no additional gain in modularity can be obtained due to local switch. Then, a new graph whose vertices are the communities resulting from the first step is build and the process is repeated on the new graph to reduce computation time.
Spectral method (*SN*) of (Newman, 2006) identifies community structure by finding leading eigenvectors corresponding to largest eigenvalues of a modularity matrix. In this method, the problem of modularity optimization is *translated* to the problem of vector partitioning of modularity matrix.

3.3.3 Dynamic process

Walktrap of (Pons and Latapy, 2005) defines a pairwise *dynamic distance* between nodes of a graph and then applies traditional hierarchical clustering to detect community structure. The distance is formulated using the transition probability of a random walker based on the concept that nodes belonging to the same community tend to *"see"* other nodes in the same way.
Infomod of (Rosvall and Bergstrom, 2007) uses an information theoretic model where a *signaler* try to send the structure of a network over a limited capacity transmission channel to a *receiver*. The network must be encoded in community structure in a way that minimizes the transfered information and the information loss tat the side of receiver.
Infomap of (Rosvall, Axelsson, and Bergstrom, 2009) represents networks by a two-level structure description. Analogically, each node in a network is encrypted by

a unique codeword composed by two parts: a prefix representing the community to which it belongs and a suffix representing the local code. Detecting community structure becomes equivalent to searching the coding rule to minimize the average code length describing random walks on the network.

3.3.4 Statistical inference

Stochastic Block Model (*SBM*) of (Riolo et al., 2017) uses a Monte Carlo sampling scheme to maximize a Bayesian posterior probability distribution over possible divisions of the network into communities. This probability implies an expected network model to be fitted from the observed network data. In this block model variant, the authors employ a new prior on the number of communities based on a queueing-type mechanism to calculate posterior probability. We analyze in the following sections both traditional *SBM* and *degree-corrected* version *DCSBM* (Karrer and Newman, 2011), which is proved to perform better in practice.
Order statistics local optimization (*Oslom*) of (Lancichinetti et al., 2011) measures the statistical significance of a community by calculating the probability of finding a similar one in a null model. Following this concept, nodes are gradually aggregated into communities to find significant communities. Then nodes are considered to be swapped between communities in order to increase significance level.

3.3.5 Other methods

Spin glass model (*RB*) of (Reichardt and Bornholdt, 2006) finds communities by fitting the ground state of a spin glass model. Instead of favoring only intra-community edges and penalizing inter-community edges like the traditional modularity, this model also favors inter-community non edges and penalizes intra-community non-edges.
Label propagation (*LPA*) of (Raghavan, Albert, and Kumara, 2007) exploits the topology of networks to infer community structure. It is closely related to the context of message passing paradigms or epidemic spreading. The principled idea of this method is based on the concept that nodes should belong to the community of most of their neighbors. Hence, they gradually update their memberships according to their incident nodes.
Speaker-listener label propagation (*SLPA*) - of Xie and Szymanski (Xie and Szymanski, 2012) modifies the propagation mechanism above by a new label update strategy. Also, instead of keeping only hard membership information, each node is equipped by a memory to contain the labels that it receives. Then, in the update phase, nodes transmit the membership to their neighbors according to the membership frequency in the memories.
Mixing global and local information (*Conclude*) of (Meo et al., 2014) combines a dynamic distance with a modularity optimization process to identify community structure. Firstly, the authors define a new pairwise proximity function using random and non-backtracking walks of finite length to determine distances between vertices. Then, the muti-level modularity optimization strategy of *Louvain* method (Blondel et al., 2008) is employed in combining with the defined distance to find community structure.

Table 3.1 summaries the methods presented previously grouped by different approaches. Since community detection is getting more and more attention in the network science community, there is a huge volume of work that has been published

in the recent years to evaluate different methods including both theoretical and empirical approaches. However, there is not any quantitative definition of community that is explicitly implemented inside algorithms, therefore it is challenging to distinguish the topological differences of community structures using different methods, even when the associated concepts are quite theoretically discernible. Additionally, it is still not clear yet whether a proximity in the assumption of community concept will engender a structural similarity of communities that could be detected. Our comparative analysis in the next chapters will try to address these questions in more details.

Approach	**Reference**	**Label**	**Order**
Edge removal	(Girvan and Newman, 2002)	GN	$\mathcal{O}(nm^2)$
	(Radicchi et al., 2004)	RCCLP	$\mathcal{O}(m^4/n^2)$
Modularity optimization	(Clauset, Newman, and Moore, 2004)	CNM	$\mathcal{O}(m\log^2(n))$
	(Blondel et al., 2008)	Louvain	$\mathcal{O}(n\log(n))$
	(Newman, 2006)	SN	$\mathcal{O}(nm\log(n))$
Dynamic process	(Pons and Latapy, 2005)	Walktrap	$\mathcal{O}(n)$
	(Rosvall and Bergstrom, 2007)	Infomod	NA
	(Rosvall, Axelsson, and Bergstrom, 2009)	Infomap	$\mathcal{O}(m)$
Statistical inference	(Lancichinetti et al., 2011)	Oslom	$\mathcal{O}(n^2)$
	(Karrer and Newman, 2011)	(DC)SBM	Parametric
Other methods	(Reichardt and Bornholdt, 2006)	RB	$\mathcal{O}(n^2log(n))$
	(Raghavan, Albert, and Kumara, 2007)	LPA	$\mathcal{O}(m)$
	(Xie and Szymanski, 2012)	SLPA	$\mathcal{O}(m)$
	(Meo et al., 2014)	Conclude	$\mathcal{O}(n+m)$

TABLE 3.1: A summary of presented community detection methods. The first column group methods into theoretical concepts, the second column shows the original reference, the third column indicate the abbreviation of the methods that will be used in following analysis illustration and the final column shows estimated time complexity with $m \approx n$ for sparse graphs.

Chapter 4

Evaluating community structure

We presented in the previous chapters many fundamental concepts related to general properties and characteristic of real-world networks that are exploited to discover community structure as well as many commonly used discovery methods in the literature. However, community evaluation has not been discussed apart from the modularity function, which is as the same time used as a common objective function for many discovery methods. In this chapter, we will focus on a descriptive approach to evaluate structural quality of meta-data communities.

4.1 Community structure evaluation

There are actually two principled ways that are normally used to evaluate the efficiency of any community detection method: using *validation metrics*[1] to compare ground-truth information with discovered community structures, or using structural *goodness metrics* to deduce characteristics of detected communities[2]. In this chapter, we will first focus on the evaluation of community structure using topological metrics which is a subclass of goodness metrics. The other type of goodness metrics based on network modeling, such as the modularity (Appendix A.1), as well as community evaluation using validation metrics will be investigated in more details in Chapter 6.

Remind that we are not trying to deliver an exhaustive list of metrics to evaluate communities and algorithms. Many meticulous surveys and reviews that can be found in the literature do a much better job, for instance some of them can be found in Table 4.1. These surveys encompass a huge variety of community qualities, characteristics and metrics that one could expect to analyze in some specific cases. Instead, we are interested in a shortlist of commonly studied measures in order to perform profound analysis on detectable communities and to infer their characteristics. These analysis also justify our later choices in the comparative analysis of various community detection methods, which is the main content of the next chapter.

4.1.1 Community using topological metrics

Many metrics have been invented to quantify the quality of community structure. In practice, most of them are constructed with an intuition to favor groups of vertices

[1] Sometimes called *similarity metric*

[2] In fact, there is an hybrid approach that tries to characterize communities by combining the two approaches. However this approach is very context-sensitive and can not be applied in a generic manner. Therefore, methods belonging to this approach will not be mentioned in this work

Reference	Main content
(Chakraborty et al., 2017)	Survey of different state-of-the art metrics for community analysis.
(Labatut and Orman, 2017)	Interpretable metrics for charactering community structure.
(Khan and Niazi, 2017)	A survey of surveys on community detection.
(Fortunato and Hric, 2016)	Community validation approaches using benchmarks, meta-data and real world networks.
(Hric, Darst, and Fortunato, 2014)	Comparing structural communities and meta-data in networks
(Van Laarhoven and Marchiori, 2014)	Some requirements of property for community quality functions.
(Yang and Leskovec, 2013)	Evaluating various qualities of community structure based on meta-data information.
(Malliaros and Vazirgiannis, 2013)	Community detection and metrics for quality evaluation, especially in directed networks
(Orman, Labatut, and Cherifi, 2012)	Evaluating community structure based on a topological approach.
(Vinh, Epps, and Bailey, 2010)	Some adjusted information theoretic measures for comparing clustering partitions.
(Mislove et al., 2007)	Statistical measurement demonstrating structural characteristic of social networks

TABLE 4.1: A summary of some reviews and studies on community structure measurements, metrics, detection methods and analysis.

with many connections between the members and few connections from the members to the rest of the graph. According to specific cases, the mathematical formulation of this intuition could be different. Some inceptive efforts have been given to regularize this kind of community quality functions. For instances, (Van Laarhoven and Marchiori, 2014) suggests to impose six properties on quality metrics including: permutation invariance, scale invariance, richness, monotonicity, locality and continuity. We present in this section some well-known structural quality functions which are widely used in the literature. They are later used in our empirical analyses to characterize interaction patterns between nodes in community structure of networks.

Firstly, we remind some notations that will be used to describe structural characteristic of communities. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of $n = |\mathcal{V}|$ vertices and $m = |\mathcal{E}|$ edges can be represented by an associated adjacency matrix A. Given a community C of n_C vertices being a subgraph of $\mathcal{G}$ in an arbitrary partition P, a function $f(C)$ or $f(P)$ quantifies a structural goodness feature of community C or the whole partition P according to a particular expectation of community structure. Let m_c be the number of edges inside community C, $m_C = |(i,j) \in \mathcal{E} : i \in C, j \in C|$, l_C be the number of edges that connect C to other vertices outside of C, $l_C = |(i,j) \in \mathcal{E} : i \in C, j \notin C|$. Any vertex i belonging to community C has an *internal* degree k_{iC}^{int} and an external degree k_{iC}^{ext} satisfying $k_{iC}^{int} + k_{iC}^{ext} = k_i$, where k_i is the total degree of vertex i. The internal and external degree can be expressed via the adjacency matrix A as: $k_{iC}^{int} = \sum_{j \in C} A_{ij}$ and $k_{iC}^{ext} = \sum_{j \notin C} A_{ij}$. If vertex i in community C has $k_{iC}^{ext} > 0$ and

$k_{iC}^{int} \geq 0$, i is called *boundary vertex* since i has neighbor(s) outside of C. Otherwise, if $k_{iC}^{ext} = 0$ and $k_{iC}^{ext} > 0$, i is called *internal vertex* which only has connections with other vertex in the same community. We formulate some structural goodness functions that are classified in the following groups (Yang and Leskovec, 2013):

Internal connectedness measures

- *Average internal degree*: discloses the absolute average number of degree inside the community of interest. It is desirable that communities have a high number of internal degree instead of external degree. Since the number of total degree in a graph is fixed for a given graph, maximizing internal degree also means minimizing the number of inter-community edges. The average internal degree is calculated as follows:

$$f(C) = \frac{\sum_{i \in C} k_{iC}^{int}}{n_C}. \tag{4.1}$$

- *Internal density*: measures the edge density of a community. It is considered as one of the major structural quality since it is expected that there must be much more edges inside than edges connecting the community with the rest of the graph.

$$f(C) = \frac{m_C}{n_C(n_C - 1)}. \tag{4.2}$$

- *Scaled density*: in practice, edge density shows some weaknesses in evaluating real world networks since the number of edges often increases linearly with respect to the size of the community, but the number of possible edges increases quadratically. Therefore, edge density often favors small communities over large communities and is not directly applicable when there is a high variation of community sizes. Scaled density is a kind of normalized edge density which is defined as n_C times the density of the community (Lancichinetti et al., 2010), (Labatut and Orman, 2017):

$$f(C) = \frac{m_C}{(n_C - 1)}. \tag{4.3}$$

  It is easy to see that the concept of scaled density is very close to the average internal degree presented in Equation (4.1). Actually, for a large community, $n_C - 1 \approx n_C$ and the scaled density approaches the average internal degree (multiplying by 0.5).

- *Compactness*: implies that a good community should be at the same time highly populated by edges and vertices must be close from one to another, i.e it must have a high edge density and a small diameter (Creusefond, Largillier, and Peyronnet, 2016). The compactness is calculated as follows:

$$f(C) = \frac{m_C}{d_C}, \tag{4.4}$$

  where d_C is the diameter of community C being the geodesic distance between the two farest vertices of C.

- *Fraction over median degree (FOMD)*: compares the internal degree of vertices in a community with the median degree d_m of the graph to which the community

belongs. The *FOMD* is defined as the fraction of vertices in C that have internal degree higher than d_m. It implies a requirement that a good community must have many vertices relatively more connected to than the majority of vertices in the graph. It is calculated as follows:

$$f(C) = \frac{|i : i \in C, |\{(i,j) : j \in C\}| > d_m|}{n_C}. \tag{4.5}$$

- *Triangle participation (TPR)*: calculates the fraction of vertices in a community that participate in at least one triadic closure with other vertices of the same community. This metric reflects a characteristic usually observed in real world communities where direct neighbors of a node are likely to be connected. It is computed as:

$$f(C) = \frac{|i : i \in C, \{j,k \in C, (i,j) \in \mathcal{E}, (i,k) \in \mathcal{E}, (j,k) \in \mathcal{E}\} \neq \emptyset|}{n_C}. \tag{4.6}$$

- *Clustering coefficient (CCF)*: is a well-known metric to measure the *transitivity* of community whose local version (at node level) has been mentioned in Section 2.2.2. The global version is calculated on the level of graph. The concept is very close to that of the TPR metric. However, instead of measuring the actual fraction of vertices participating in triangular connections inside a community, it quantifies the fraction of triangles over the total number of such pattern that could be established from the set of vertices of the community. Hence, the clustering coefficient penalizes more heavily the expansion of a community by getting weakly and locally connected vertices. Many variants of the clustering coefficient exist, in the following analysis, we compute this metrics as follows:

$$f(C) = \frac{\sum_{i,j,k \in C} A_{ij}A_{jk}A_{ki}}{\sum_{i,j \in C} A_{ij}A_{jk}} = \frac{\text{6 x Number of triangle}}{\text{Number of paths of length two}} \tag{4.7}$$

- *Hub dominance*: reflects the notion of *hub*, *authority* or *influencer* in a network or a community. Internal edges of a community can be distributed in various ways around its vertices, either concentrating around a few numbers of highly centralized ones or uniformly divided into every vertex. The hub dominance metric is designed to identify the level of central organization around well connected nodes. The higher the score, the more likely the community of interest has a hub-like structure. Again, there are many way to define the notion of hub dominance, we use the following presented by (Lancichinetti et al., 2010):

$$f(C) = \frac{max_{i \in C} k_{iC}^{int}}{n_C - 1} \tag{4.8}$$

External connectedness measures

- *Expansion*: computes the number of edges per vertex that point outside the community. It reveals accessibility from one community to the others in a network. The higher the expansion score, the more it is in contact with the rest of network. In a common sense, a good community should be relatively isolated and hence has a low expansion score, which is calculated as:

$$f(C) = \frac{l_C}{n_C} \tag{4.9}$$

- *Cut ratio*: measures the fraction of existing edges between a community and the rest of the network to which it belongs over the total all possible of such edges. The concept is similar to the internal density, but it characterizes the external density. It is calculated as follows:

$$f(C) = \frac{l_C}{n_C(n - n_C)} \tag{4.10}$$

Hybrid measures

Many structural metrics combine internal and external connectivity factors to quantify the quality of communities, i.e. they include both internal degree and external degree in their calculation of goodness score. Some of commonly used are listed in the following:

- *Conductance*: measures the fraction of total edge volume that point outside a community C.

$$f(C) = \frac{l_C}{2m_C + l_C} \tag{4.11}$$

- *Embeddedness*: is defined as the ratio between the internal degree and the total degree of a community. Actually it is the complement to one of conductance:

$$f(C) = \frac{2m_C}{2m_C + l_C} \tag{4.12}$$

- *Separability*: measures the ratio between internal connections and external connections of nodes belonging to a community. This feature is based on the concept that a good community could be well separated by a rupture in edges distribution. This feature is quantified by the following function:

$$f(C) = \frac{m_C}{l_C} \tag{4.13}$$

- *Maximum ODF (Out Degree Fraction)*: is the maximum of the fraction of edges connecting a vertex inside C to other vertices outside. A good community is expected to have a low maximum value of out degree fraction.

$$f(C) = \max_{i \in C} \frac{k_{iC}^{ext}}{k_i} \tag{4.14}$$

- *Average ODF* is defined similarly to the maximum ODF metrics, but instead of taking the maximum value, the average ODF is taken:

$$f(C) = \frac{1}{n_C} \sum_{i \in C} \frac{k_{iC}^{ext}}{k_i} \tag{4.15}$$

- *Deviation ODF*: is proposed to measure the variation of external connectivity of vertices in a community to external vertices. The idea of this metric is to investigate how inter-community edges are distribute among vertices of a community. The difference in the distribution of external edges around boundary vertices characterizes the interaction of the community to the rest of the network. It is calculated as follows:

$$f(C) = \frac{1}{(n_C - 1)^{\frac{1}{2}}} \left(\sum_{i \in C} \frac{k_{iC}^{ext}}{k_i} - \frac{1}{n_C} \sum_{i \in C} \frac{k_{iC}^{ext}}{k_i} \right)^{\frac{1}{2}} \tag{4.16}$$

  We demonstrate in the next section that a combination of different structural goodness functions could reveals interesting patterns in community structure and could be used to evaluate ground truth communities as well as communities discovered by any detection method. Such combinations shed light on the community structure in real world networks.

4.2 Meta-data structure in real-world networks

Evaluating a network partition by comparing it with a reference provides useful information about the global fitness with expected structures. However, when the reference community structure (ground-truth) is not well understood or chosen arbitrary according to the availability of meta-data, this approach could result misleading interpretation. In fact, although meta-data is commonly assigned as ground-truth in order to justify the performance of community detection methods, its is not always a good choice since they correspond to different aspects of networks. While meta-data identifies different information about nodes and edges of a network, ground-truth on the other hand is the expected structural community structure charactering the interaction between nodes. Recently, Peel *et al.* demonstrate clear evidences about the uncertainty relationship between ground-truth and meta-data, which arise different possible scenarios to be considered when a method fails to find good division correlating to meta-data in a network (Peel, Larremore, and Clauset, 2017). Specifically, it is important to understand the relation between meta-data and network structure in order to appraise the appropriateness of using meta-data as ground-truth. In this section, we introduce a novel method presented in (Dao, Bothorel, and Lenca, 2017b) that allows one to expose structural information of communities in a network partition in a comprehensive way. This method, although simple, sheds light on the structure of meta-data affiliation groups in many real world networks as well as provides a useful tool to evaluate communities detected by different community detection methods.

4.2.1 Community anatomy via Out Degree Fraction

The idea behind quality goodness metrics is that given a partition, they indicate how the component subgraphs fit their concepts of community. We present a methodology that assist to analyze communities in networks based on the study of Out Degree Fraction (ODF) (presented in Section 4.1.1) of vertices in communities of a partition.

Specifically, when evaluating community structure of a partition, it is very important to know how component communities interact with each other. Since ODF-based measures quantify the internal and external participation of vertices within

communities, they can be used to deduce whether a vertex is a boundary or an internal member of its community as well as the interaction of different communities within a partition. Therefore, a statistical observation of ODF values of vertices in a community can help classify communities into several structural archetypes.

We employ the average and the standard deviation of fraction of out degrees, denoted by *meanODF* and *sdODF*, characterizing the amount of external interaction and the distribution of external edges around boundary vertices respectively. Remind that *meanODF* and *sdODF* scores of community C can be computed from Equation (4.15), (4.16) as:

$$meanODF(C) = \frac{1}{n_C} \sum_{i \in C} \frac{k_{iC}^{ext}}{k_i} \tag{4.17}$$

$$sdODF(C) = \frac{1}{(n_C - 1)^{\frac{1}{2}}} \left(\sum_{i \in C} \frac{k_{iC}^{ext}}{k_i} - \frac{1}{n_C} \sum_{i \in C} \frac{k_{iC}^{ext}}{k_i} \right)^{\frac{1}{2}} \tag{4.18}$$

A low *meanODF* value implies that vertices of the community under consideration connect mostly with other vertices inside it while a high *meanODF* means that vertices connect preferably to vertices in other communities rather than to the ones in its own. We could refer low *meanODF* and high *meanODF* characteristics to assortative and disassortative structure respectively. A medium value of *meanODF* in this case signifies a hybrid structure of the community. Different classes characterizing community external interactions in function of *meanODF* and *sdODF* scores are illustrated in Figure 4.1. Such that, for a given decomposition of a graph into communities, locating the associated couple of values $(meanODF, sdODF)$ could help to describe the principle components of the community structure.

There are several goodness metrics that could be used to describe community as introduced in Section 4.1. One might wonder why we chose the average and the standard deviation of ODF values of vertices in order to describe a community. In fact, each quality metric has its own meaning and reveals a different aspect of community structure Yang and Leskovec (2013). Because the notion of community also changes according to domains of application and analysis purposes, there is actually no universal metric that can generalize the goodness of communities. Generally, one would expect a clustering where the majority of edges reside between nodes in a same cluster while there are few edges that cross to other clusters. The *meanODF* and *sdODF* are used together in order to describe the distribution edges among nodes in an informative way.

4.2.2 Structural archetypes of communities

Following this line of argumentation, we classify communities into different structural groups based on their node orientations and their structure homogeneities. Community structures in real networks are undeniably much more complex and can not just only be described by *meanODF* and *sdODF* values. However, we are going to show that this simplification helps to give a general view of networks by qualifying community anatomy. Here, we suggest to classify communities into 6 following groups, which are illustrated in Figure 4.1:

- *Conventional communities* (S1 - low *meanODF* and low *sdODF*): This structure corresponds to the traditional definition of community where the majority of

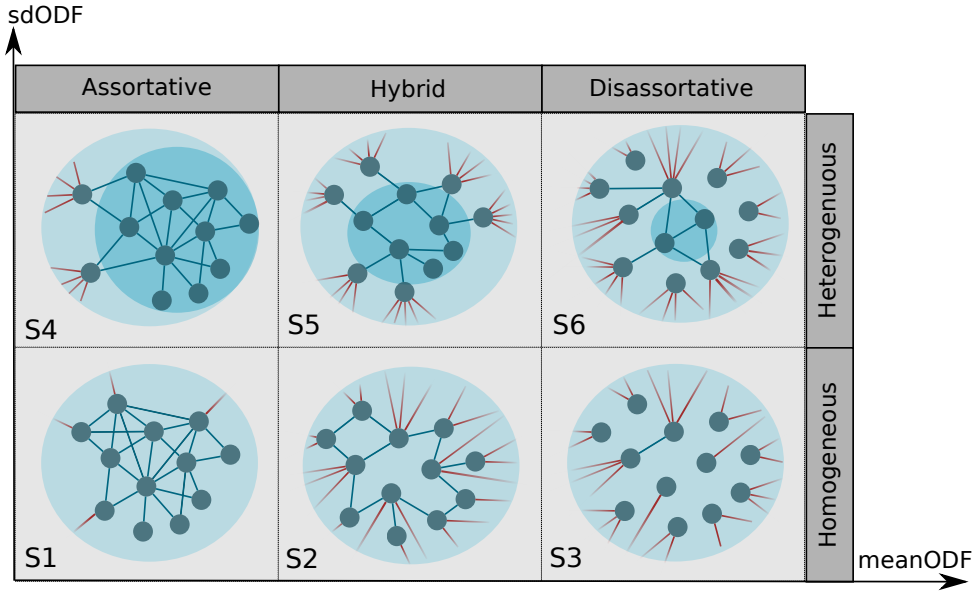


FIGURE 4.1: Six representative structures that can be measured by community's nodes out degree fractions (*meanODF* and *sdODF*). Blue edges represent intra-community connections and red edges (stubs) represent inter-community connections. Dark background zones in *S*4, *S*5, *S*6 structures illustrate a core-periphery arrangement (Dao, Bothorel, and Lenca, 2017b) ©Springer International Publishing AG 2017.

edges locate inside communities. Most of actual community detection methods are based on this notion. In addition, community's out degrees are homogeneously spread over its nodes.

- *Casual communities* (S2 - medium *meanODF* and low *sdODF*): Modular structure is not very clear in this type of community since there is not a clear propensity in node connections inside and outside of communities.
- *Extrovert communities* (S3 - high *meanODF* and low *sdODF*): This structure exposes an explicit disassortative structure where members in a same community are not joined together generally, but rather connect with members of other communities.
- *Full-core communities* (S4 - low *meanODF* and high *sdODF*): This group of communities shows a striking similarity with ones of *S*1 structure since both possess relatively dense inner connections. The only distinction between *S*1 and *S*4 structure is that *S*4 contains a few numbers of *active connector* nodes, which attract most out links. These connectors form a peripheral zone, whereas the other nodes constitute a core as illustrated in Figure 4.1.
- *Half-core communities* (S5 - medium *meanODF* and high *sdODF*): These communities also display core-periphery structure, but there is not anymore a huge dominance of core nodes over periphery nodes like that of in structure *S*4.
- *Seed-core communities* (S6 - high *meanODF* and high *sdODF*): Core-periphery structure in this class of communities is degenerated or even disappeared since out-bound connectors predominate in the whole community. Most nodes connect mainly outside their community with a few exceptions. This structure

Network	N	E	C	S	$\bar{\mu}$	Community nature
Livejournal[a]	4.0M	34.7M	664414	10.79	0.95	User-defined communities
Youtube[a]	1.13M	3.0M	16386	7.89	0.91	User-defined groups
DBLP[a]	0.32M	1.05M	13477	53.41	0.62	Publication venues
Amazon[a]	0.33M	0.93M	75149	30.22	0.58	Product categories

[a] http://snap.stanford.edu/data/

TABLE 4.2: A summary of networks in used with meta-data communities: N number of nodes (in millions), E number of edges (in millions), C number of communities, S average community size, $\bar{\mu}$ average conductance of communities.

have many similarities with $S3$ structure and $S5$ structure and could be considered as a transition state of community evolution between $S3$ and $S5$.

Descriptive evaluation process

The following process helps to decompose network partitions into classes of structurally similar communities. For a given network partition:

1. Compute *meanODF* and *sdODF* values over all communities (cf Section 4.2.1).
2. Present each community by its couple of values (*meanODF*, *sdODF*) to observe the distribution of these quality metrics.
3. Choose thresholds for each quality metric in order to describe desired structure qualities for communities.
4. Identify structure profiles of all communities based on a representative map (cf Figure 4.1) defined from step 3.

Replacing *meanODF* and *sdODF* in step 1 by other couples of quality metrics could also provide further structural information on community structures of networks under consideration. Based on contextual requirements or characteristics of the dataset of interest, thresholds to be chosen in step 3 could be varied and must not necessarily cover the whole range of score. In this latter case, the methodology also serves as a filter to eliminate unqualified communities. In the next section, the presented processes will be applied on well-known real-world network data with meta-data information about vertex affiliation (usually used as ground-truth). This description helps to complement structural aspect of meta-data community and indicate how likely structural goodness could be improved in comparison to meta-data by using community detection techniques on the network.

4.2.3 A descriptive evaluation of meta-data communities

We analyze undirected, unweighted networks with meta-data communities on SNAP dataset (Leskovec and Krevl, 2014). These communities are overlapped and do not cover the whole network, which means one node can belong to no community or can be member of many communities at a same time. The community sizes, the overlap sizes and the community memberships per node in these networks follow a power-law distribution (Yang and Leskovec, 2013).

The dataset used in this analysis is summarized in Table 4.2. Among the four networks, *Livejournal* network is an online blogging community where users declare

their friendships. *Youtube* network represents a social network on Youtube video sharing website. *DBLP* computer science bibliography co-authorship network is constructed in a way that two authors are connected if they published at least one paper together. *Amazon* co-purchased network represents products which are frequently bought together on Amazon website. A description of these networks and measures on their ground-truth communities can be found in Table 4.2.

Our descriptive approach presented above could help to disclose more detailed information about community composition of networks that using only one single conventional quality goodness metric could not. Here, we take the average conductance $\bar{\mu}$ (introduced in Equation (4.11)) as an example. This metric could tell us a global score of community quality, but they can not distinguish many different structures that exist simultaneously in networks. For instance, the average conductance $\bar{\mu}$ shows that there are above 90% of edges in *Livejournal* and *Youtube* that cross communities, meanwhile these numbers are about 60% in *DBLP* and *Amazon*. However, one could not gain more insight into the differences of community structure between *Livenetwork* and *Youtube*, or between *DBLP* and *Amazon*. In fact, by using a two dimensional representation of community structure as introduced in the process above, we disclose that community composition is very discernible in these networks.

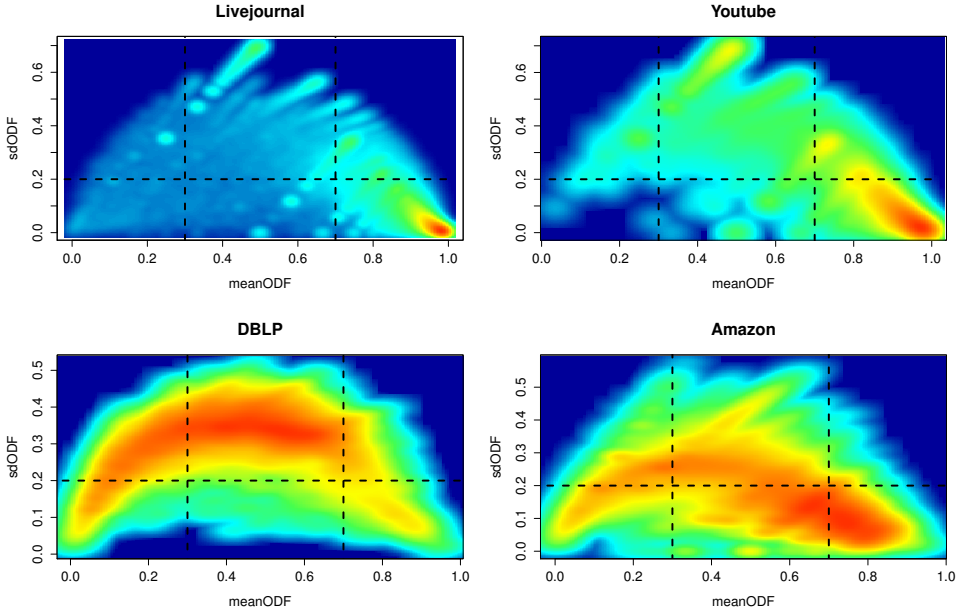


FIGURE 4.2: The distribution of meta-data communities represented by *meanODF*, *sdODF* scores. Each community in a network corresponds to a point in the heatmap. The more communities found in an area, the more the corresponding color moves toward a hot color. Deep blue color corresponds to an absence of community. The dashed lines represent thresholds between the 6 presented structures $S1$ to $S6$.

Figure 4.2 presents the landscape of *meanODF*, *sdODF* values of all ground-truth communities in the 4 networks. Since no discontinuous transition is found on the distribution, we classify arbitrarily these communities into the 6 groups as presented in Section 4.2.1 by choosing thresholds for *meanODF* at 0.3, 0.7 and for *sdODF* at 0.2. The landscape helps us to analyze the composition of ground-truth community

structures in each network. We remind that the density landscapes in Figure 4.2 do not represent the networks themselves, but the community structures in these networks.

Network	S1	S2	S3	S4	S5	S6
Livejournal	0.29	0.74	**90.17**	0.31	3.88	4.61
Youtube	0.08	2.36	**65.36**	1.37	**17.55**	**13.28**
DBLP	6.28	2.07	4.87	**23.44**	**57.86**	5.48
Amazon	8.33	**31.13**	**23.57**	9.13	**26.63**	1.21

TABLE 4.3: The composition of communities in the 4 networks (in percentage). Bold values indicate dominant structure(s) of community class found on each network.

We can see that the structural patterns of communities within 4 networks are totally distinct. Normally, one would expect that ground-truth communities in a network have a quite similar structure, but the density landscapes in Figure 4.2 illustrate a more mixture community composition. While in *Livejournal* and *Youtube* networks, the majority of communities have a similar structure, those in *DBLP* and *Amazon* networks vary in a much more larger range. Table 4.3 describes a global composition of the 4 networks in terms of the 6 basic structural groups ($S1$ to $S6$). We find that $S3$ structure occupies around 90% and 65% of communities in *Livejournal* and *Youtube* networks respectively. This implies the fact that most users in these networks usually have friendships outside their communities rather than inside. In addition, there are many closely-knit members in *Youtube* network, who are not very active outside their communities ($S5$ and $S6$).

In *DBLP* and *Amazon* networks, although there is aways a dominance of some structures, we notice a more equilibrate repartition of communities over the landscapes. In the case of *DBLP*, nearly 60% of publication venues ($S5$) attract a variety type of authors in term of cooperation profile. These communities could represent traditional publication venues which gather at the same time high influence authors and newcomers. Meanwhile, there is about 23.44% publication venues where presented just a few active *eminent* authors. In *Amazon* network, the high presence of $S2$ and $S3$ structures explains that products are more often co-purchased with ones of other categories. Besides, there are also many miscellaneous product categories ($S1$, $S4$, $S5$) which consist of a high portion of products that are mostly complemented by ones in the same categories. Further analysis in natures and functionalities of products need to be conducted in order to understand this commercial network.

Clearly, this descriptive decomposition of community structure exposes additional information that a single quality goodness metric, such as conductance, could not be able to reveal. The issue raised here is that, on the contrary to the common belief that real-world communities in social networks are normally assortative, disassortative communities assigned by vertex labels are not very uncommon in practice (Dao, Bothorel, and Lenca, 2017b). Consequently, community structure deduced from meta-data is not always suitable as ground-truth communities. In our specific case, since it is expected that community methods must identify dense subgraphs, there is even an inverse correlation between expected structure's quality and meta-data community structure's quality. That is to say, meta-data communities usually exhibit poor structural qualities that are normally not expected to be the outcome of a community detection process. Hence, it is likely that using a community detection method will help to discover sub-graphs with better structural qualities. From

that notice, we would like to go further to see how different community detection methods could improve structural qualities. The next section is hence dedicated to a quantification of different methods in improving some popular structural quality notions. We will demonstrate that this quantification of structural quality improvement provides useful information that helps to rank different methods in function of their capacity to identify good communities according to some certain metrics. Consequently, our study could also help to determine suitable community detection methods to be employed in order to optimize a given quality.

4.3 A quantification of community quality improvement

In this section, we are interested in quantifying the performance of some popular community detection methods in their capacity to identify good structural clusters. We rely on the assumption that there exists communities whose structures are better than *"ground-truth"*[3]. To avoid further confusion, we use the word *meta-data* instead of *ground-truth* to describe communities identified in a semantic way in real-world networks (for instance, authors participating to a publication venue, users showing the same interest on a social network, etc.) as well as planted communities in synthetic networks. The main idea of this analysis is to use meta-data communities as a reference in a quantitative ratio as follows:

Suppose that a method M discovers n_M communities in a network, which contains n_0 meta-data communities, we define a goodness ratio which quantifies the improvement of quality feature Q promoted by method M by (Dao, Bothorel, and Lenca, 2017a):

$$R(M,Q) = \frac{\left[\sum_{i=1}^{n_M} g_Q(C_i)\right] / n_M}{\left[\sum_{j=1}^{n_0} g_Q(C_j)\right] / n_0}, \tag{4.19}$$

where g_Q is one of the goodness function representing quality Q presented in section 4.1.1 of community C. This ratio can vary from zero to infinity. $R(M,Q) = 1$ indicates that the method M provides communities that are as good as meta-data communities in terms of quality Q, while $R(M,Q) > 1$ and $R(M,Q) < 1$ implies an enhancement and a degradation respectively. As some quality metrics show general agreements in evaluating structural quality of communities (Yang and Leskovec, 2012), we only choose some representative goodness functions in our quantification measured by Equation (4.19) to demonstrate the effectiveness of community detection methods as follows:

- *Density*: captures the idea that nodes in a community must be well connected. It quantifies the fraction of edges inside C over the total possible edges which could be established in C - Equation (4.2).
- *Compactness*: suggests that good communities should be at the same time dense and easily reachable from nodes to nodes. This quality is calculated by the fraction between number of internal edges and the diameter of community C (Creusefond, Largillier, and Peyronnet, 2015) - Equation (4.4).

[3]In fact, the term *ground-truth* community is sometimes abused in the context of community detection evaluation. When they are groups of nodes attributed by the same label(s), they can be called meta-data communities.

- *Clustering coefficient*: is a well-known metric which is used to evaluate community quality. It is based on the concept that pairs of nodes with common neighbors are more likely to be connected - Equation (4.7).
- *Community modularity*: measures the difference between edges inside C and the expected number of such edges in a random network with the same degree distribution - Equation (A.1).
- *Embeddedness*: reflects how much the direct neighbors of a node belong to its community. It is measured as the ratio of internal degree to the total degree of a community. - Equation (4.12).
- *Separability*: is based on the concept that a good community should be well separated by a rupture in edges distribution. This function measures the ratio between internal connections and external connections of nodes inside a community - Equation (4.13).

4.3.1 Network dataset with meta-data

Graph	N	E	$\widehat{k}$	$\bar{\alpha}$	CCF
zachary[4]	34	78	4.6	-2.2	0.26
football[4]	115	613	10.7	-9.1	0.41
polblog[4]	1222	16714	27.4	-3.7	0.23
youtube[5]	39841	224235	11.3	-2.8	0.06
livejournal[5]	84438	1521988	36.1	-2.4	0.77
dblp[5]	317080	1049866	6.6	-3.3	0.31
amazon[5]	334863	925872	5.5	-3.6	0.21
lfr1	5000	26836	10.7	-3.0	0.19
lfr2	10000	24617	4.9	-3.1	0.31
lfr3	25000	133429	10.7	-3.1	0.02
lfr4	100000	480978	9.6	-2.5	0.18
lfr5	100000	1056963	21.1	-2.5	0.06

TABLE 4.4: A description of dataset with meta-data in use. N - number of nodes, E - number of edges, $\widehat{k}$ - average degree of nodes, $\bar{\alpha}$ - estimated power law exponent of node degree sequence, CCF - clustering coefficient.

In this experiment, we need to collect graphs with meta-data information, which could be constructed from real world networks of synthetic networks. Hence, we reuse graphs presented in Table 4.2. Besides, some synthetic networks with built-in communities from LFR benchmark (Lancichinetti, Fortunato, and Radicchi, 2008) in the second part. These synthetic networks are created in a way that their structural parameters approach those of real-world networks as shown in Table 4.4 with corresponding meta-data communities in Table 4.5. Specifically, node degree in networks follows power-law distributions with exponent coefficients around -2.5. The average degrees are set from 5 to 20 to acquire relatively sparse networks. Besides, in order to obtain a variety of community quality, the *lfr1-lfr5* networks are configured with mixing parameters $\mu \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ which represents the probability that an edge of a node is connected to nodes outside of its community.

[4]http://www-personal.umich.edu/ mejn/netdata/
[5]https://snap.stanford.edu/data/

Graph	C	S	A	$\bar{\beta}$	Meta-data
zachary	2	17.0	1.0	NA	Membership
football	12	9.6	1.0	NA	Team groups
polblog	2	611	1.0	NA	Political alignment
youtube	5000	14.6	1.8	-2.2	Subscription
livejournal	5000	27.8	1.7	-2.8	Membership
amazon	75149	30.2	6.8	-2.1	Product categories
dblp	13477	53.4	2.3	-3.1	Publication venues
lfr1	515	13.6	1.4	-3.1	Planted groups
lfr2	1473	6.8	1.0	-3.3	Planted groups
lfr3	3002	21.7	2.6	-2.1	Planted groups
lfr4	9434	13.1	1.2	-2.5	Planted groups
lfr5	4729	21.2	1.0	-2.6	Planted groups

TABLE 4.5: A description of structural information of meta-data groups. C - number of communities, S - average community size, A - community membership per node, $\bar{\beta}$ - estimated power law exponent of community size distribution

4.3.2 Experimental results

Method	Section	Sep	Emb	Den	Com	CCF	Q
Fast greedy	3.2.3	6.18	1.46	2.79	1.79	0.99	2.71
Louvain	3.2.3	**11.01**	**1.50**	2.68	**6.02**	0.94	**12.67**
Infomap	3.2.5	2.24	1.26	3.34	0.96	0.90	0.75
Walktrap	3.2.5	1.87	1.19	**3.35**	0.78	0.93	0.65
Oslom	3.2.6	1.69	1.10	1.29	1.21	1.05	0.83
Label propagation	3.2.7	2.72	1.40	1.84	1.15	1.11	1.06
Speaker-Listener LPA	3.2.7	5.34	1.39	2.54	1.19	1.03	0.84
Conclude	3.2.7	1.42	1.13	2.52	0.72	**1.33**	0.63
Average ratio		4.06	1.30	2.54	1.73	1.03	2.52

TABLE 4.6: Average goodness score ratios on real-world networks. Ratio between average structural goodness scores of detected communities over those of meta-data communities calculated based on equation (4.19). Sep - *separability*, Emb - *embeddedness*, Den - *density*, Com - *compactness*, CCF - *clustering coefficient*, Q - *cluster modularity*, Avg - *average quality improvement score*. The best method of each quality is written in bold.

We measure all goodness scores of all communities detected by each method and calculate goodness ratios based on Equation (4.19). The average ratios are showed in Table 4.6 and 4.7 for real-world networks and synthetic networks respectively. Each row corresponds to a method and each column corresponds to a goodness metric.

Surprisingly, we observe a significant quality improvement in most methods and goodness metrics, even in synthetic networks where it is widely believed that planted communities are the best decomposition to be found, there are still improvements of many qualities in general. This phenomenon is explainable since LFR benchmark only create communities based on a mixing parameter condition which is not always preferred by all goodness functions. Hence, it is generally possible to get higher goodness scores just by some simple actions such as merging, dividing

Method	Section	Sep	Emb	Den	Com	CCF	Q
Fast greedy	3.2.3	**1.94**	1.22	0.35	**14.97**	0.80	**36.70**
Louvain	3.2.3	1.57	1.21	0.09	7.10	0.73	26.62
Infomap	3.2.5	1.22	1.08	0.95	0.99	0.98	1.00
Walktrap	3.2.5	1.22	1.07	**1.27**	0.79	1.04	0.83
Oslom	3.2.6	1.10	1.04	0.86	1.13	0.97	1.25
Label propagation	3.2.7	1.20	**1.64**	0.88	0.87	0.87	0.79
Speaker-Listener LPA	3.2.7	1.14	1.04	1.04	0.84	0.96	1.03
Conclude	3.2.7	1.00	0.97	1.26	0.76	**1.22**	0.77
Average ratio		1.30	1.16	0.84	3.43	0.95	8.62

TABLE 4.7: Average goodness score ratios on synthetic networks. The abbreviations are reused from Table 4.6.

communities, etc. However, we can see in Table 4.7 that no method can improve all goodness scores of synthetic communities at the same time. Each method will indeed improve some goodness scores while reduce some others in a proportionate manner as can be observed in Table 4.6 and 4.7.

Importantly, it can be seen that the average improvement of goodness scores in real-world networks is generally higher than that of in synthetic networks. This is completely reasonable since meta-data communities in synthetic networks are planted based on many topological conditions while meta-data communities in real-world networks are often chosen by semantic meanings, functional criteria or sometimes in a subjective way. As a consequence, there is obviously less correlation between meta-data communities and structural communities in real-world networks. For this reason, community detection algorithms could ameliorate more remarkably structural goodness scores of real-world meta-data groups.

Density is the only goodness metric that shows a global degradation in synthetic networks. Though, it is totally explainable since the link density of a subgraph C is measured by the ratio between the number of links inside C and the total number maximum of links that could be formed, which is $n_C(n_C - 1)/2$. While link density increases linearly with the size of C in sparse graphs, the number of possible links increases quadratically. So it is clear that *density* favors small communities in general. Since synthetic community sizes follow power-law distributions with high exponent coefficients, there are plenty of tiny communities in synthetic networks. This explains why *density* is not often improved. *Walktrap* is always the method that identified the most dense sub-graphs in two cases (real-world networks and synthetic networks) due to a large number of small communities in comparison to the other methods. Other analyses in Chapter 6 will clarify more on this point.

Since both metrics *separability* and *embeddedness* are both built on the notion that good communities have relatively higher number of internal edges than number of external edges, one can remark that there is a correlation between these two metrics throughout all methods in both cases. *Compactness* favors short diameter communities where nodes are easily accessible from one to another. This conception explains why methods that discover nodes of networks in a local manner such as *Louvain*, *Fast greedy* usually improve significantly this goodness feature. Even though communities detected by these methods are normally very large, the associated diameters are often upper-bounded.

Modularity is the most atypical among the studied metrics as it is improved more

significantly in synthetic networks. Since modularity is designed as an accumulative function, it can be misbehaved by using a ratio between two averaged cluster modularity (i.e the standard modularity is measured as a sum over all communities of a partition). Unsurprisingly, it is not unpredictable to see that modularity optimization-based methods such as *Fast greedy* and *Louvain* enhance remarkably this metric. Our experiments confirm that methods detecting larger communities are likely to obtain higher modularity values, as also shown in the well-known resolution limit problem (Fortunato and Barthelemy, 2006).

4.4 Conclusion

In conclusion, we demonstrate in Section 4.3 a detailed quantification showing how good communities discovered by some popular detection methods in comparison to meta-data communities in terms of six well-known quality functions. Our findings reinforce the result presented in Section 4.2 showing that real-world communities are not structurally dense in most cases. Specifically, many real-world communities, such as user-defined groups in *Livejournal* or *Youtube* as presented in Table 4.2 are disassortative, i.e. there are normally more edges connecting nodes of different groups than edges connecting nodes of the same group (defined by node label). Since the functionality of any community detection method is searching for relatively densely connected sub-graphs, it is very likely that quality scores favouring dense structures could be augmented. Consequently, using disassortative (or unknown) community structure as ground-truth in order to validate a community detection method will probably result in misleading conclusions. Therefore, a pre-analysis of real-world community structure is necessary in case that it is used as an expected outcome. Otherwise, synthetic networks with planted communities should be used in the validation of a community discovery method. Also, it is better to verify the outcome of a method by several quality functions since the improvement of a quality could be a predictive signal of the deterioration of another quality. Among all methods that we analyzed, no one exhibits permanently good results in all cases (quality functions) and inversely, no one exhibits bad results in all cases. That is the reason why determining an expected objective function is a very essential presumption that help to choose well performed methods. Based on this notice, our results presented in Section 4.3.2 provide a global reference demonstrating the performance of some popular community detection methods that could assist network practitioners in deciding eligible methods according to their quality criteria. Finally, the analyses in this chapter can lead us to some brief conclusions:

- Meta-data communities (node attributes) in real-world networks are not always structurally good. Hence using them as ground-truth in the evaluation of community detection algorithms needs to be performed with caution. Further processes to analyze the correlation between meta-data and structure information need to be conducted. (Peel, Larremore, and Clauset, 2017) propose some interesting approaches.

- In many case, some algorithms could identify communities that structurally better than meta-data communities, even in synthetic benchmarking networks (LFR). However, an improvement of some qualities could also imply a diminution in some other qualities. It means that if one knows exactly what kind of structure she or he needs to find in her or his network, it is possible to find a method that can perform better than ground-truth information.

- Some algorithms may perform better in optimizing some certain qualities. For instance, *Louvain* method finds significantly high *modularity*'s structures and *Walktrap* tends to choose small and dense communities, although they both use the same objective function (modularity) at their final step. Hence, looking only at final objective function could lead to undesired outcomes.

Besides, our finding showing that meta-data communities are not structurally good raise a necessity to interpreting and characterizing structural communities identified by different community detection methods. This characterization is important in the sense that it helps to understand structures that can be obtained by using community detection algorithms as well as how can we model community structure. This content will be the main focus of Chapter 5.

Chapter 5

Characterizing community structure

In this chapter, we focus on characterizing structural communities obtained by using different community detection methods in Section 5.1. Specifically, we reintroduce some notable community quality topological metrics that help to characterize community structure in Section 5.1.1. Then, Section 5.1.2 introduces the dataset containing networks of different categories that will be analyzed in this chapter as well as in Chapter 6. Next, Section 5.1.3 demonstrates a pre-analysis of these topological metrics based on structural communities detected on the presented dataset. This pre-analysis provides guidance information to select representative metrics characterizing community structures. We describe in Section 5.1.4 some popular interaction patterns of nodes found in structural communities by using a combination of two representative metrics. In Section 5.1.5 these interaction patterns are matched to well-known network models of the literature. Finally, Section 5.2 illustrates an association between networks of some well-studied categories with the characterized structures and models.

5.1 Structural community characterization

In the previous section, we unravel structural information of real-world communities in some large-scale networks assigned by different node labels. Our analysis show that community detection techniques can identify sub-graphs with substantially improved internal degree fraction, meaning that they detected significant communities. However, we desiderate to go further. It is worth exploring community structures in an in-depth analysis to answer the intriguing question: *"What do structural communities in real world networks look like?"* or, *"Is there any significant difference between structural communities across network categories?"*. Therefore, we are interested in characterizing structural communities in several real world network categories. It is expected that a well understanding of community structures could help network analysts to discern different types of community that could be found on their networks. Furthermore, the insight into community structure may guide for a good conception of detection mechanism or lead to appropriate choices of community detection technique (Dao, Bothorel, and Lenca, 2018a).

We focus on the evaluation of structural communities, which means communities are distinguished by interaction between their nodes through edges but not by contextual information neither network meta-data. One could criticize this approach since it is also possible that real communities in networks are not structurally good but yet well cohesive according to a more *natural* sense of community. This remark

is logical whilst invalid since community detection methods are designed to capture structural organization of networks, not community in a wide sense[1]. Additionally, a generic analysis using only interaction information enables a comparative approach to contrast communities throughout different network categories, which are not allowed by sophisticated approaches using contextual information.

5.1.1 Topological metrics

We propose to use structural metrics to characterize community structure on different networks. Some of them are introduced in Section 4.1.1 as a tool to compute the goodness of communities. In this part, we proceed an analysis on different metrics in order to choose suitable ones that can efficiently distinguish and describe vertex interactions in communities. Since it is not expected that a finite set of structural features could fit every intuition of community and the choice of any set of metrics would be adversarial unless a specific context is clearly defined under a constrained circumstance. We restrict our list of quality metrics of interest in the analysis by applying the following criteria from the highest to the lowest priority:

- Since we are characterizing communities in different types of networks, we are only interested in metrics which delineate communities themselves, not based on a hypothesis that they are created from a network model nor relative relation with the global structure of the network where they are found (such as Cut ratio which depends on the number of vertices of the whole graph; different variants of modularity impacted by null models; or Description Length resulted from a dynamic process on a global scale) even though their efficiency in identifying meaningful community structures has been proven.
- Potential metrics for the analysis must be relatively uncorrelated from one to another throughout a wide range of networks in order to illustrate different aspects of structural characteristics. Some metrics could be very similar in their concepts, hence have high mutual information. Only one representative will be analyzed in this case.
- A metric whose concepts can be represented intuitively and visually in order to describe most distinguishable characteristics is preferable than a metric that reflects statistical ideas which are difficult to be presented by simple topologies. Therefore, in order to characterize community structure, we restrict our analysis on topological metrics.

Following the previous indications and some preliminary analysis to keep the characterization in control, we selected some topological metrics introduced in Section 4.1.1 and remind their topological meanings as follows:

1. Metrics based on internal edge density
    1.1. *Density* (Equation 4.2): captures the idea that nodes in a community must be densely connected wherever possible. It quantifies the fraction of edges inside a community over the total possible edges that could be established.

[1] Actually, they are more commonly used to deduce *"real communities"* or predict missing meta-data information with some assumptions about their correlations.

1.2. *Scaled density* (Equation 4.3): is a kind of normalized density which is defined as the density of the community multiplied by community size. This normalization is usually applied to palliate an issue due to the fact that the number of edges in a sparse network increases linearly with its size, however the number of possible edges increase quadratically. It is believed to reflect better edge density concept in real world networks (Lancichinetti et al., 2010).

2. Metrics based on centralized/hub structure

2.1. *Hub dominance* (Equation 4.8): Internal edges of a community can be distributed in various ways around its nodes, either concentrating around a few numbers of high centralized nodes or uniformly divided into every node. The hub dominance metric is designed to identify the level of central organization around well connected nodes. The higher this metric of a community, the more likely it has a hub-like structure. Hub dominance can be considered as a normalized version of degree centrality introduced in Section 2.2.2.

3. Metrics based on triadic structure

3.1. *Community clustering coefficient* (Equation 4.7): reflects the probability that the adjacent vertices of a vertex are connected. This is a well-known metric which is usually used to evaluate modular structure in networks. It is based on the concept that pairs of nodes with common neighbors are more likely to be connected (Watts and Strogatz, 1998), (Barrat et al., 2004).

3.2. *Triangle partition ratio* (Equation 4.6): measures the fraction of nodes in a community that participate to at least a triadic structure. It is used to measure the quality of communities, as it is expected that in good communities, most of nodes must cohesively connected to each other and establish compacted structure.

4. Metrics based on external connectivity

4.1. *Expansion* (Equation 4.9): measures the number of edges per node that point out side a cluster. It represents the relative out degree of a cluster over its size. The higher the expansion of a community, the stronger the its connection with the rest of the network.

4.2. *Conductance* (Equation 4.11): represents the fraction of degrees of a community that points outside over the total of its degrees. The conductance reveals how much the direct neighbors of a node in the community belong to neighborhood communities. Leskovec *et al.* show that finding a configuration in networks that minimizes the conductance of communities helps to identified good network community profile (Leskovec et al., 2008).

4.3. *Average Out Degree Fraction* (Equation 4.15): indicates the average of out degree fraction of nodes in a community, a low value implies that nodes in the community connect primarily with other nodes inside the community while a high value means that nodes connect preferably to nodes in other communities rather than to the ones in its own.

4.4. *Maximum Out Degree Fraction* (Equation 4.14): reflects the maximum of fraction of edges of a node in a community that connect to the external nodes. This metric helps to quantify the interaction of the most active node of a community with the rest of the network.

We conduct an empirical analysis in order to understand how topological metrics are correlated in practice. Since our objective in this part is to characterize different modular structures that could potentially be identified on real-world networks, we employ community detection techniques as a tool to discover communities. The following analysis is based on a premise that community detection methods allow to discover modular structures of networks. The partitions provided by clustering methods are supposed to yield a low granular structures inside communities. By examining the these structures, we are inspecting how nodes in complex networks constitute communities, how are they connected and is there any difference in the way that they interact in different networks.

We chose a few numbers of methods presented in Section 3.2 whose performances have been proven in the literature in order to examine different structural metrics. The only criterion we take into account is that these methods use different approaches to discover communities. These methods are used to detect communities in a large network dataset, which will later serve our analysis of metrics. Detection methods used in this section are resumed in Table 5.1. While the edge betweenness method (Girvan and Newman, 2002) is based on edge centrality detection in order to break networks into several communities; the Louvain method (Blondel et al., 2008) optimizes local modularity by iteratively folding nodes into meta-nodes; the label propagation (Raghavan, Albert, and Kumara, 2007) determines the community of a node by considering the memberships of its neighbors; and the Infomap method (Rosvall and Bergstrom, 2008) relies on finding a configuration that maximizes the compression of a random walks represented by an encoded binary sequence. Of course one could argue that by using only a few numbers of methods, it is likely that some kind of structures are not well covered in the analysis. Although it is a very pertinent requirement, within this study, the authors find that the utilization of some representative methods could already help to reveal substantially many interesting community structures. Further analysis with other discovery approaches could probably disclose many other interesting patterns.

5.1.2 Categorized network dataset

In this section, we describe some statistical properties of networks that will be included in the following analysis. It is expected that networks in each category are spread in a wide range of structural measures. However, available biological networks that have been published and analyzed widely are relatively small in comparison to the other networks of the other families. Besides, due to the complexity of the analysis process, we limit the domains of interest at 5 categories which are commonly researched and where numerous networks are available. The number of networks considered is 108 which is relatively large in comparison to many studies. Many notable related work where some of these networks are also employed to study community structure could be mentioned for a quick reference: Orman *et al.* use 6 networks to evaluate the structure of communities discovered by several detection techniques (Orman, Labatut, and Cherifi, 2012); Lancichinetti *et al.* use 15 networks to characterize structural communities (Lancichinetti et al., 2010); Hric *et al.* use 16 networks to reveal differences between structural communities and ground

Method	Approach	Reference
Edge betweenness	Edge centrality detection	Girvan and Newman, 2002
Fast greedy	Modularity optimization	Clauset, Newman, and Moore, 2004
Louvain	Multilevel modularity	Blondel et al., 2008
Spectral	Vector partitioning	Newman and Girvan, 2004
Walktrap	Dynamic distance	Pons and Latapy, 2005
Infomap	Information compression	Rosvall and Bergstrom, 2008
Label propagation	Topological closeness	Raghavan, Albert, and Kumara, 2007
Spin glass	Energy model	Reichardt and Bornholdt, 2006

TABLE 5.1: A summary of community detection methods used to study community structure in our analysis. They are used as a tool to identify latent modular structures hiding in a large network dataset.

truth (Hric, Darst, and Fortunato, 2014) ; Leskovec *et al.* use over 100 networks to analyze network community profile (Leskovec et al., 2008) and 230 networks to evaluate the goodness of ground-truth communities in social networks, within this number, 225 samples of the Ning online social networking platform's networks[2] are aggregated (Yang and Leskovec, 2013). Table 5.2 resumes the composition of networks that have been analyzed in this section.

Some notable structural measures of networks in the dataset are illustrated in Figure 5.1. It is noticeable that apart from biological networks which are relatively small, the other classes cover quite a wide range of number of nodes, edges, mean degree, clustering coefficient and edge density. Since real world networks are relatively sparse, the number of edges increase linearly in function of the number of nodes and consequently, the edge densities decrease linearly by the number of nodes (since the number of possible connections increase quadratically by the number of nodes in a community). This sparsity property can be easily noticed from Figure 5.1(a,d). Specifically, the number of edges increases linearly in function of the number of nodes with equivalent rates among different network categories as can be deduced from the gradients of the linear estimates. From Figure 5.1(b), it can be seen that the average degree of the networks in the dataset varies principally between 1 and 100 edges per node except for 2 communication networks. Also, the majority of networks has an average degree of approximately from 10 to 20 connections. In a global point of view, networks in the dataset have a quite strong modular quality since most of them have relatively high clustering coefficient as shown in Figure 5.1(c).

5.1.3 Choosing representative topological metrics

In other to characterize structural communities in different types of networks, we apply various community detection methods on the dataset grouped by network category. Once communities are produced, the topological metrics are used to investigate the structure of detected communities. Since many metrics reflect close

[2]https://www.ning.com/

Category	Size	Nodes	Edges	Notable networks
Biological	7	1860	10763	Protein, yeast
Communication	9	39595	195032	Email, forums
Information	25	38358	159812	Citation, Amazon
Social	37	6888	49666	Facebook, Youtube
Technological	19	18431	48494	Internet, P2P
Miscellaneous	11	4298	49033	Ecology, synthetic
Total*	108	1.99M	9.08M	

TABLE 5.2: A summary of network dataset used in this analysis where **Size** is the number of networks analyzed in each category, **Nodes** and **Edges** indicates the average number of nodes and edges of networks in each category respectively. *The last row shows the total number of networks, nodes and edges in the whole dataset. This dataset is collected from several sources including: http://networkrepository.com (Rossi and Ahmed, 2015), http://konect.uni-koblenz.de (Jerome, 2013), http://snap.stanford.edu (Leskovec and Krevl, 2014)

.

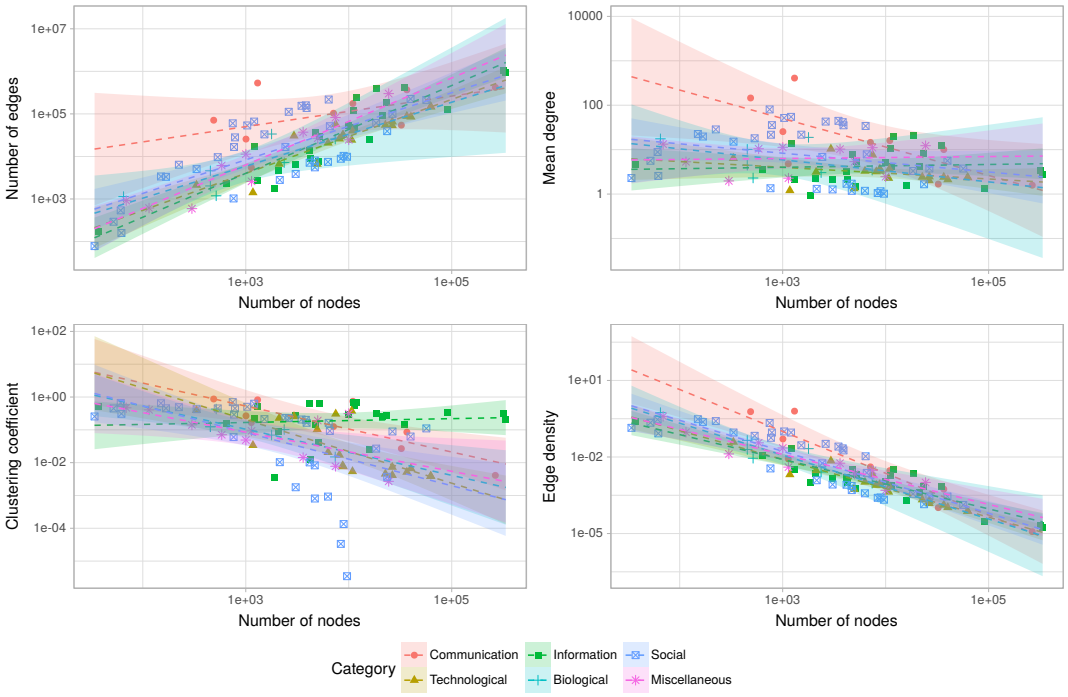


FIGURE 5.1: From left to right, up to bottom, we illustrate structural measures of the 108 networks: (a) Number of edges as a function of the number of nodes, (b) Mean degree $\langle k \rangle$ as a function of the number of nodes, (c) Clustering coefficient in function of the number of nodes, (d) Edge density in function of the number of nodes. The colored backgrounds represent the 95% confidence intervals of the relations estimated from the dataset using a linear regression model for the corresponding variables on each network category.

structural properties, we analyze the correlations between the corresponding qualities on the detected community sets. This analysis allows to select only the most representative structural metrics to delineate community structures.

Figure 5.2 illustrates the correlation matrices of different structural qualities measured on various community sets identified by the set of community detection methods in Table 5.1 over 5 classes of networks and the whole dataset in Table 5.1. Only communities whose sizes are at least 3 nodes are taken into consideration in the figure since many metrics are meaningless for too small communities (which contain one or two nodes). It is important to note that although some statistical metrics are only significant when measuring on large communities, the corresponding correlation matrices for large scale communities resemble globally with those of Figure 5.2. Specifically, a calculation using only large communities of more than 10 nodes gives quite similar and consistent correlation scores. The employment of representative some certain quality metrics can be globally justifiable on the whole range of community size scales.

As we can see in Figure 5.2, there are two groups where metrics are consistently correlated from one to another. The first group includes *maxODF, meanODF* and *conductance* which represent community external connection with very high correlation coefficients (except for *maxODF* and *meanODF* in information networks with a relatively weak relation of 0.51). Besides, the *expansion* metric also belong to this group in technological, information and biological networks with high correlation scores and more loosely in the other types of networks. The second group consists in *TPR* and *CCF* which expose triadic tight-knit structures and are observed with very high correlation scores in every case of network category. The lowest correlation score between *TPR* and *CCF* is reported at 0.81 in information networks and approximately around 0.90 in all the other cases. Without loosing the generality, in our analysis, these 2 groups of metrics could be reduced to two representative metrics representing these two kinds of structural properties.

Hub dominance (*hub_dom*) is the only metric who is quite independent of all metrics in the two previous groups in every network category. The highest absolute correlation score between *hub_dom* with these metrics is 0.42 with *maxODF* in social networks, which is still a relative low correlation. This latter, however, is generally correlated with *density* except for the case of communication networks where they are quite orthogonal. In the mean while, scaled density (*sc_den*) shows an inconsistent association throughout the studied network categories. It is close to *CCF* and *TPR* in biological networks but approaches *expansion* in social networks.

Based on this analysis, the above community quality metrics can be grouped in 6 classes that are presented in Table 5.3 according to their correlations over the studied dataset. In other words, these quality metrics are more correlated with ones in the same groups than with the others. Consequently, it is preferable to describe community structure using a cross combination of metrics in these groups. We present in the following section a characterization of internal community structure by a descriptive approach using an association between metrics in 2 different groups. Then we demonstrate by empirical evidences that our approach helps to recognize different community structures in communication, information, technological, biological, social, ecological and synthetic networks.

In fact, Figure 5.2 discloses that internal and external structures of communities are generally not related in structural communities. Meaning that having information about community internal structure would not provide much information about the external structure and how communities interact. They reflect different facets of community structure in networks. In Section 4.2, we used two external topological

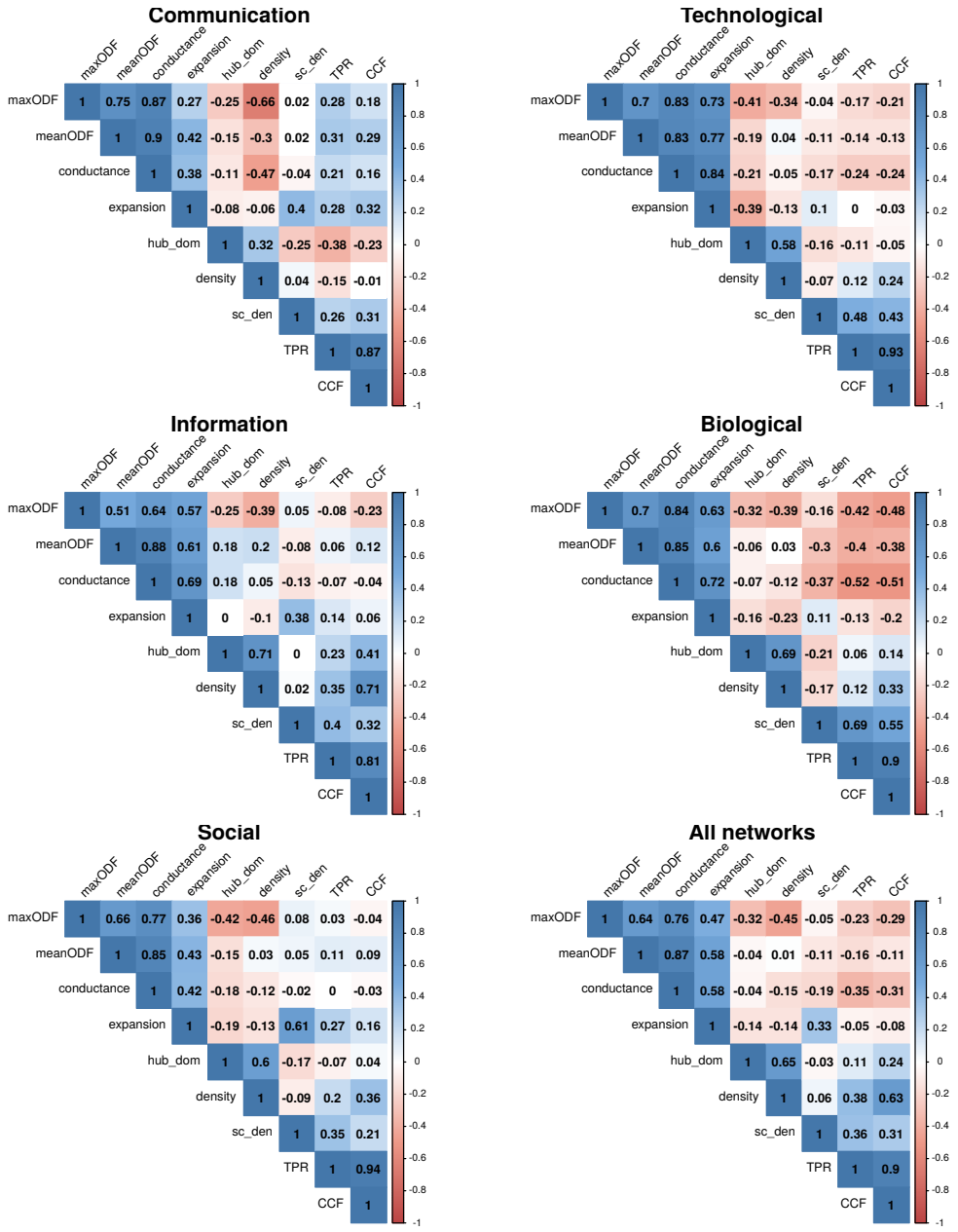


FIGURE 5.2: The Pearson correlations of community topological metrics measured on the communities detected by the set of community detection methods on the network dataset. These correlation are calculated based on scores of metrics measured on communities that contain at least 3 nodes. Metric correlations are analyzed by group of networks in different domains. Quality metrics are presented in the 6 sub-figures in the same order for a comparative observation. Correlation scores with low estimated significant levels (P-value > 0.01) are reproduced in a blank background. *maxODF*/*meanODF*, *hub_dom*, *sc_den*, *TPR*, *CCF* are shorten forms of maximum/average out degree fraction, hub dominance, scaled density, triangle participation ratio and community clustering coefficient respectively.

Metrics	Common concept
maxODF,meanODF,conductance	External activeness
expansion	External connectivity
hub_dom	Centralized connectivity
density	Internal edge density
sc_den	Average internal density
CCF, TPR	Internal triadic closure

TABLE 5.3: Groups of quality metrics that reflect different community structure aspects. Two metrics belong to a same category if they show a high correlation over the sets of structural communities. The **Common concept** column precises common structural features that members of each group reflect.

metrics to characterize different community archetypes and demonstrated community composition in real communities. Consequently, in this section, we focus merely on internal topology. However, the method is extensible for other choices as long as the metrics of interest expose meaningful connection patterns.

5.1.4 A bivariate description of topological communities

In this part, we present a categorization of community structure in an intuitive way to illustrate different modular structures detected in the network dataset. This can be considered as an extension of our previous proposition in evaluating communities using a descriptive approach (Dao, Bothorel, and Lenca, 2017b) the for internal aspect of community structure. We propose a categorization of modular structures using a couple of representative goodness variables to reflect highlight structural characteristics of communities in real world networks. Here, we focus on internal community structure, i.e. *density, sc_den, CCF, TPR* and *hub_dom*, will be in the shortlist of interest.

Which metrics?

It is well-known that *density* have a weakness in describing communities of different sizes since in real networks, the number of edges normally increases linearly with its size (real networks are often sparse) but the number of possible connection increases quadratically. As a consequence, the quality of large communities is usually under evaluated in comparison to small communities. Scaled density (*sc_den*) palliates this issue by multiplying the density with the community size, so mathematically its concept is very close with the average degree of a community which is measured by $\langle k \rangle = \frac{2m_C}{n_C}$. This metric reflects a very important feature of communities and is often used to evaluate community quality in a common sense. However, given a specific value of scaled density, one have several ways to redistribute edges inside a community in a manner that its internal topology changes crucially. In other words, scaled density does not characterize community internal configuration of degree. This is the reason why we do not use scaled density or traditional density to represent community topology.

The clustering coefficient and the triangle participation ratio (*CCF* and *TPR* respectively) are relatively close in their definition and it has been proved to be highly correlated through the previous empirical analysis. They both reflect an important topological feature by implying the concept that two arbitrary neighbors of a node

in a community should be also connected. This idea is somehow relatively close with the two variants of density since a network with high *CCF, TPR* scores is normally dense; however the opposite way is not always correct, which means a dense network does not necessarily have many triangular connections. Here, we select one metric among *CCF* and *TPR* to describe a common structural property called *transitivity*. Depending on the topology of networks or communities under consideration, one metric will work better than the other. On a same network, *CCF* score is generally lower than *TPR* score and hence *CCF* has a better resolution for networks where triangles are dense. On the the other side, *TPR* magnifies better topological differences in networks where only a few triangles exist. A further investigation on the dataset shows that there is approximately 90% of networks whose clustering coefficients are larger than 0.01 and this number is around 60% for a coefficient of 0.1 (see Figure 5.1(d)). This evidence leads to a preference of *CCF* over *TPR* to describe the clique dominance characteristic since the networks of interest are quite dense.

Another topological dimension that we employ to describe communities is *hub dominance* which is represented by *hub_dom* metric. Similarly to *CCF* and *TPR*, this metric reflect a structural feature of edge organization in a network or community. Specifically, it characterizes whether edges are distributed around one or a few members of their community and make them becoming hubs of connection. We illustrate in the next section that the combination two dimensions quantified by a couple of values (*CCF, sc_den*) reveals distinctive topological structures that could help to get insights on how communities in different networks look like.

Characterized topological community

After choosing two characteristics corresponding to two dimensions of community quality space, we describe internal community structures in different locations of this space. In order to maintain a clear distinction of representative topologies in different coordinates stays clear, we profile them in a coarse-grained description level. Specifically, we considerate 4 fundamental coordinated zones corresponding to 4 underlying topologies which are emphasized in Table 5.4. These classes of topologies could be explained as follows:

Type	Transitivity	Hub dominance	Topology
1	Low	Low	String-based
2	High	Low	Grid-based
3	Low	High	Star-based
4	High	High	Clique-based

TABLE 5.4: Four distinctive topologies characterized by *Transitivity* (*CCF*) and *Hub dominance* (*hub_dom*). There is no clear boundary between high and low values in the two dimensions, it is to be specified in accordance with the context. The distinction is more clear for medium and large size communities.

- **String-based topology** of a community is determined by low values of transitivity and hub dominance metrics. The low scores in these two representative dimensions regulate that there is relatively nearly no presence of clique structure nor hub node. For large communities, there could be one or a few hubs and cliques established, but not enough to dominate the global structure. These communities can be considered as a consequence of a ramification

between several sub-strings which generate a few loops and hubs in their intersections. String-based topologies could have a form that looks like chains, braids, rings, etc. as shown in Figure 5.3(a) depending on the context.

- **Grid-based topology** can be recognized by high values of transitivity and low values of hub dominance metric. The absence of hub nodes in the community organization is probably the most common feature with the string-based topology. Hence there is a homogeneity in the connection pattern between nodes of the grid-based topology. Besides, a high value of transitivity imply that the majority of nodes participate in tight-knit triangular structures which could themselves, at the same time, be attached between one to another to create larger and compacted structures. Grid-based communities generally have large sizes since small ones are usually degenerated into strings, loops or hub structures. In other words, grid-based structures are not recognizable by observing in a small scale or a local scale of communities. Popular topologies of this family consist of lattice topology, partially mesh topology as shown in Figure 5.3(b).

- **Star-based topology** which sometimes can be considered as tree-based topology is probably one of the most popular structures in networks of many fields. It can be perceived by low values of transitivity and high values of hub dominance. A low transitivity indicates that there is not or very few cliques. On the other hand, a high hub dominance value implies the occurrence of a “key connection" which attracts many edges in its community to become a hub. Some popular topologies which could be found in this class include: flake structure with one central hub and several peripheral hubs; hierarchical tree structure. There is actually a close relation between star-based/tree-based and string-based topology such that in some contexts, a hierarchical tree could be seen as a string and vice versa depending on the point of view. The essential difference of these two topologies which can be observed from our representation space is that the more *edge-attractive* the hub(s) in a community, the more it approaches the *star-based* topology. Note that in graph theory, a tree is an acyclic connected graph. However, in this context, trees accompanied by a few loops are classified in star-based topology unless loops dominate excessively the global community structure. Some representative star-based topologies are shown in Figure 5.3(c).

- **Clique-based topology** is quite common in small and very small communities but very rare in medium and large communities. It is recognized by high scores of transitivity and hub dominance. A simple interpretation of this class of topology is that every node must be connected with every other node of its community in an ideal situation. In a more relaxed context, nodes are not required to connect with all other nodes, but with a majority in order to establish a tight and compact structure. The clique-based topology is quite close to the grid-based topology in many ways. The most notable difference between them is that in a clique-based community, every node must be in the neighborhood of the other nodes of the community (direct connection or by one/two intermediate connections maximum), whether it is not necessary that every node must be close to each other in grid-based topology. Some representative clique-based topologies are shown in Figure 5.3(d).

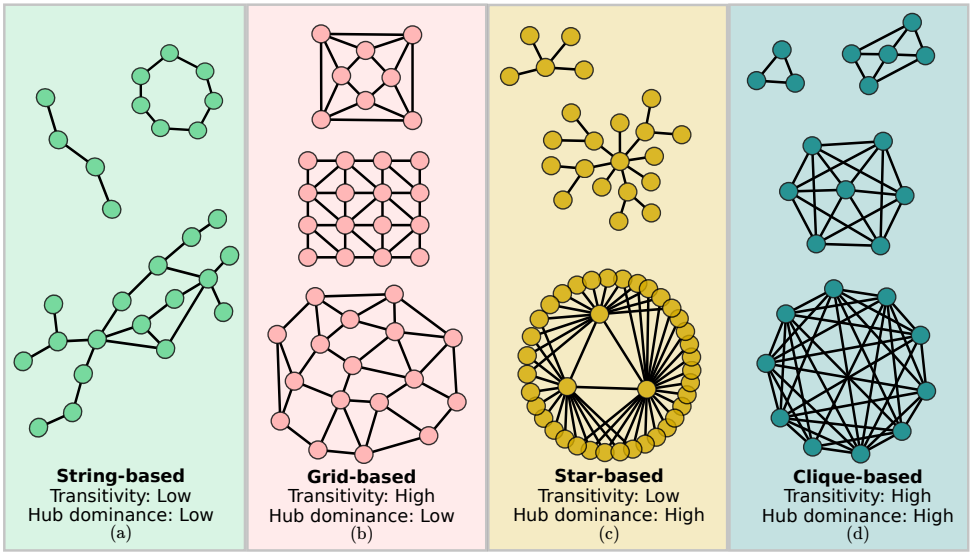


FIGURE 5.3: Topology families, from left hand side to right hand side (a) String-based, (b) Grid-based, (c) Star-based, (d) Clique-based. Depending on the context, one community can belong to different topological families according to specific criteria of analyst reflected by their determination of frontiers between these families.

A community structure whose transitivity and hub dominance scores are medium needs more investigation to be deduced. Since neither hub, clique nor random structure could dominate the whole community, its topology depends on the distribution of hubs and cliques in the community. It can be composed of a mixture of different component structures presented previously to become a homogeneous and more complex topology. It can also be a simple attachment between various dissimilar structures to establish a heterogeneous unit. In a point of view of dynamic community's evolution, communities in this class might be considered as being in a transition period between elementary structures. Alternatively, it could be a saturated state where communities attain a certain diversity and remain their complex structures. Further extent researches are deserved to cover more exhaustive aspects of communities in specific cases.

5.1.5 Locating network models in our topological space

Based on the idea that real networks and communities are constructed throughout different mechanisms, their topologies could be in some ways mimicked by using graph generative models. We attempt to locate networks created by popular graph models of the literature in the presented space in order to match them with the most resembling representative topology.

- **Erdős-Rényi model** (Erdős and Rényi, 1959) is among the first models proposed to describe the generation of *random graphs*. In this models, two parameters are required to generate a graph which is a fixed number of vertices n and a connection probability p between two arbitrary vertices (alternatively the number of edges m). Each pair of vertices is then connected independently of the other pairs with the probability p, which reflect the randomness

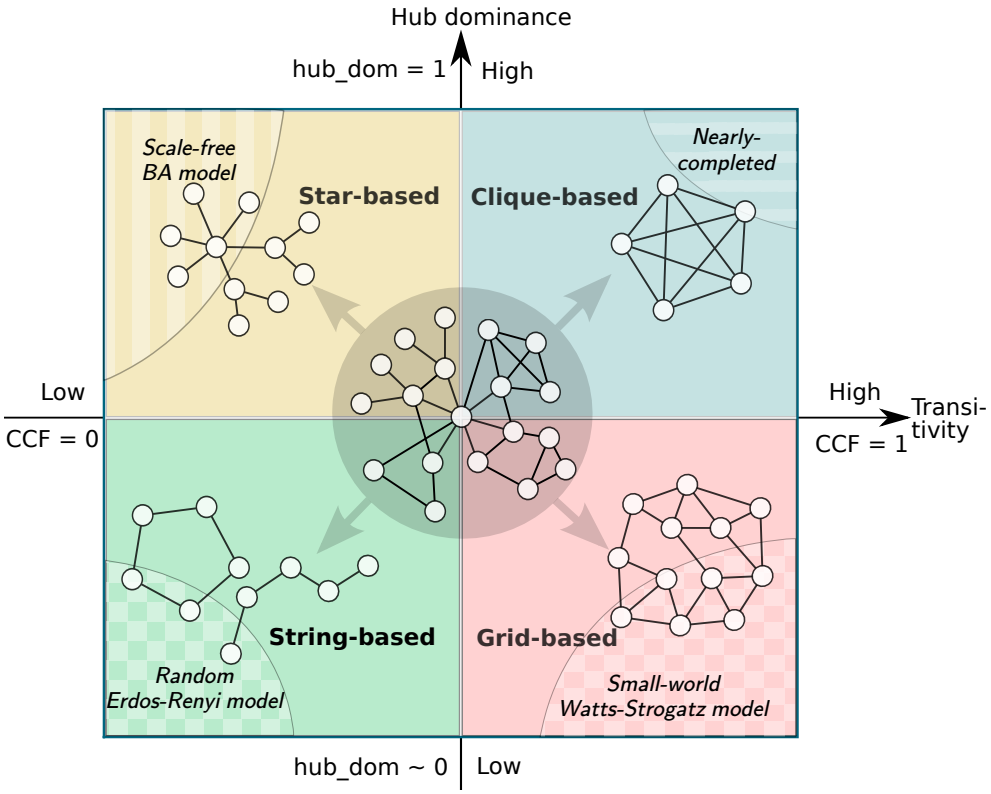


FIGURE 5.4: A categorization of internal community structure according to two topological property dimensions: hub dominance and transitivity represented by *hub_dom* and *CCF* respectively. Four representative topological communities are exemplified in 4 coordinating zones according to their corresponding *(hub_dom,CCF)* scores. The borders between different topologies are usually not clear and can be delineated according to the context. Characteristic community size should be taken into consideration when separating characterized zones since the bigger the community size, the more likely that hubs and cliques become less significant, which means lower thresholds will be more plausible.

property of the resulting graph. The expected number of edges and mean degree of the graph is calculated by $\langle m \rangle = \frac{pn(n-1)}{2}$ and $\langle k \rangle = p(n-1)$ respectively. The distribution of degree is binomial or Poisson for large graphs (Newman, 2001b). If we set n and p parameters of the model in a way that the model creates a random graph whose average degree approaches real networks: $\langle k \rangle = p(n-1) = c > 1$, where c is a constant and $c \ll n$; the graph will almost surely have a big component containing a large portion of vertices and very small components of less than $\mathcal{O}(\log(n))$ vertices. This configuration produces vertices that have all around $c > 1$ connections. In this context, without any further mentions, we refer to random graphs as ones created by this configuration, whose average node degrees approach those of real networks. Since a random network is constructed from a homogeneous stochastic mechanism, there is normally no hubs nor cliques which means low transitivity and low hub dominance values. A typical random graph constructed with a small value of p will have its largest component topology resembles the string-based

topology as shown in Figure 5.3(a). In an extreme regime, when the probability of connection p approaches 1, the associated random graph becomes *nearly complete* as the average degree $< k >$ approaches $n - 1$, which means every vertex connects with almost every other vertex as illustrated in Figure 5.3(d). The location of typical random graph's topology in function of two dimensions: transitivity and hub dominance is illustrated in Figure 5.4 in the bottom left-hand conner which associates to low scores of *CCF* and *hub_dom*.

- **Watts-Strogatz model** produces networks with *small-world* property, which normally means that any arbitrary pair of nodes can be connected through a small number of intermediate nodes and the average geodesic distance grows proportionally to the logarithm of the number of nodes n of the network: $L \propto \log(n)$. The model is built to characterize the observation that many real world networks show this property of small path length connectivity and highly clustered like regular lattices which implies a high presence of triadic closures (Watts and Strogatz, 1998). The generation of a small world network can somehow be considered as an interpolation between regular pattern networks and random networks. From a ring lattice with n nodes and k edges per node, each edge is redistributed randomly with a probability $0 < p < 1$. The authors find that a small value of p reduce significantly the path length characteristic of a regular network where nodes are only connected locally. This can be explained as rewired edges create shortcuts between remote areas of the network and hence reduce considerably network characteristic distance. A typical small-world network can be described using an intermediate value of p, so that the distance of two arbitrary nodes are very small, the clustering coefficient stay high since the random perturbation is not strong enough to break the local structures of nodes in the lattice ring. Besides, the shape of the degree distribution in the network is quite similar to that of a random graph where every node has around k neighbors and there is normally no hub dominance phenomenon. The topology of a typical small-world network is relatively homogeneous and looks like a grid-based topology from a local observation as shown in Figure 5.3(b). The location of its topology in function of two dimensions: transitivity and hub dominance is illustrated in Figure 5.4 in the bottom right-hand corner which associates to high *CCF* scores and low *hub_dom* scores.

- **Barabási-Albert (BA) model** (Barabási and Albert, 1999) is originated from a discovery that the distribution of vertex degrees in many real world networks such as: genetic networks and World Wide Web networks, are quite heterogeneous. Specifically, vertex connectivity follows a *power-law distribution*, which means the probability that a vertex connecting to k neighbors in its network equals $p(k) = Ck^{-\alpha}$ where the constant C is fixed by a normalization requirement and α is the power-law coefficient. This coefficient varies between 2 and 3 in many networks where the degree sequences are estimated to follow this model. Networks possessing this statistical feature are called *scale-free* by Barabási *et al.* to highlight the scale invariance property. This feature is explained by the authors as a consequence of two main mechanisms: firstly, networks expand gradually by attracting new vertices to existing ones; secondly, these new vertices have a tendency to attach preferentially to vertices that are already well connected. That is why this model is often known as preferential attachment model, implying that the more connected a vertex, the more likely it receives new edges. This mechanism makes scale-free networks hub-profuse since *"richer nodes get richer"*, and hence hub dominance values

of scale-free networks are usually high. On the other hand, the associated clustering coefficients are usually low and are decayed quickly in function of network sizes (Klemm and Eguíluz, 2002), (Fronczak, Fronczak, and Hołyst, 2003), which means low transitivities. Consequently, typical scale-free networks have a close structure with that of star-based topologies as depicted in Figure 5.3(c). The location of scale-free networks in function of two dimensions: transitivity and hub dominance is illustrated in Figure 5.4 in the top left-hand corner which associates to low *CCF* and high *hub_dom* scores.

5.2 Community profiles in different network categories

In this section, we show empirical evidences to associate structural communities in real world networks with corresponding topologies determined by the bivariate representation. In order to do that, first *CCF* and *hub_dom* quality scores are calculated over the whole set of communities detected on the network dataset by the presented algorithms. Later, these communities are located in the characterized space in function of their couples of values *(CCF,hub_dom)* which represent transitivity and hub dominance respectively. The distribution of communities on this two dimensional space helps to match the most corresponding topologies with each set of communities thanks to the topology characterization presented in the previous section. Since it has been noticed that some structural characteristics might differ between small communities called *micro-communities* and large communities called *macro-communities* (Lancichinetti et al., 2010), we proceed to analyze them separately. Figure 5.5 delineates the distributions of small communities of 10 nodes or less in 6 different network groups including communication, technological, information, biological, social and miscellaneous networks as described in Table 5.2. The homologous distributions for large communities of more than 10 nodes are depicted in Figure 5.6.

At a first sight, it is easy to remark that there is a much higher diversity of structures at the large scale communities than at the small scale communities as the distributions are much more expanded over the space in the former case. It is reasonable since there are much more possibilities how nodes can be connected in a large community than in a small one. Hence large communities' structures are more distinctive and at the same time more complex. Specifically, most of small communities are found around two axis where $CCF = 0$ or $hub\_dom = 1$, especially at their crosspoint where $CCF = 0$ and $hub\_dom = 1$. It means star-based and hub dominated structures are very well representative for small communities of every network category. On the other hand, grid structure is totally absent at this size scale, which is quite predictable since it requires a large number of nodes for a grid to be formed. Additionally, the heavy-tail degree distribution recognized in many real world networks make grids less likely to be established.

In information and miscellaneous groups, communities are much more rich in structure comparing to the other categories at both scales. Concretely, besides star-like modules, there are also many clique-like communities and mixture structures since clustering coefficient values in these groups stretch across the whole range. Similarly for hub dominance values which are measured approximately from 0.4 to 1 at the small scale and from 0 to 1 at the large scale. Although there are some differences in community structure between various network categories, at a small scale, it not very obvious to distinguish them using the proposed representation. We introduce in the following part a detail inspection, especially for large communities,

which would reveal essential distinctions between community structure of each network category. The distribution of communities over the profiled map characterizes the mesoscopic structural identity of networks.

5.2.1 Communication networks

Communication communities consist in subnetworks of message exchange in social networks, email communications, discussions in forums, etc. From the bivariate distributions of communities shown in Figure 5.6(a) and 5.5(a), it can be recognized that structural communities are quite homogeneous in terms of topology in both large and small communities. The majority of them have star-based topologies with very strong hubs which connect to almost every other node in their communities and very few number of clique connections. In other words, communication communities are in general very remarkably high centralized and very low transitive. This property is less clear in large communities than in small communities since the larger a community, the more likely non-hub nodes have chances to create interconnections and possibly establish peripheral hubs. This mechanism also gives rise to a few numbers of multi-hub topologies in large communities. Besides, a small number of hub-absent communities and mesh communities can be discerned. However, they are quite outnumbered by hub structures in this network category. This revelation denotes that exchanges in communication networks often happen around some *central elements* which convey access to their surrounding elements. Figure 5.7 illustrates some typical structural community topologies that have been identified in the communication network dataset. Among them, star-like topologies with one dominating hub as shown in Figure 5.7(a),(b) are among the most representative. Besides, there are also communities where hubs are less influential and the presence of a few cliques can be recognized as illustrated in Figure 5.7(c),(d). However, within the list of network categories that has been analyzed in this study, communication communities show a clearest and strongest hub-periphery connection pattern with more than 80% of communities where there are at least 1 node connected to at least 90% of node members in its community and very few periphery-periphery connections. By consequence, communication communities are commonly quite sparse in comparison to other types of networks. Moreover, previous study demonstrated in Figure 5.4 helps to infer that communities networks reveal strong *scale-free* property. Consequently, a preferential attachment mechanism with an amplified connection probability to hub nodes would efficiently mimic the structure of real world communication networks.

5.2.2 Technological networks

Technological communities include subnetworks in peer-to-peer Gnutella file sharing networks, Internet, highway and airport circulation systems, etc. The most notable similarity between technological communities and communication communities is the high presence of hub-based topologies, especially in small communities as can be seen in Figure 5.5(b). In large communities, however, technological communities show a quite discernible connection pattern as hubs are less *powerful* in their local as can be interpreted from Figure 5.6(b). Quantitatively, the majority of hubs in technological networks embrace around 40% to 60% of nodes in their communities. Additionally, the withdraw of super dominating hubs is replaced by the occurrence of more triadic connections in technological communities. It can be explained by

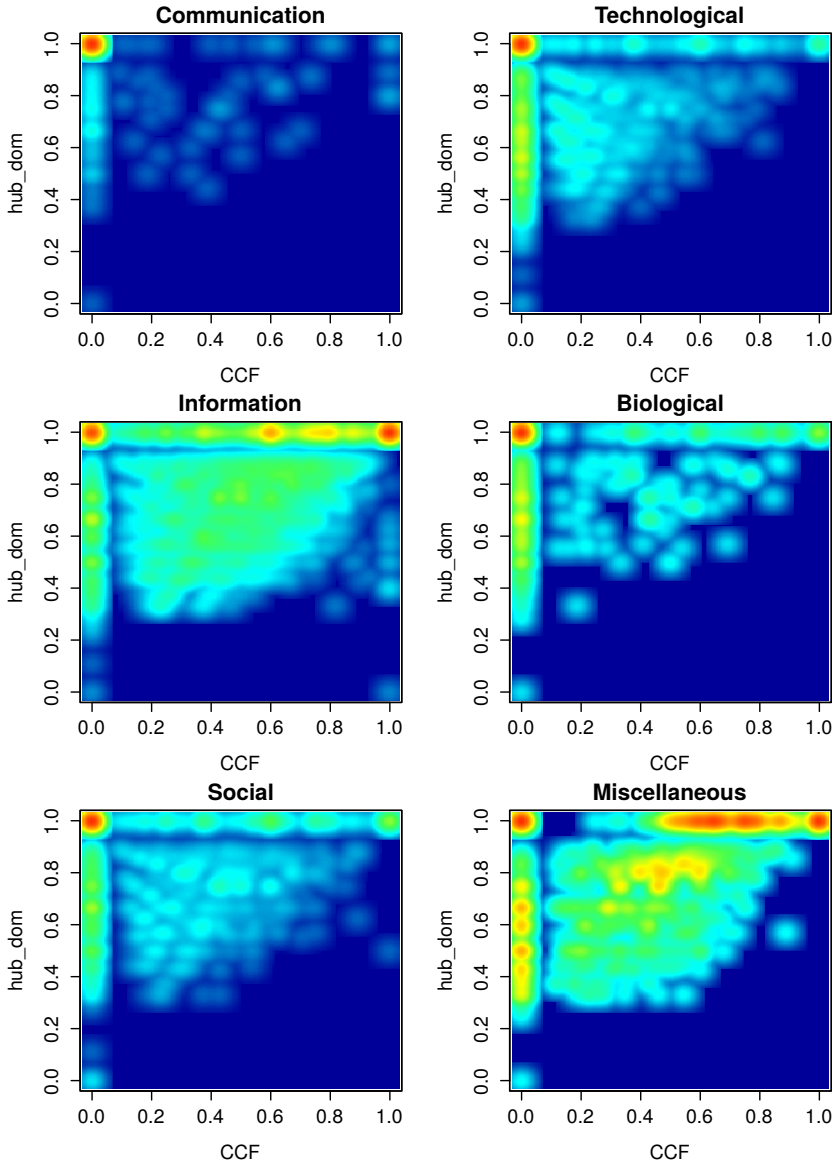


FIGURE 5.5: Heat maps of distributions of small structural communities detected on different categories of networks are presented on a two dimensional space characterized by transitivity (CCF) and hub dominance (hub_dom). Only communities of **10 nodes or less** are included. From left to right, top to bottom (a) Communication, (b) Technological, (c) Information, (d) Biological, (e) Social, (f) Miscellaneous consists in power networks, ecological networks, artificial networks, etc.

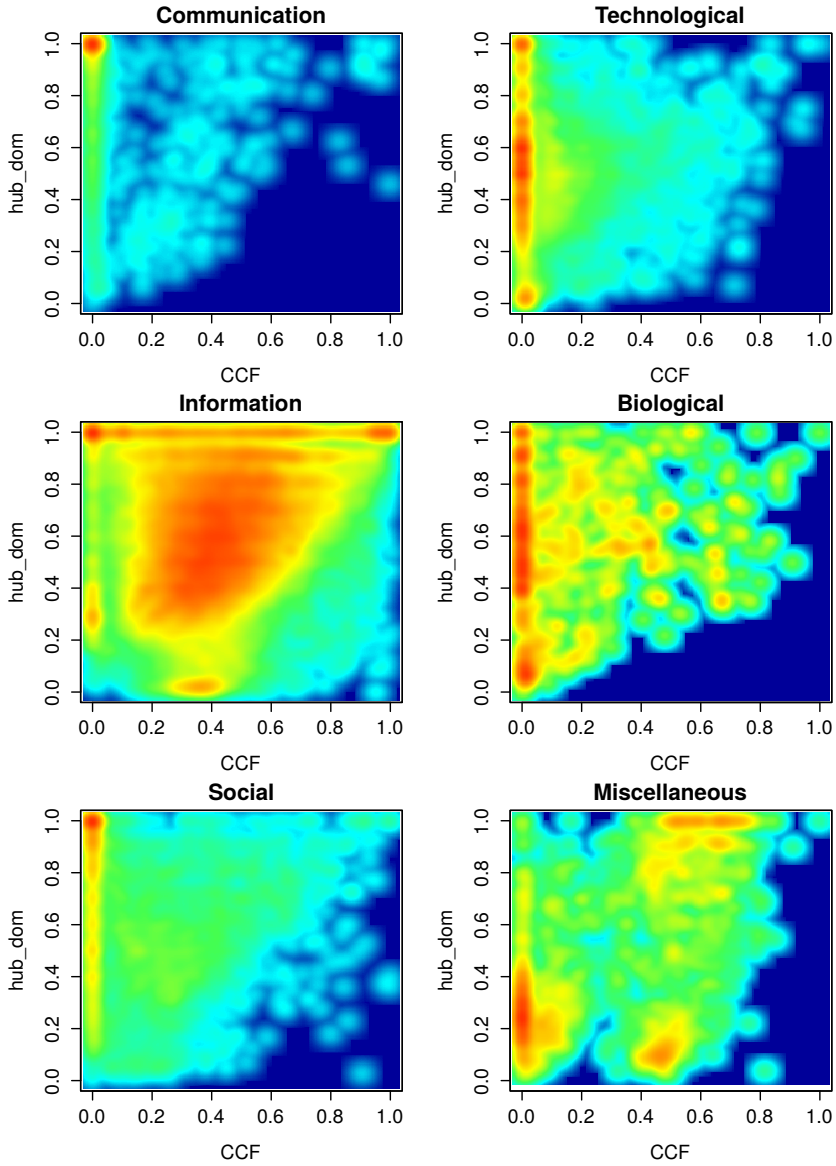


FIGURE 5.6: Heat maps of distributions of large structural communities detected on different categories of networks are presented on a two dimensional space characterized by transitivity (CCF) and hub dominance (hub_dom). Only communities of **more than 10 nodes** are included. From left to right, top to bottom (a) Communication, (b) Technological, (c) Information, (d) Biological, (e) Social, (f) Miscellaneous consists in power networks, ecological networks, artificial networks, etc.

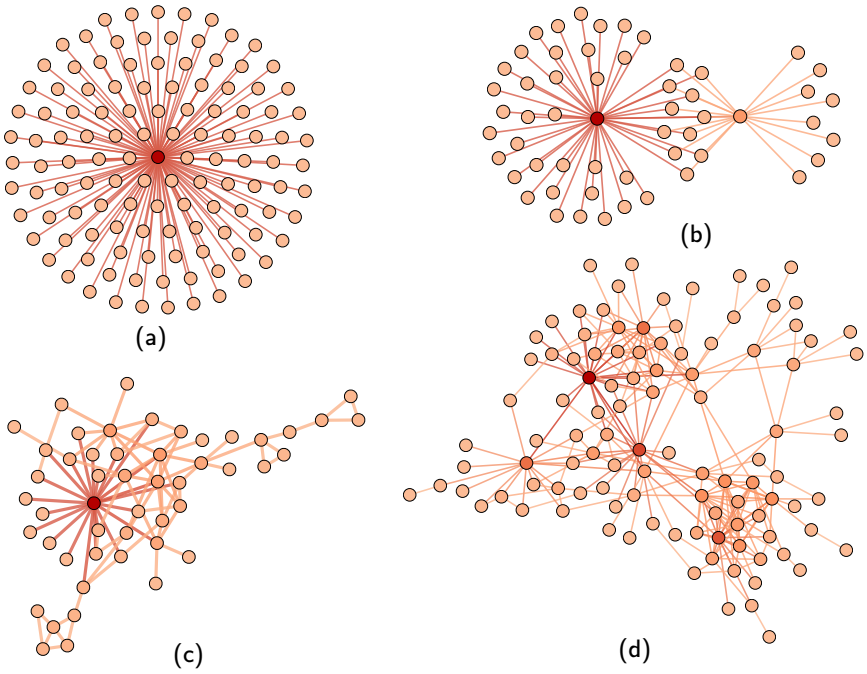


FIGURE 5.7: Some representative topologies detected in *Communication networks* with their corresponding scores (CCF, hub_dom). Topologies are ordered from the most famous to the less famous in their network category as shown in Figure 5.6(a), 5.5(a). Hub nodes are darker than peripheral nodes. (a) Email traffic in an European research institution (Rossi and Ahmed, 2015) community - $(0, 1)$; (b) Wikipedia adminship vote (Leskovec and Krevl, 2014) community - $(0.03, 0.87)$; (c) Email communication Enron network - $(0.07, 0.90)$; (d) Community of email exchange in an university - $(0.28, 0.23)$.

the fact that in some infrastructure networks such as highway networks or the Internet, hubs are often constructed to have a controlled influence and are normally compensated by resilient connections or supplement hubs in order to reduce workload, vulnerability or crucial impact caused by their dysfunctionality. Figure 5.8 illustrates some community topologies that have been identified in the technological network dataset. Topologies whose hubs connect to around a half of node members as depicted in Figure 5.8(a),(b) are among the most representative of networks in this class. There is usually a stratification in the connection pattern as many nodes are connected to a central node by intermediate nodes. This phenomenon can be considered as a presence of hierarchical organization frequently found in technological systems. Besides, there is also a considerable number of star-based structures such as those of communication case and string-based structures as shown in Figure 5.8(c) and 5.8(d) respectively. In a general view, the scale free property is quite clear although hub attractiveness is relatively reduced comparing to communication networks. A preferentially attachment fitness provided by a model such as Barabási-Albert would allow to imitate well technological structural networks.

5.2.3 Information networks

Information communities contain subnetworks in citation networks, scientific collaboration networks, research engine networks, recommendation networks, etc. Within the studied networks, information networks exhibit the most diverse topological

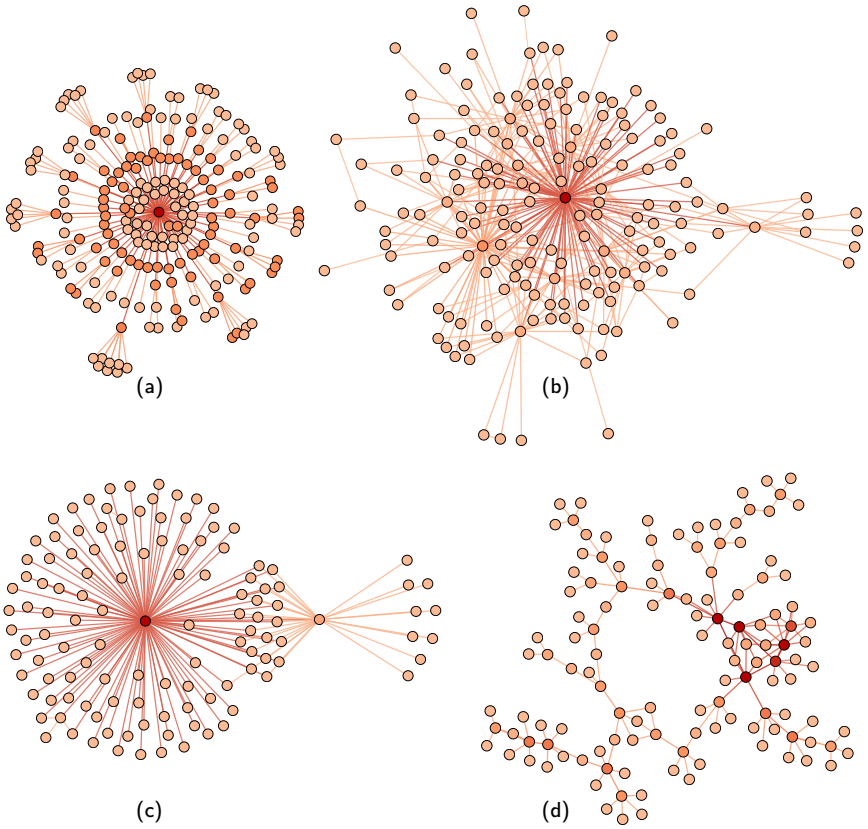


FIGURE 5.8: Some representative topologies detected in *Technological networks* with their corresponding scores (CCF, hub_dom). Topologies are ordered from the most famous to the less famous in their network category as shown in Figure 5.5(b), 5.6(b). Hub nodes are darker than peripheral nodes. (a) A community of users of the Pretty-Good-Privacy algorithm for secure information interchange - $(0.01, 0.48)$; (b) WHOIS Internet IP community - $(0.07, 0.65)$; (c) A community of AS Caida Internet infrastructure recorded in 2007 - $(0.01, 0.92)$; (d) A Gnutella peer-to peer network community - $(0.01, 0.07)$.

pattern with the bivariate distribution of communities expanded over a wide range of hub dominance axis and transitivity axis as shown in Figure 5.5(c), 5.6(c). Globally, information communities are different from communities of the other network categories by their high transitivity. Such that cliques are very well presented in many information networks as depicted in Figure 5.9. Many information communities can be considered as mixtures of different basic topologies of star-based, string-based, clique-based and grid-based such as the community of collaboration in Arxiv Condensed Matter network shown in Figure 5.9(h). The presence of hubs in information networks is still high, however they are not anymore the only elements who connect different members of networks. Consequently, information networks are normally much more dense and well connected than other types of networks of the same size scale. This is probably the most representative connectivity feature of information networks. Similar results related to dense and clique structures have been also found by Lancichinetti *et al.* (Lancichinetti et al., 2010). Figure 5.9(a-h) depict some representative communities that have been discovered in some information networks. While the structure in Figure 5.9(d) resembles a star-based topology with a sequence of periphery-periphery connections; the one in Figure 5.9(e) of Arxiv

High Energy Physics collaboration looks like a complete network with some ill-connected nodes. Figure 5.9(c,g) demonstrating web and recommendation systems reveal a mixture structure where hubs can be well recognized and clique presence is also remarkable at the same time. The hybrid structure is globally more blended in communities of Figure 5.9(a,b,h) than the others. The diversity in the structure of information networks can be explained by the way we define this category. In fact, a commercial recommendation system could be very unalike a web citation or a collaboration network, even though they are all considered to be information systems in the network science community. Furthermore, their structures are normally exposed to several complex phenomena that regulate network interactions. Hence, simulating information networks merits more investigation on each concrete case to determine the mechanism that reflects well the mesoscopic organization.

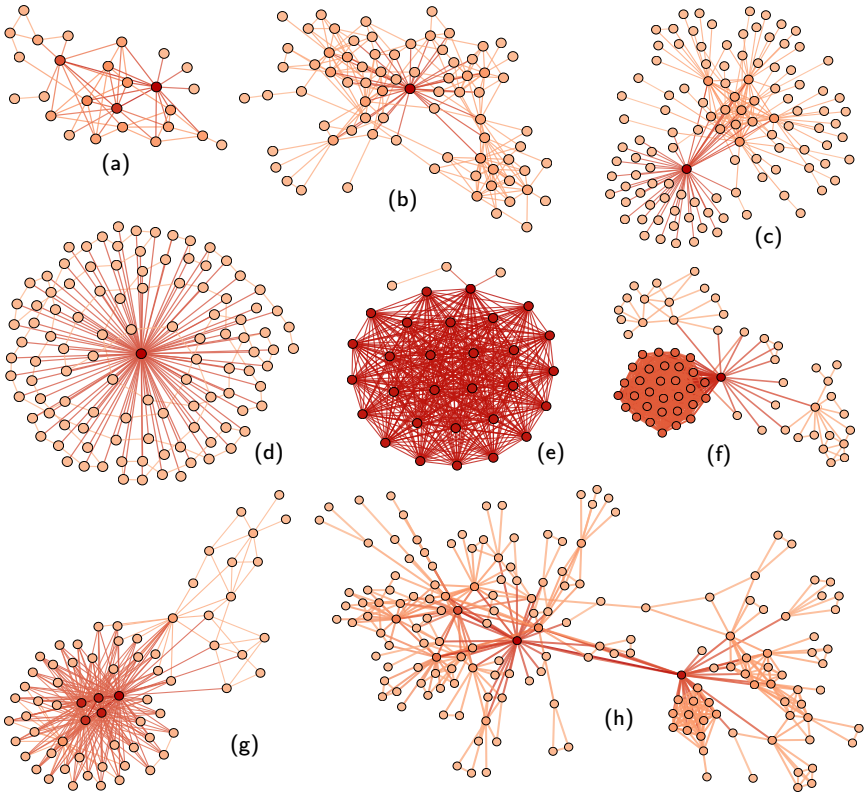


FIGURE 5.9: Some representative topologies detected in *Information networks* with their corresponding scores (CCF, hub_dom). Topologies are ordered from the most famous to the less famous in their network category as shown in Figure 5.5(c), 5.6(c). Hub nodes are darker than peripheral nodes. (a,b,g) Amazon recommendation groups of products - $(0.40, 0.52)$, $(0.33, 0.45)$ and $(0.24, 0.76)$ respectively; (c) An educational web system cluster - $(0.30, 0.43)$; (d) A group of Indochina websites recorded in 2004 - $(0.05, 0.98)$; (e-f) A community of Arxiv High Energy Physics collaboration - $(0.99, 0.97)$ and $(0.95, 0.99)$; (h) A collaboration community of Arxiv Condensed Matter network - $(0.44, 0.36)$.

5.2.4 Biological networks

Biological communities comprise subnetworks in brain networks, yeast networks, protein-protein interaction networks, metabolic reaction networks, etc. In some

ways, their topologies resemble with technological networks as it can be observed through their distributions in Figure 5.6(b) and 5.6(d). The most remarkable discrimination of connection pattern between biological networks with the other ones are their string-based rich structure as can be seen through communities shown in Figure 5.10(a), (b), (c). The high presence of chains or strings in biological networks has been also found by the other studies using different approaches such as in (Lancichinetti et al., 2010), (Guimerà, Sales-Pardo, and Amaral, 2006). This may be caused by the fact that many biological pathways, which are series of molecular interactions, are included in the analysis and contribute to the high presence of strings. Additionally, many biological networks are only constructed partially due to high complexity in construction time and technical constraints in biochemistry (Newman, 2010). Therefore, we often observe and analyze small fragments of networks where many connections are missing.

Still, there exist biological networks whose topologies are star-based or hybrid as those of communication networks, technological networks or information networks. However, the hub dominance is globally less important as biological communities are normally small and hubs connect to much less number of their surrounding neighbors. A local observation on biological networks probably discloses random structures in many parts of the networks although hubs are still well widespread. This emergence of random structures could be the most typical characteristic that differs biological networks from the others. Finally, popular properties such as *scale-free*, *small-world* are less significant in biological class than in information or technological class.

5.2.5 Social networks

Social communities involve subnetworks of friendship networks, share or re-tweet networks, followings in Google Plus, Facebook, Twitter, Youtube, etc. Our analysis shows a high similarity in the distribution of large communities in the social networks and communication networks as depicted by Figure 5.6(a), Figure 5.6(e). For small communities, social networks are closer to technological networks and biological networks as shown in Figure 5.5(b), 5.5(d), 5.5(e). A reasonable explanation for the popularity of the star-based topology in social network is that there are many well-known users who are followed or subscribed by a large number of peoples and are becoming mega-connected hubs. Additionally, many samples of social networks that are studied consist of ego networks of celebrities in social media, which makes them intrinsically high centralized around some mega-hub nodes. The only difference with communication communities that has been found in this study is that there are generally more connections between peripheral nodes in social communities. This can be interpreted by the fact that friendship relations or following interactions are generally more frequent than communication interactions. Although different networks of social and communication have been used in this analysis, it makes sense to explain that many users are connected in a social media without or very few communicating interactions in the same channel. For example, two users could be connected on Facebook as friends, but they never exchange any message on the Facebook conversation platform which makes that the number of social connections exceeds the number of communications. Figure 5.11 demonstrates some popular topologies of communities in social networks. Note that these topologies are not chosen to argument the differences between various social networks and it is not the objective of this study. They are listed to illustrate some typical and representative structural communities that we discover in the network dataset. Social

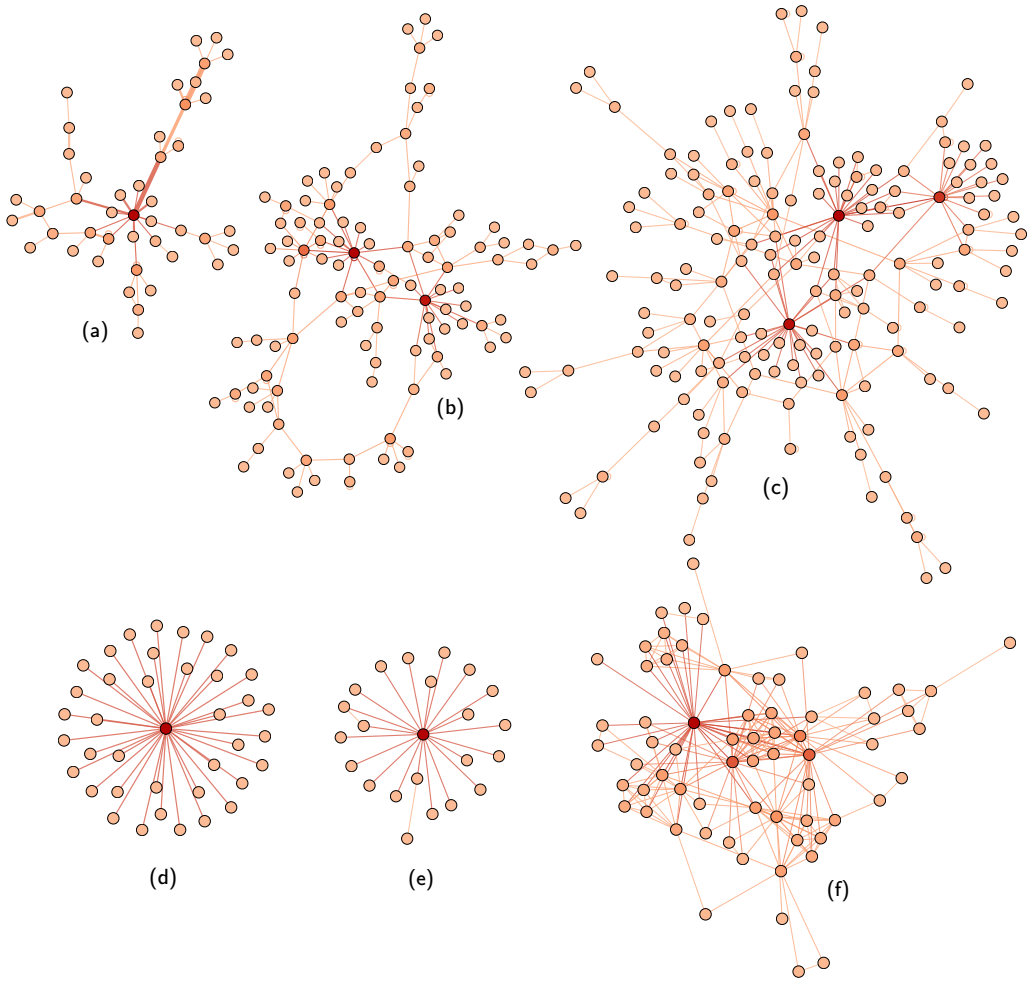


FIGURE 5.10: Some representative topologies detected in *Biological networks* with their corresponding scores (CCF, hub_dom). Topologies are ordered from the most famous to the less famous in their network category as shown in Figure 5.5(d), 5.6(d). Hub nodes are darker than peripheral nodes. (a) A circuit of medulla of drosophila fly brain - $(0.06, 0.44)$; (b-c) A protein-protein interaction network of yeast - $(0.03, 0.16)$ and $(0.05, 0.16)$ respectively; (d-e) protein interactions of drosophila melanogaster $(0, 1)$ and $(0.01, 0.95)$; (f) A cluster of human disease network $(0.47, 0.51)$.

networks show a clear *scale-free* property as in communication and technological networks, however they are less affected by mega-hubs and are partially occupied by clique-based structures and many random connections like that of *small-world* phenomenon.

5.2.6 Ecological, infrastructure and synthetic networks

This group covers subnetworks in ecological networks, some power system networks, sport competition networks, synthetic networks, etc. Here, we find many structures, especially in Lancichinetti-Fortunato-Radicchi (LFR) synthetic networks (Lancichinetti, Fortunato, and Radicchi, 2008), that are not very popular in the previously studied networks. Specifically, except for information networks, structural

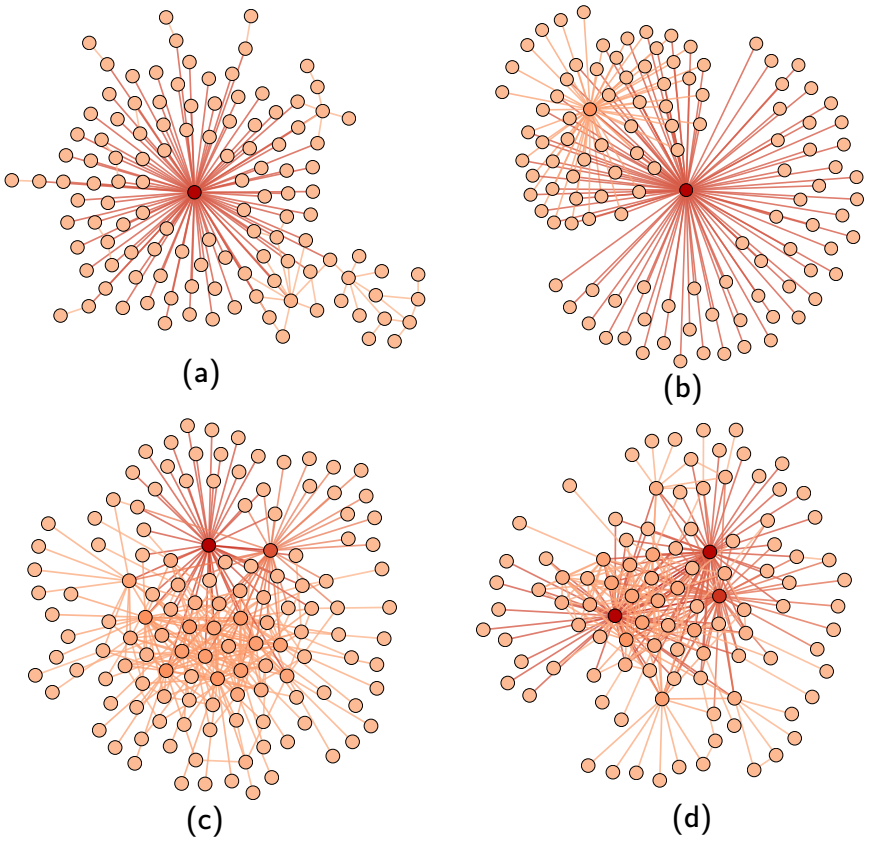


FIGURE 5.11: Some representative topologies detected in *Social networks* with their corresponding scores (CCF, hub_dom). Topologies are ordered from the most famous to the less famous in their network category as shown in Figure 5.5(e), 5.6(e). Hub nodes are darker than peripheral nodes. (a) A structural community in Youtube video sharing friendship network - $(0.01, 0.81)$; (b) A community in Google Plus network - $(0.02, 0.95)$; (c) A political re-tweet network in Twitter - $(0.12, 0.60)$; (d) A subnetwork of location-based social networking Brightkite - $(0.27, 0.51)$.

communities in the other types of networks are usually very hub-centralized and relatively low in transitivity. On the contrary, in LFR networks, cliques are quite popular and normally aggregated to produce compacted structures as illustrated in Figure 5.12(c), which makes the communities highly transitive. Additionally, although structures of LFR networks are regulated by many configuration parameters, their hubs generally have less impact in their neighborhoods than those of real world networks such as in social or communication. This is one property that makes a huge difference between LFR benchmarking networks and real world networks. Some other discovered structural communities are illustrated in Figure 5.12. In a general view, community detection methods identified well compacted sub-graphs in most of the cases.

Our previous empirical study uncovers that networks across different categories including communication, technological, information, biological and social networks might have different community structures and can be described by distinguishable characterized topologies.

The difference of modular topology between networks in various categories could help to construct network profiles or network signatures by domain of study, and hence open a possibility for creating adapted network generative models, network

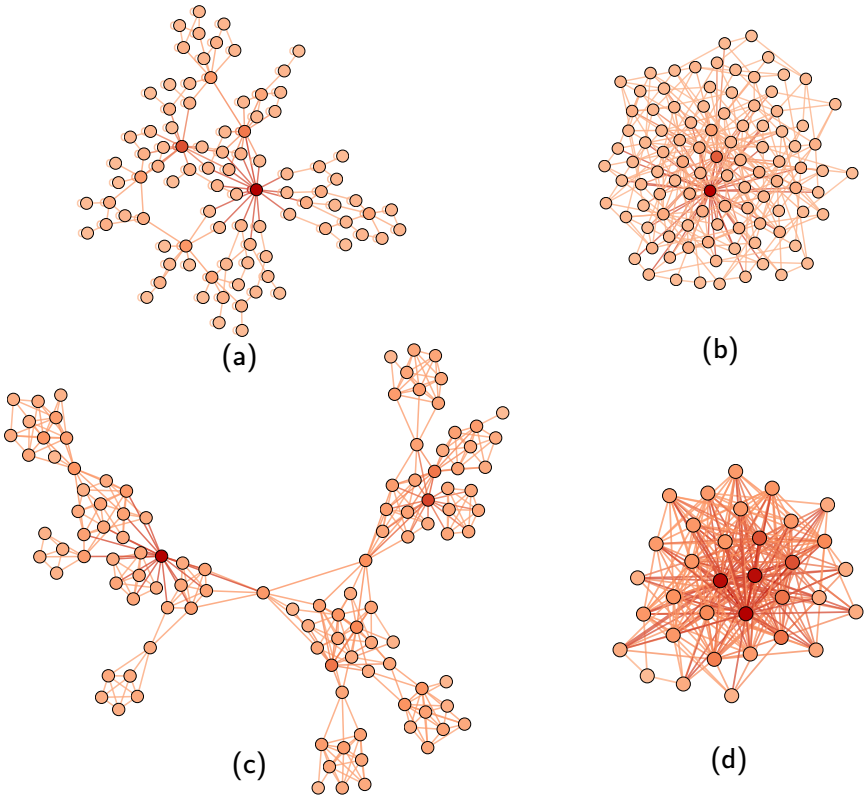


FIGURE 5.12: Some representative topologies detected in miscellaneous group with their corresponding scores (CCF, hub_dom). Topologies are ordered from the most famous to the less famous in their network category as shown in Figure 5.5(f), 5.6(f). Hub nodes are darker than peripheral nodes. (a) A cluster of a power network system - $(0.07, 0.21)$; (b) A quadratic sieve of a factorization of a 130 bit number - $(0.08, 0.39)$; (c) A cluster of a Lancichinetti-Fortunato-Radicchi (LFR) synthetic network (Lancichinetti, Fortunato, and Radicchi, 2008) - $(0.56, 0.18)$; (d) A cluster in an ecological network - $(0.51, 0.94)$.

class prediction algorithms, dynamical processes simulation and analysis, etc. Specifically, since networks in each domain reveal some particular modular structures, the mechanisms which are responsible for their creations, evolutions, degradations are also discernible. Hence, different simulation or analysis strategies will generate different impacts on the networks in a predictable way if their structures are well understood. In other words, the network structure profiling assists to achieve suitable network analysis processes and to interpret obtained results without requiring expensive brute force analysis.

5.3 Related work

Many efforts have been devoted to characterizing community structure in networks, each one with a specific approach to reveal different structural properties. In the best of our knowledge, we have not yet well-known empirical study that leverage the availability of a large corpus of real-world networks in order to understand topological properties across different domains. Although many researches have been well inspecting different quality metrics to understand community structure in ad-hoc

networks, a concrete and systematic method for summarizing and extracting topological information is still in demand. The content presented in this chapter is a part of an effort to discover real-world networks and community structures. Even being a novel approach, our work were strongly inspired by several closely-related studies in the art, which can be cited in the following.

Lancichinetti *et al.* characterize community structures of complex networks in different domains by observing the evolution of various qualities such as community scaled density, average shortest path, max internal degree, etc. in large scale networks according to discovered community size (Lancichinetti et al., 2010). The evolution of these qualities in function of number of nodes in each cluster helps the authors to deduce and characterize different structures found in many class of networks such as: Internet, communication, information, biological and social networks.

Guimera *et al.* demonstrate that modular networks in real world can be classified into distinct functional classes depending on the composition of connection profiles between their nodes (Guimerà, Sales-Pardo, and Amaral, 2006). Specifically, by using two metrics including within-module degree z and participation ratio P (Guimerà, Sales-Pardo, and Amaral, 2004), a node in a community is characterized by seven different roles of hubs and non-hub nodes. Once the role of every node in a network partition is determined, the connectivity profiles of interactions in the network can be analyzed. Specifically, the authors determine two main classes of networks based on the presence of role-to-role connectivity profiles. The first class called *string-periphery* includes metabolic and air transportation networks which are rich in ultra peripheral interactions and hub interactions. The second class called *multi-star* includes protein interactome and Internet networks which are, on the other hand, rich in ultra peripheral-provincial hub interactions.

Leskovec *et al.* investigate the variation of community structure in large scale networks using *conductance* metric (Leskovec et al., 2008). In fact, the authors measure the variation of the lowest community conductance in function of community size. This variation depicts a so-called *network community profile* which helps to characterize community quality over a wide range of size scales. The authors also point out that communities attain the best quality (in terms of conductance) at a characteristic size of around 100 nodes and provide evidences of a high presence of core-periphery community structure in real networks through numerous empirical experiences. In another paper (Yang and Leskovec, 2013), the authors also compared the performance of 13 quality functions in terms of their efficiencies to identify community goodness properties such as density, cohesiveness as well as the consistency of these quality functions to many simulated perturbations. The studies contribute to the understanding of the property of different quality functions.

Coscia *et al.* generalize the problem of community detection discovery by reconsidering the question of what can be considered to be a community (Coscia, Giannotti, and Pedreschi, 2011). The authors then resume popular methods in the literature according different quality aspects such as density-based, vertex similarity-based, action-based or influence propagation-based. A definition-based classification of community discovery methods according to a large number of community features is then introduced. This classification approach shifts the attention from how communities are detected to what kind of communities to detect and provides another point of view regarding to community detection.

The most common and fundamental point between this study and the previously mentioned work is the exploratory objective to characterize communities in complex networks by observing qualities using statistical metrics. Concretely, we contribute

a methodology to describe community topologies in a systematic and generic way that can be extended to any category of networks. This means one can mechanically apply the same analysis procedure to explore community structures of any network of interest.

5.4 Conclusion

In this chapter, we provide a novel analysis process to categorize mesoscopic organization of networks into four essential topological groups which show different node interaction patterns. Each representative group is then associated to the corresponding graph generative model that produces a high similarity in connection patterns. Surprisingly, our empirical study uncovers that networks across different categories including communication, technological, information, biological and social networks might have different community structures and can be described by distinguishable characterized topologies. These differences sheds light on how different network models should be used to represent real-world networks and also how these models should be parameterized. For instance, *Barabási-Albert* model can be adopted to describe communication or technological networks while *Watts-Strogatz* model could be better for information networks. It is worth noting that a development of domain-specific clustering techniques is envisioned as an important task (Fortunato, 2010).

Revealing the difference of modular topology between networks in various categories could help to construct network profiles or network signatures by domain of study, and hence open a possibility for creating adapted network generative models, network class prediction algorithms, dynamical processes simulation and analysis, etc. Specifically, since networks in each domain reveal some particular modular structures, the mechanisms which are responsible for their creations, evolution, degradation are also discernible. Hence, different simulation or analysis strategies will generate different impacts on the networks in a predictable way if their structures are well understood. In other words, the network structure profiling assists to achieve suitable network analysis processes and to interpret obtained results without requiring expensive brute force analysis.

Finally, since we focused on characterizing different aspects of community structures, community detection algorithms are used to compare networks across different domains without questioning too much about their performance. This approach allows to shed light on different ways that nodes in real-world networks interact with each other. Certainly, the differences between these algorithms can not be neglected, which is why the next chapter will be dedicated to a more comprehensive analysis on this direction.

Chapter 6

Comparative evaluation of community detection methods

In Chapter 5, we have focused on characterizing community structure in networks and proposed a generalized method to categorize node interactions by several topological categories. We used community detection as *magnifier* to look inside large-scale graphs and extract structural information. The difference between discovery methods in identifying communities have not been questioned to leave the space for analyzing network structure. In this chapter, another investigation be will be conducted in order to access the performance of community detection methods according to different quality criteria. These are two essential parts that need to be inquired in order to be able to determine suitable detection mechanisms. Although there are several comparative approaches to highlight the distinction between these methods, we focus on some of the most primary aspects. Section 6.1.1 demonstrates a meticulous analysis about the computation time performance in a comparison to theoretical calculations. Section 6.1.2 reveals the fitting quality, i.e. the number of clusters detected by each method, as it is an important factor that one would consider for a clustering problem. The analyses in the latter section give raise to our novel approach proposed to estimate the closeness between different community detection methods based on community size distribution, which will be presented in Section 6.2. Then, many advanced quality functions are examined to profile the behaviors and the performances of community detection methods. We are interested in studying the similarity of methods in their capacity of discovering communities having some expected qualities in Section 6.3. Finally, we estimate empirical pairwise proximity of these methods by comparing community sets that they discover on the network dataset in Section 6.4. These analyses reveal interesting information that are useful for drawing conclusion about detection performance.

We reuse the dataset introduced in Chapter 5, which consists in more than one hundred networks of five different categories. The information about these networks are summarized in Table 5.2 and some important statistical measures on are illustrated in Figure 5.1 of Chapter 5. As mentioned previously, this dataset consists in networks whose structural properties spread over a wide range of values. This make our analysis less impacted by the dependence of detection performance on the input data.

Approach	Method (section)	Label	Implementation
Edge removal	Girvan-Newman (3.2.2)	GN	igraph
	Radicchi et al. (3.2.2) for $g = 3$	RCCLP-3	Authors[1]
	Radicchi et al. (3.2.2) for $g = 4$	RCCLP-4	Authors
Modularity optimization	Clauset et al. (3.2.3)	CNM	igraph
	Blondel et al. (3.2.3)	Louvain	Authors[2]
	Newman (3.2.4)	SN	igraph
Dynamic process	Pons et al. (3.2.5)	Walktrap	Authors/igraph
	Rosvall et al. (2007) (3.2.5)	Infomod	Authors[3]
	Rosvall et al. (2009) (3.2.5)	Infomap	Authors[4]/igraph
Statistical inference	Lancichinetti et al. (3.2.6)	Oslom	Authors[5]
	Karrer et al. (3.2.6)	DCSBM	Authors[6]
Other methods	Reichardt et al. (3.2.7)	RB	igraph
	Raghavan et al. (3.2.7)	LPA	igraph
	Xie-Szymanski (3.2.7)	SLPA	Authors[7]
	Demeo et al. (3.2.7)	Conclude	Authors[8]

TABLE 6.1: Formal implementation sources of community detection methods included in our analyses. The implementations of igraph tool (in R, Python or C/C++) can be founded at http://igraph.org/.

6.1 Preliminary analysis of detection methods

6.1.1 Computation time performance

Since computation time is a crucial factor to be considered in the selection of an algorithm, it is worth analyzing experimental performances to see how different community detection methods accomplish their task in real-world networks. By reusing the dataset summarized in Table 5.2, we proceed to assess official implementations of community detection methods introduced in Table 3.1. These implementations are provided officially either from their own authors or popular network analysis tools, which can be easily accessed from a large public. The corresponding sources of implementations which have been used are outlined in Table 6.1.

We employed all implementations stated above to identify community structures on all networks contained in the dataset and measured the time needed for each implementation to accomplish. The default parameters configured by the implementations are kept unchanged during the test. The calculations were executed on a server equipped by an Intel Xeon CPU E5-2650 with 32 cores of 2.60 GHz and a memory capacity of approximately 100 GBytes. However, due to the high complexity of some methods, only processes that finish in a practical amount of time (less than 4 hours) are taken into account. However, for a reference purpose, we let some of longer computations go on, for example, *Conclude* method took approximately 9 days to identify community structure on a network of 300 thousand vertices and 1 million edges; *GN* method did not finish its calculation for networks of more than

[1]Published at http://homes.sice.indiana.edu/filiradi/resources.html
[2]Published at https://sourceforge.net/projects/louvain/
[3]Published at http://www.tp.umu.se/~rosvall/code.html
[4]Published at http://www.mapequation.org/
[5]Published at http://www.oslom.org/
[6]Published at http://www-personal.umich.edu/~mejn/
[7]Published at https://sites.google.com/site/communitydetectionslpa/
[8]Published at http://www.emilio.ferrara.name/code/conclude/

4 thousand nodes and 40 thousand edges within 2 days. Consequently, the experiments that theoretically require too much time are neglected in the test. It is also worth noting that the calculations of communities on large-scale networks are also restrained by limited memory, therefore calculations that are supposed to be finished within 4 hours but required too much memory can not be shown here neither. We repeat the calculations 5 times on average for each pair graph/method to reduce the fluctuation impact. Eliminating all the cases that do not satisfy our requirements, the final successful rate (number of partitions identified over the number of possible tests) ended up at around 44.72%, mainly due to time/memory surpassing.

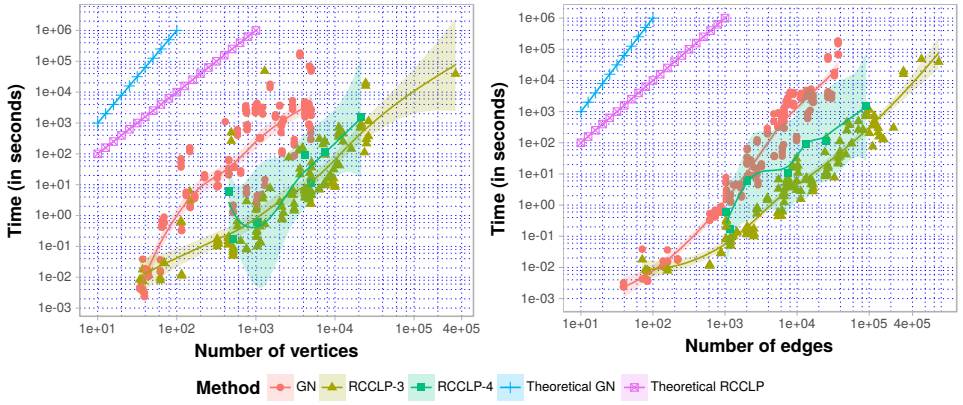


FIGURE 6.1: The execution time needed by GN, RCCLP-3 and RCCLP-4 methods to identify community structures on networks of the dataset.

In the following figures (from 6.1 to 6.6) that illustrate the analyses on experimental time consumption, some conventions are commonly used. Points in the figures correspond to separated executions that have been measured. The solid lines with the same corresponding colors to the points are estimated relations between computation time and network size (number of vertices and number of edges) using a local regression model (Cleveland, 1979). The dark colored backgrounds around the regression curves represent 95% confidence intervals of the model parameters. Besides, we show the worst case theoretical execution time (number of calculation needed in this case) of associated algorithms are included for a comparative reference purpose. From the analysis of structural characteristics of the dataset as shown in Figure 5.1(a), it is noticeable that most networks are sparse, i.e the number of edges (m) increase in a linear function by the number of nodes (n). Hence, in our estimate, we plot theoretical execution time by assigning $n = m$. For the simplicity of illustration, we grouped the measures of the methods by their approaches (Table 6.1).

The first group of methods consists in centrality detection techniques to identify community structure. As we can see in Figure 6.1, the *GN* method can not be accomplished in our test for networks of more than 4 thousand of nodes or 30 thousand of edges. The outcome is quite reasonable since the theoretical estimation for this method is $\mathcal{O}(nm^2)$, which grows quickly in function of network size. Remind that one of the primal purpose of the *RCCLP* method is to reduce the time complexity of the *GN* method. We can easily observe that this objective is achieved since the *RCCLP-3* reduces an order of around 10^3 times for graphs from 3 hundred nodes. *RCCLP-3* can well function with graphs up to millions of edges. However, when we proceeded the same test with *RCCLP-4*, the method rarely reached its terminus for

large graphs as well as small graphs. As we can see in the figure, there are few dots at the two sides. The reason is that there are not many (or even absent) 4-step close paths on real world networks. As it is not very probable that such structures exist in small graphs, finding them in large graphs also require a huge amount of time, *RCCLP-4* shows a poor performance in our test. Therefore, this configuration of the method is not recommended, as well as versions with $g > 4$ would logically poorly perform. It is also worth noticing that *RCCLP-3* and *RCCLP-4* are extremely memory consuming and are not suitable for limited resource devices. Finally, theoretical and practical time seem to find a consensus as the increments of time in function of network size are quite consistent in the three cases.

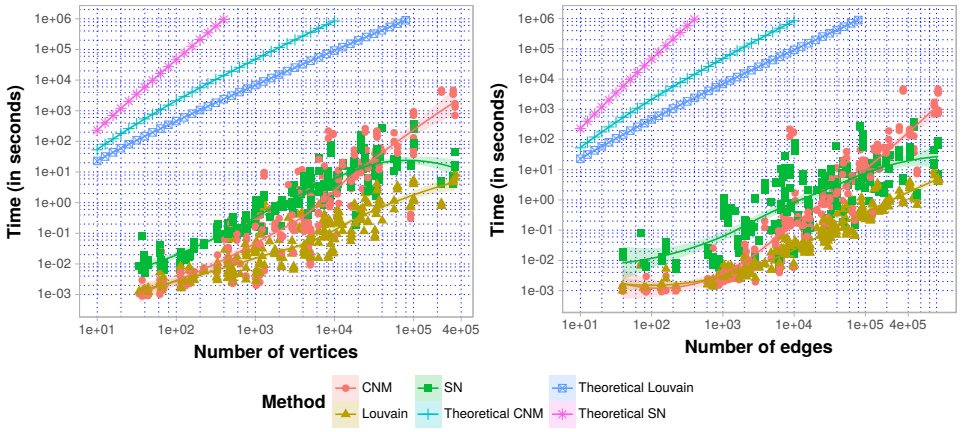


FIGURE 6.2: The execution time needed by CNM, Louvain and SN methods to identify community structures on networks of the dataset.

The next group includes methods using modularity optimization processes whose experimental measures are shown in Figure 6.2. Practically, the three methods in this family require a reasonable time for calculating community structures. The most time consumed experiment took less than 2 hours for a graph of 1 million edges. *Louvain* method is the fastest in this group whose computation increases approximately in linear time. It took only 9 seconds for the largest graph. Among the three methods, the optimization using spectral approach is the most expensive. However, all of these three methods have higher performance than the methods in the edge removal group previously stated. The experimental results also justify theoretical estimates about the complexity of these methods.

Similarly to the two previous group, the computation time needed by methods in the dynamic process group is illustrated in Figure 6.3. In terms of time consumption, this group shows a better performance with respect to the first group, but generally worse than the modularity optimization group (except for the Walktrap method for small and average size graphs). Among them, *Infomod* has the poorest performance. In the meanwhile, *Walktrap* and *Infomap* work asymptotically equally good with a slightly better rendition for *Walktrap* in small and average size graphs.

The same analyses for methods in the two final groups are shown in Figure 6.4 and Figure 6.5. We can easily see that *DCSBM* and *Oslom* have practically identical performance in terms of time consumption with a slightly less expensive on the side of *DCSBM*. In the last group, the results are quite discernible between different methods. The label propagation method *LPA* shows a clear distinctive curve indicating its out-performance over the other methods. Besides, *SLPA* works quite well, but less fast than *LPA* although it employs some special techniques to reduce the computation time (as mentioned in Section 3.2.7). This difference in the performance

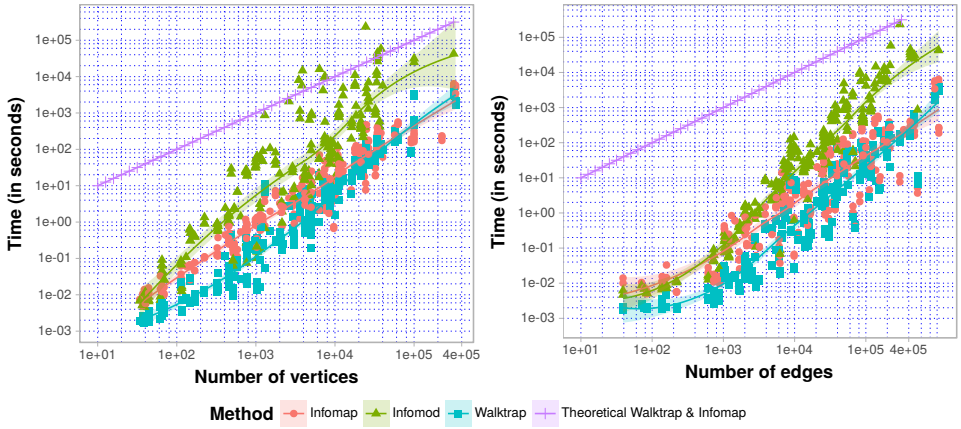


FIGURE 6.3: The execution time needed by Infomap, Infomod and Walktrap methods to identify community structures on networks of the dataset.

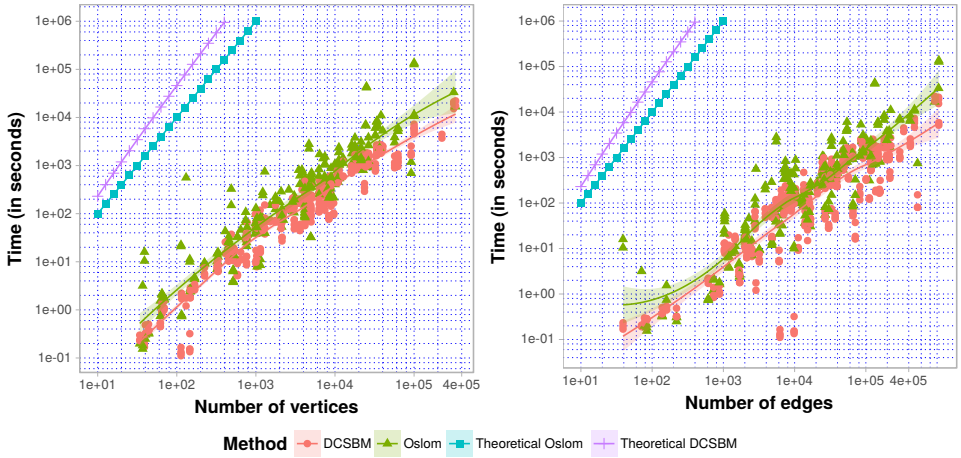


FIGURE 6.4: The execution time needed by DCSBM and Oslom methods to identify community structures on networks of the dataset.

is due to the more complicated mechanism that *SLPA* uses in comparison to *LPA*. The fact that *SLPA* has to reserve dedicated memories for all nodes of the network to stock the membership information that they received during the detection process and update them regularly to transfer into their neighbors makes it demanding. Therefore, despite of a 5 to 10 times of improvement in the label update strategy, the global performance can not surpass that of *LPA* method. In terms of scalability, *LPA* and *SLPA* seem to exhibit the same comportment which is nearly linear for small and medium graphs but accelerate in large graphs. The spin glass model *RB* manifests a better than expected presentation with an undeviating linear augmentation. The only unexpected behavior is spotted in *Conclude* method, as when the size of input graphs exceed some thousands, the required time has been inflated by a factor of n, making it very demanding for large graphs.

Finally, we aggregate all the analysis measures in the 5 previous groups into a common illustration as shown in Figure 6.6. At the same time, for a more convenient observation, we remove all the points corresponding to the experiments and keep only the regression curves, which are the estimates of execution time for these methods in function of number of vertices on the left hand side and number of edges

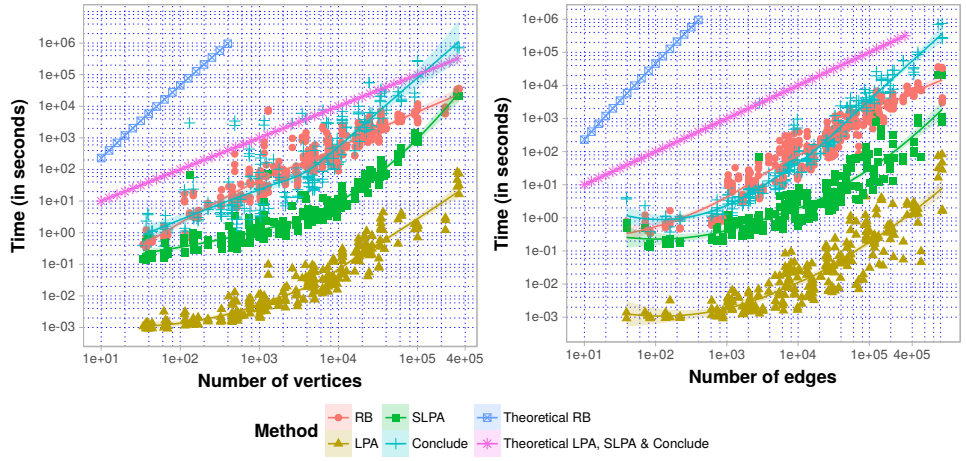


FIGURE 6.5: The execution time needed by RB, LPA, SLPA and Conclude methods to identify community structures on networks of the dataset.

on the right hand side. At a first sight, it is easy to see that except for *GN*, the necessary execution time for all other methods are limited in a range that increases polynomially in function of network size, which reflect well theoretical estimates. This range is upper-bounded by *Conclude*/*Oslom* and lower-bounded by *LPA* which corresponds to worst and best performed method(s) respectively. Another important information which can be deduced from this figure is that, for most real world networks of size in the range up to 1 million edges, choosing a fast detection method could economize an order of 10^3 times to 10^5 times calculation effort. This is an important element to be considered in applications where time consuming is a serious problem.

We demonstrate in Table 6.2 the ranking of these methods according to our test for reference purpose. *GN* and *RCCLP-4* are not involved in this ranking since they failed to accomplish their tasks in large graphs, which also means they are the most time consumed methods within the methods that we analyze. We show both the ranking by the average and the median of time. Since the average-time ranking is heavily affected by the measures on large graphs, i.e. methods that succeeded to discover communities on very large graphs are lower ranked than methods that were not able to do so. In these cases, the ranking by median is more accurate and it reflects well the relative performance on small and medium graphs between the methods. For large graphs, using the ranking by average would better fit.

6.1.2 Analysis on community size distribution

The number of latent communities that should be induced from a given network is one of the major question in community detection context (Fortunato and Hric, 2016), (Riolo et al., 2017), equivalent to the subject of the expected number of clusters in classical clustering problem. Observing the number of communities discloses useful information about the mesoscopic structure of a network. Specifically, the variation of the number of communities in a network implicates different level of resolutions, and according to the context, one would prefer to observe the modular structure of a network of interest regarding a certain resolution. An analogous way to describe the concept of resolution is the distance from an object that we prefer in

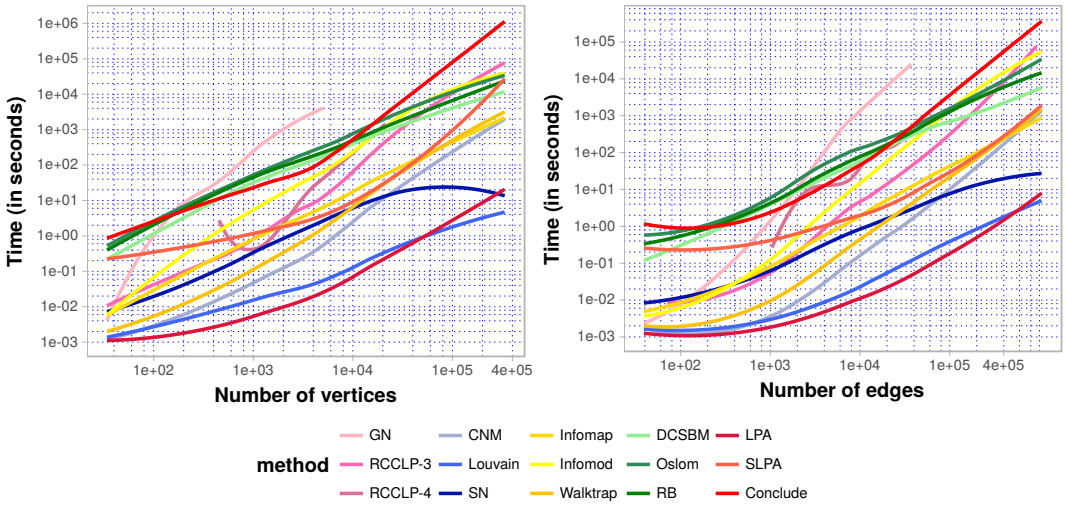


FIGURE 6.6: The estimated execution time needed for each method to identify community structures on networks of the dataset using a local regression model. Methods of the same theoretical family (in the same group) are represented by chromatically similar color.

order to contemplate it. The closer we get to an object, the more its detailed microstructures that could be perceived, in the meanwhile the less information about the global organization that tends to be clear. Although several multi-resolution approaches (Lambiotte, 2010), (Pons and Latapy, 2011) incorporating resolution parameters into their solutions to provide more flexible mechanisms and different modular scales of networks, it is not always obvious to regulate appropriately these parameters without ad-hoc cases. The inclusion of multi-resolutions parameters, of course, widen the possibility of understanding networks, but in the expense of the automatic aspect that is sometimes required in clustering problems (at least for neophytes - #smiling emoticons).

In this section, by using the same network corpus presented in the previous sections, we are motivated to evaluate this aspect of the mentioned detection methods. By keeping all default configurations of the implementations unchanged as previously done to ensure the consistency of future results, we proceed to explore the resolutions of these methods in a comparative way. From the antecedent analyses, some modifications will be applied on our testing process as follows:

1. From the observation of the network size distribution in Figure 5.1(a) as well as the previous computation time analyses, the linear relation between number of vertices and number of edges of networks in our corpus becomes evidenced. As a consequence, it will be redundant to address the relation of dependent variables in respect of these two latter predictors. Therefore, only analyses in function of number of vertices will be introduced.

2. In community detection problem, showing only the numbers of communities discovered on networks or their statistical derivatives would not always be enough. Assume that the sizes of communities in an arbitrary network follow a negative power-law distribution, its means that the number of communities depends heavily on the number of tiny communities. Therefore, we also observe the distribution of community size to discern the differences between methods which could not be recognized by seeing solely the number of blocks.

Method label	Rank by average	Rank by median	Scalability
RCCLP-3	9	8	Low
CNM	5	3	Medium
Louvain	1	2	High
SN	3	5	High
Walktrap	4	4	High
Infomod	12	9	Low
Infomap	6	7	Medium
Oslom	11	14	Low
DCSBM	8	12	Low
RB	10	13	Low
LPA	2	1	High
SLPA	7	6	Medium
Concude	13	11	Low

TABLE 6.2: Ranking of analyzed methods according to their amount of time consumed to identify community structure on networks of the dataset. Methods are ranked by average time required and median time. The scalability shows experimental result for the possibility of processing large scale graphs.

3. Due to a huge number of required calculations and a limited hardware resource, discovering processes in the last section were interrupted unless they are finished in a few hours. Here, some more efforts have been flexibly given if a method is supposed to be finished in a reasonable amount of time.

For a given network in the dataset, we applied all of the presented methods to identify the set of communities predicted by each one and measured their volumes. Similarly to the last part, for the simplicity of observation, we group methods by different families depending on their approaches. We illustrate the obtained results of community repartition measures in Figure 6.7 to 6.11 by using some conventions as follows:

Conventions for Figure 6.7 to Figure 6.11

1. A figure (denoted A) on the top contains three following sub-figures:

   1.1. The central figure (A1): shows a scatter plot about the distribution of communities in function of the number of vertices of the network to which they belong. The solid lines in the figure represent the estimated average community size in function of number of vertices using a local regression model (Cleveland, 1979). Dark colored backgrounds around the lines are 95% confidence intervals of the estimates.

   1.2. The top figure (A2): exhibits marginal density distributions of communities found in each range of network sizes. They are rendered from a Gaussian kernel estimator.

   1.3. The right figure(A3): illustrates another type of marginal density distributions of communities in function of their sizes. They are also rendered from a Gaussian kernel estimator.

2. A figure on the bottom (denoted B) presents the number of communities in function of the number of vertices of different networks as well as the estimated relation between these variables using the regression model stated above. Dark colored backgrounds around the lines shows 95% confidence intervals of the estimate relations.

Edge removal approach: GN, RCCLP-3 and RCCLP-4

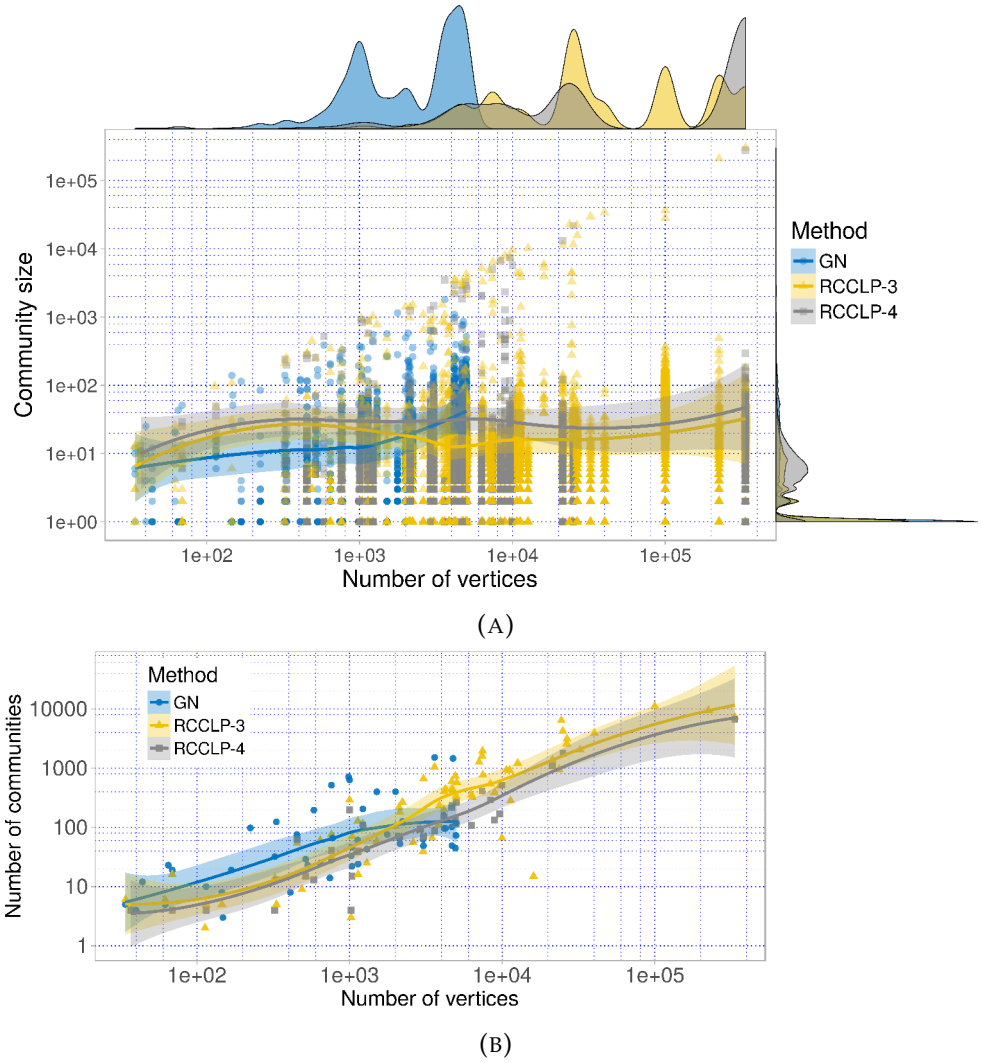


FIGURE 6.7: Fitting quality of GN, RCCLP-3 and RCCLP-4 methods on the networks of the dataset.

From Figure 6.7, we can notice again that *GN* method can only be able to function on small and medium networks due to its high complexity, which is quite obvious from theoretical analysis. *RCCLP-3* and *RCCLP-4* can detect up to the largest networks in our corpus. By observing the right marginal density distribution, surprisingly, all of these methods identify a huge number of singleton communities[9]. The average number of singleton communities is around 24% which can be up to 60% in some cases. The reason for this aberrant phenomenon is that in some dense and small networks, there exists too many high and equivalent central vertices and

[9]Communities that contains only one vertex

edges. The separating mechanism employed by this approach keeps removing central nodes or edges until a large number of vertices are isolated, creating singletons or very small communities. Since *GN* only works on small graphs, it is highly impacted by this phenomenon in our experiment. Besides, in a global observation, we can see in the top figure that the majority of communities detected by these methods are very small for the same reason. From Figure 6.7(A), we can see that a large number of communities have only less than 10 vertices even in very large networks which can be deduced from Figure 6.7(A1). This makes the number of communities increase rapidly as one can remark on Figure 6.7(B). Remind that the distributions of community size have right-skewed shapes, meaning that the majority of communities are small and most of them are found under the lines of average community sizes. Therefore, the three methods of this family have very high resolutions as observed on our experiment. Notwithstanding, this result need to be understand with caution due to two reasons:

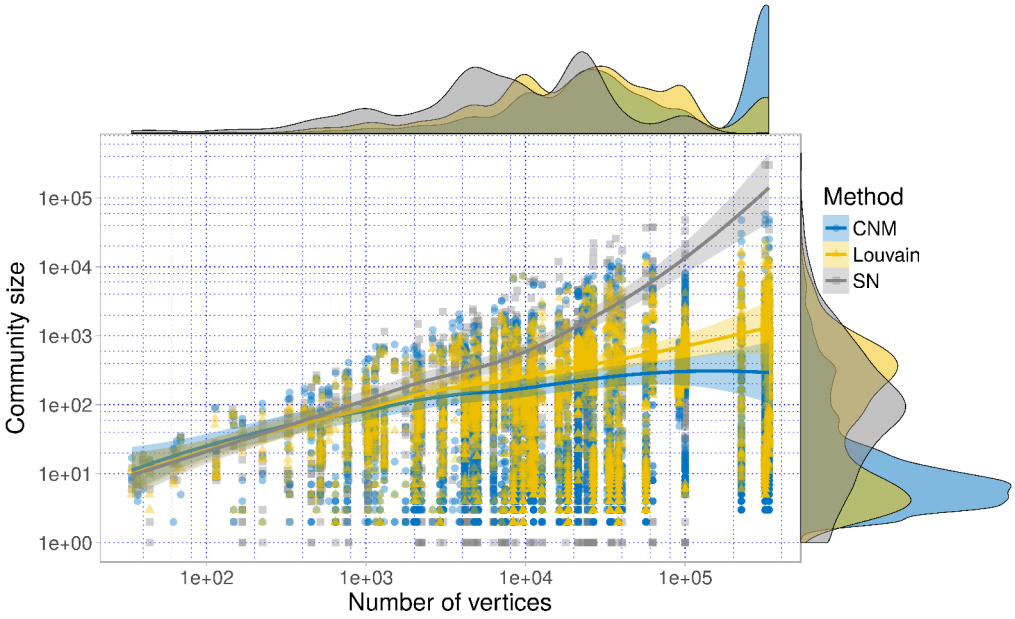
1. The density function in Figure 6.7(A1) reveals that the successful rate on discovering community structures of the three methods are distinguished fundamentally. In fact, due to the high complexity of time and memory, many networks are not successful resolved, which degrade importantly the comparison quality.

2. As a consequence of the first reason, there is a high fluctuation in the dependent variables which make the confidence intervals quite large. A deeper investigation on the quality on small and medium networks could partially palliate this problem.

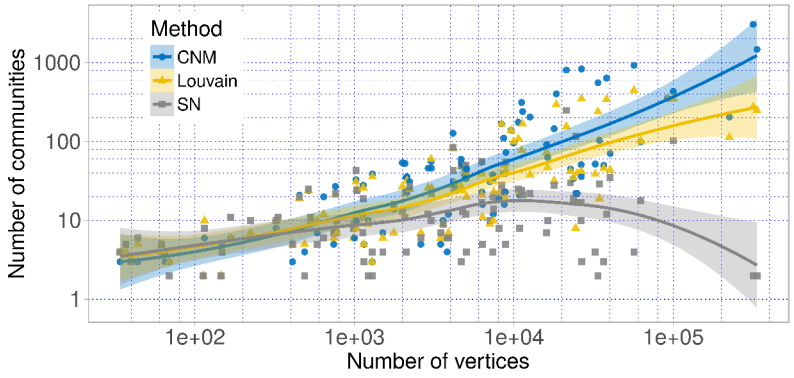
Although the previously mentioned issues, this class of methods remain the one which conjectures the highest number of communities with a great consensus. The following analyses will reinforce this remark.

Modularity optimization approach: CNM, Louvain and SN

On this second group, our measures are more complete since all three methods succeeded to resolve large networks. From Figure 6.8(A2), it can be seen that there is a regularity between the distributions of communities over the whole range of networks except for the range of very large networks. Actually, in this range, the behavior is very different with the three methods. While *CNM* determines a very large number of medium and small communities, *Louvain* identifies less small communities and more medium and large communities. On the other hand, *SN* only proposes a partition of two giant communities. For instance, if we take the *Amazon* network introduced in Table 4.2, while *CNM* detected 1480 clusters, this number becomes 249 for *Louvain* and only two for *SN*. The same phenomenon is also remarked for another example, the *DBLP* network which is also presented in Table 4.2, the corresponding numbers are 3077, 275 and 2 in the same order. This notice can also be remarked in smaller networks as can be seen in Figure 6.8(B), however gap between the number of communities reduces gradually from the right to the left of the figure. But in general, the order remain unaltered in our observation, i.e. the average number of communities detected by *CNM* is larger than that of *Louvain* which is in its turn larger than that of *SN*. Consequently, the order of community sizes are inversed since the sizes of graphs are fixed as can be seen in 6.8(A1). Another remark can be extracted from Figure 6.8(A3) about the diversity of community size,



(A)



(B)

FIGURE 6.8: Fitting quality of CNM, Louvain and SN methods on the networks of the dataset.

while *CNM* and *CN* consistently move towards small and medium communities respectively, *Louvain* on the other hand tends to propose both small and medium size communities.

Dynamic process approach: Infomap, Infomod and Walktrap

At a first glance, we can see a clear separation within the three methods. While *Infomap* and *Walktrap* display quite comparable evolution of average community size depicted by Figure 6.9(A1) as well as marginal distribution as depicted by Figure 6.9(A2-A3), *Infomod* is driven distinctly apart. Diving into the measures, we notice that in *Infomod*, there is a relatively uniform repartition of communities which is upper-bounded by the largest containing 6948 vertices. Unlike many other methods including *Infomap* and *Walktrap*, the number of medium and large communities discovered by *Infomod* does not outnumber the number of small communities as stipulated by heavy-tailed distributions. As a consequence, the total number of communities observed remains low and increases with a small constant pace.

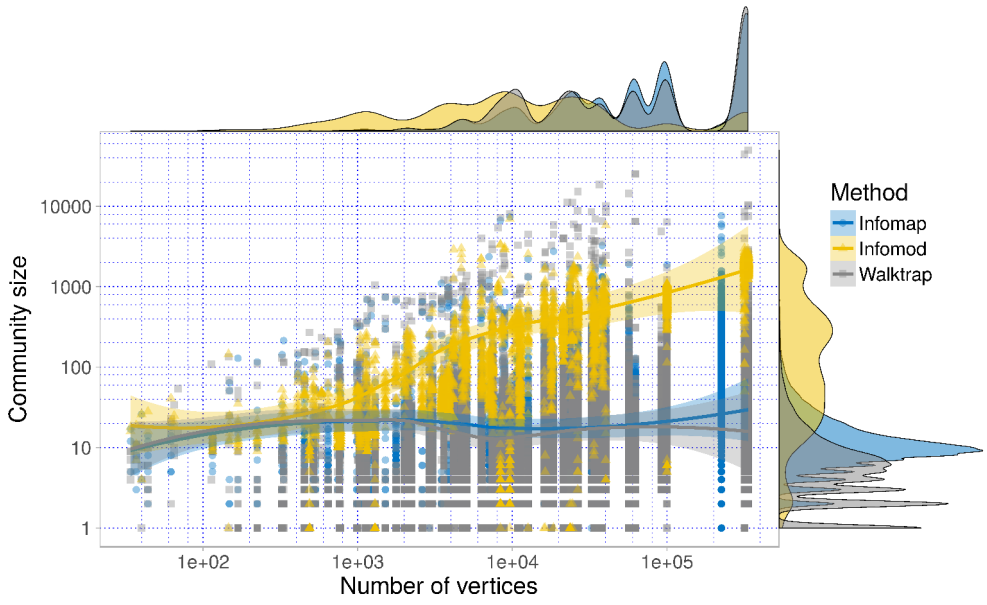
Infomap and *Walktrap* tend to keep their average community size limited around 10 to 30 over the whole range of networks. This phenomenon keeps them away from the resolution limit issue. In both methods, the most popular community size can be found around 10 nodes or smaller. Our more specific measure on the median community size shows almost similar results for *Infomap* while this number decreases slightly for *Walktrap*. Above these values, the number of communities decreases profoundly. The biggest difference between these two methods can be easily observed at the spurious region on the marginal distribution of Figure 6.9(A3). In fact, unlike *Infomap* which produces very moderately small communities, *Walktrap* identifies a huge number of isolated nodes (around 10% according to the statistics) and small communities similarly to *RCCLP-3* and *RCCLP-4*. This problem may be due to the agglomerative hierarchical clustering employed by *Walktrap* to detect communities which engenders orphaned peripheral vertices, which has been introduced in Section 3.2.1 and illustrated Figure 3.3. This problem, however, is quite simple to be palliated since these peripheral vertices could be assigned to their closest neighbor's community. By removing this issue, we have got a quite similar result for *Infomap* and *Walktrap*.

In terms of average number of communities, *Infomap* and *Walktrap* show practically the same behavior. The evolutions are nearly coincided over the whole range of networks with small confidence intervals, especially in the middle range. For medium and large networks, as seen in Figure 6.9(B), it is very likely that *Infomod* identify much less number of communities. In fact, more than 75% of *Infomod*'s partitions have less communities than those of the other two methods.

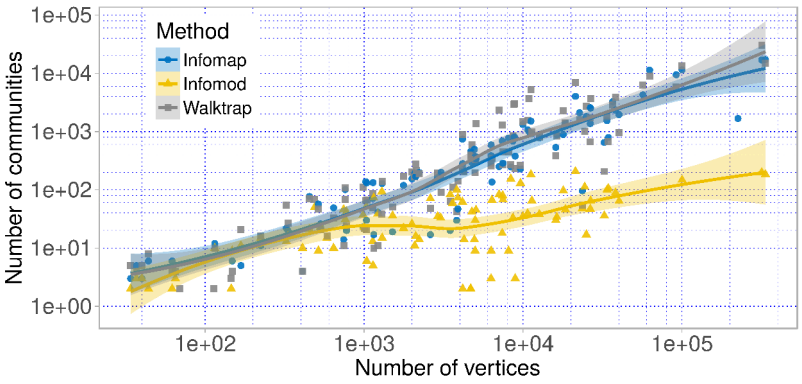
Statistical inference approach: SBM, DCSBM and Oslom

In the case of statistical inference, we see a quite similar phenomenon previously experienced in the dynamic approach. Specifically, the distributions of community size of the two implementations *SBM* and *DCSBM* are nearly coincided with a slightly higher average community size for the former. In fact, in this Bayesian block model, it is necessary that the prior distribution of number of block is given. According to different block model variants, one could assume various hypotheses about underlying mechanisms that create observed network under the corresponding regulations of block structures and define a prior probability. In the implementation that has been employed, the authors initialize the community discovering process by assigning nodes randomly to groups according to a queuing-type mechanism and then use a Monte Carlo sampling process to maximize the posteriori probability. However, the calculation becomes extremely time consuming when the maximum number of communities is too large (Riolo et al., 2017). Hence, by default, the maximum number of communities is configured at 25 as proposed, which leads to an underestimation of medium and large graphs as shown in Figure 6.10(B) as also be noticed by the authors. One can see the impact of this regulation as the number of communities approaches asymptotically 25 independently with the network size on the right hand side of the figure.

By observing the distribution of community size in Figure 6.10(A1), it is understandable that the average block size of *SBM* and *DCSBM* increases linearly in function of number of vertices. As the number of communities remains constant, the average community size must increase proportionately. Besides, the Figure 6.10(A3) also reveals that community sizes are well spread around their mean values, which makes the marginal distribution quite symmetric for both *SBM* and *DCSBM*. There



(A)



(B)

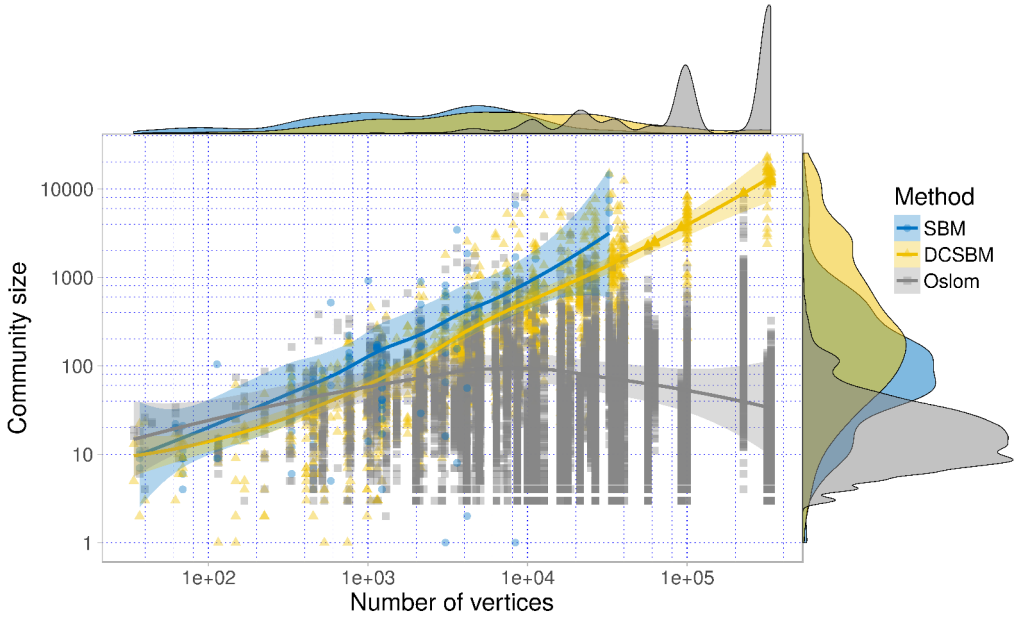
FIGURE 6.9: Fitting quality of Infomap, Infomod and Walktrap methods on the networks of the dataset.

is nearly no particular inclination towards small communities as acknowledged in some previous methods.

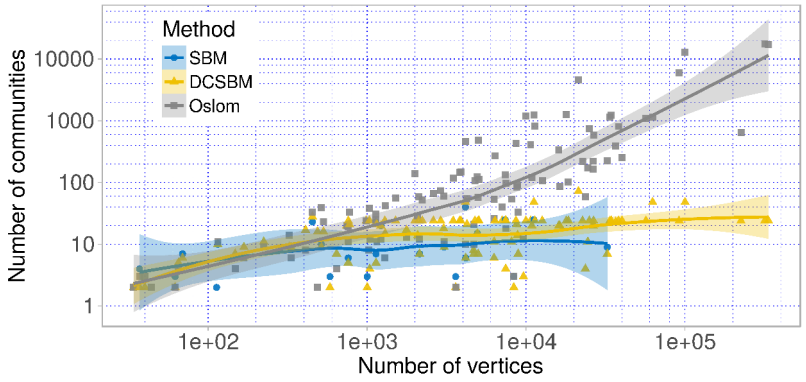
For the case of *Oslom*, the separation is quite clear. It unveils much more communities, making their sizes very small. Figure 6.10(A1) shows that the majority of *Oslom*'s communities are found under the average values of the associated partitions of *SBM* and *DCSBM*. Our demonstrations show that there is indeed a significant difference in the repartition strategies of these methods.

RB, LPA, SLPA and Conclude methods

In the last group, we discover that there is a remarkable coincidence in all distributions of the three methods *LPA*, *SLPA* and *Conclude*. In fact, the difference between them is nearly indistinguishable on the marginal measures. There is only a small discrepancy in the number of detected communities in very large networks as can be noticed from Figure 6.11(A2), such that *LPA* detected slightly more communities than *SLPA* and *Conclude*. From Figure 6.11(A3), one can see that the majority of



(A)



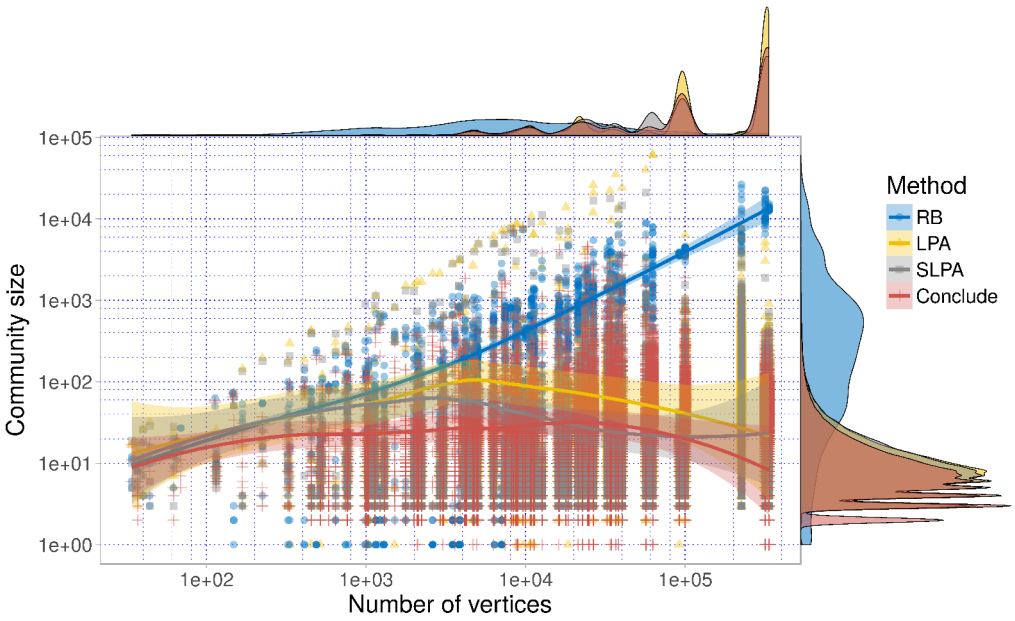
(B)

FIGURE 6.10: Fitting quality of SBM, DCSBM and Oslom methods on the networks of the dataset.

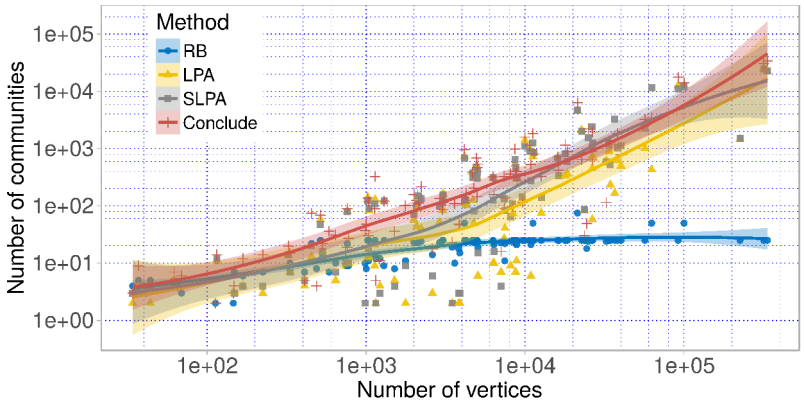
communities are quite small in these three methods. Similarly to *CNM*, *Infomap* or *Walktrap*, the majority of communities are small, i.e. have less than 10 nodes.

On the three methods, one could see that the variation of the data is significantly large, which produce also a large variation in our estimates. Since the associated prediction intervals for the estimates are likely to be larger, predictions related to community size distribution are not expected to be accurate.

On the other hand, *RB* method shows a solid consistency with much less variations in our examination. Average community size increases regularly and number of communities becomes saturated from medium size networks. The behavior of *RB* method is very resembling to that of *DCSBM* observed in Figure 6.9. Consequently, it is supposed to suffer the resolution limit for large networks. Notwithstanding, since *RB* is provided with a resolution tune parameter, the method may escape from this effect if the parameter is correctly chosen.



(A)



(B)

FIGURE 6.11: Fitting quality of RB, LPA, SLPA and Conclude methods on the networks of the dataset.

Summary

For the final step of this part, in the same manner as the previously presented time computational analysis, we put all methods in a comparative view. We aggregate the estimates of average community size and the number of detected communities in function of number of vertices in the network in Figure 6.12(A) and 6.12(B) respectively. One can see that there exists several repartition strategies hidden in these methods. If we use the preference of theoretical number of recoverable communities in a k-planted partition model (Ames, 2013), being $O(\sqrt{n})$, the studied methods could be considered to over-fit (create more than k clusters) or under-fit (create less than k clusters) as presented in Table 6.3, in the third column.

We can see that, in a general view from the second and third column of Table 6.3, methods belonging to the same theoretical class which shares a common assumption about the definition of community have a tendency to show the same fitting quality, as also discovered by (Ghasemian, Hosseinmardi, and Clauset, 2018). However,

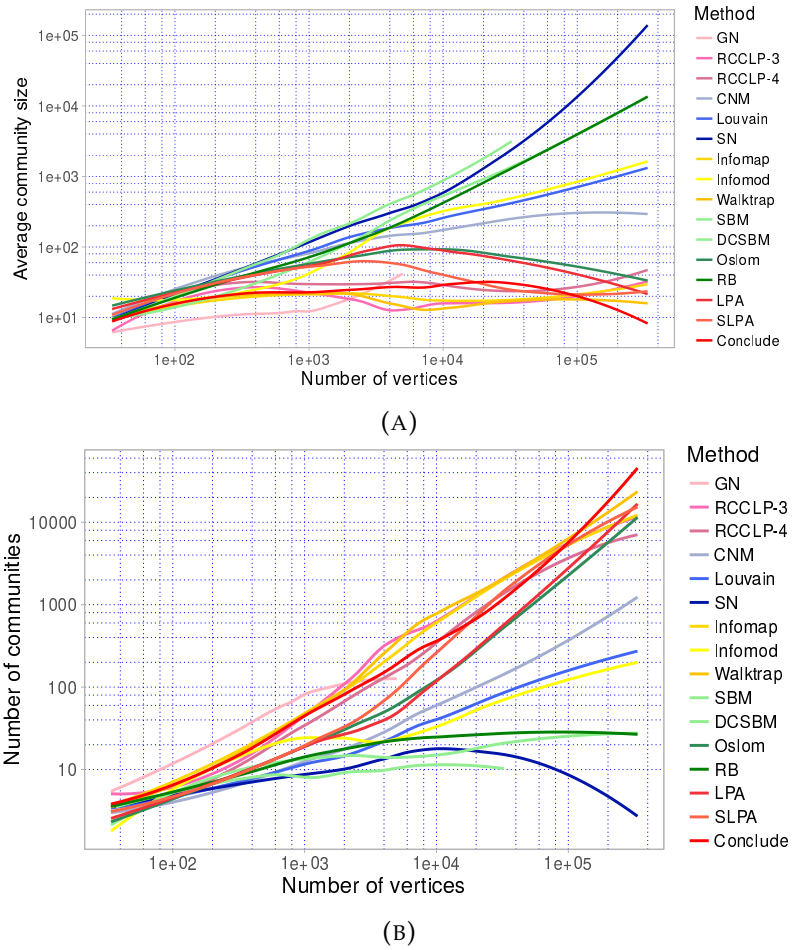


FIGURE 6.12: A summary of community size estimation quality

although being useful to help practitioners to presume the expected number of clusters a method would detect with respect to the theoretical experience, it is still very embarrassing to know which method to use since the reference is based on an hypothesis about an underlying model. This also means that if the hypothesis about the partition model change (another model than k-planted model), the expected number of communities will be diversified, and hence the indicated fitting quality preference becomes disproved. As a consequence, we propose a novel technique to estimate the similarity of community detection methods based on community size distributions in the next section. Certainly, this is only one among interesting quality aspects that differentiate one method from the others. Nonetheless, we will demonstrate that it also allows to get more insight into the difference in terms of partitioning strategy.

6.2 Similarity based on community size distributions

A very naive but efficient approach to evaluate the similarity of two methods is to inquire into the *"closeness"* of the two corresponding community size distributions. As such, two methods could be supposed to be similar if their corresponding density distributions expose a large intersection area as shown in Figure 6.13(A). From this notice, we can define our new similarity function as follows (Dao, Bothorel, and Lenca, 2018c):

Method label	Wrt. k-planted model	Fitting
GN	Bigger	Over-fit
RCCLP-3	Bigger	Over-fit
RCCLP-4	Bigger	Over-fit
CNM	Close	Over-fit
Louvain	Close	Under-fit
SN	Smaller	Under-fit
Walktrap	Bigger	Over-fit
Infomod	Close	Under-fit
Infomap	Bigger	Over-fit
Oslom	Smaller	Under-fit
SBM	Smaller	Under-fit
DCSBM	Smaller	Under-fit
RB	Smaller	Under-fit
LPA	Bigger	Over-fit
SLPA	Bigger	Over-fit
Concude	Bigger	Over-fit

TABLE 6.3: Ranking of analyzed methods according to their number of detected communities. A method is considered to over-fit if it detects asymptotically more than $\sqrt{n}$ clusters. The group numbers exhibit the estimated similarity based on fitting quality.

First, we denote two 2-tuples $(\mathcal{A}, n^a)$ and $(\mathcal{B}, n^b)$ being the multisets representing all communities detected on a set of networks $\mathcal{G} = \{G\}$ by method A and method B respectively, where $\mathcal{A} = \{x_1^a, x_2^a, ..., x_r^a\}$ and $\mathcal{B} = \{x_1^b, x_2^b, ..., x_s^b\}$ being the ascending ordered sets of sizes of communities: $1 \leq x_1^a < x_2^a < ... < x_r^a$ and $1 \leq x_1^b < x_2^b < ... < x_s^b$. The multiplicity functions $n^a : \mathcal{A} \to \mathbb{N}_{\geq 1}$ and $n^b : \mathcal{B} \to \mathbb{N}_{\geq 1}$ measure the number of communities of sizes x_i^a and x_i^b respectively. Let $N^a = \sum_{i=1}^{r} n^a(x_i^a)$ and $N^b = \sum_{i=1}^{s} n^b(x_i^b)$ being the total number of communities of all sizes detected by each method, we define a similarity function describing the closeness of A and B on $\mathcal{G}$ as:

$$S_{\mathcal{G}}(A, B) = \frac{1}{2} \sum_{i=1}^{r} \sum_{j=1}^{s} \min \left\{ \frac{n^a(x_i^a)}{N^a}, \frac{n^b(x_j^b)}{N^b} \right\} \delta(x_i^a, x_j^b), \tag{6.1}$$

where $\delta(x_i^a, x_j^b) = 1$ if $x_i^a = x_j^b$ and 0 otherwise. Equation (6.1) is simply the common fraction of same-size communities detected on $\mathcal{G}$ by both A and B: $0 \leq S_{\mathcal{G}}(A, B) \leq 1$. This definition seems to be intuitive but does not work well in practice. As illustrated in Figure 6.13(B), when the sizes interlace each other, a low score will be produced although the similarity in this case is as much as that of the case in Figure 6.13(A). Choosing an appropriate binning interval would mitigate the problem. This solution is, however very inflexible, sensible to the characteristic of data as well as to the functionality of the methods in use. A straightforward alternative can be envisioned by using a kernel density estimator to uncover the probability density function as shown by the solid lines in Figure 6.13(B). In this way, we approximate the common fraction of same-size communities of Equation (6.1) by the overlapping area of two corresponding continuous distributions. The premise behind this estimation is that two similar methods must not compulsorily produce a large portion of exactly same-size communities but rather a large portion of comparable-size ones. Hence, we consider the following estimator to take into account local information of

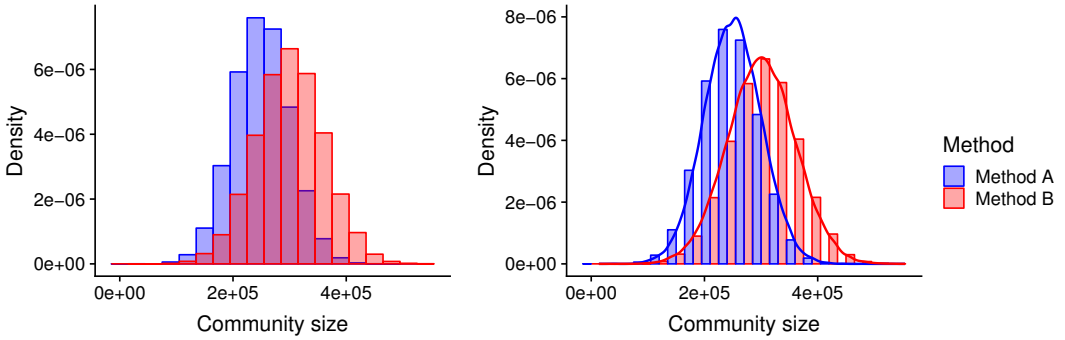


FIGURE 6.13: The distribution of sizes of communities detected by two different methods. On the left (A) overlap fraction using histogram, on the right (B) when community sizes interlace, the similarity is better estimated using a kernel density estimator.

community size x_0:

$$\widehat{f}(x_0) = \frac{1}{hn} \sum_i K\left(\frac{x_i - x_0}{h}\right), \tag{6.2}$$

where h is the bandwidth controlling the neighborhood interval around x_0 and K is the kernel function controlling the weight given to the observations $\{x_i\}$ chosen as Gaussian in our analysis. Using this estimator, we rewrite the similarity function defined in Equation (6.1) as follows:

$$S_{\mathcal{G}}(A, B) = \int \min\{\widehat{f}^{(a)}(x), \widehat{f}^{(b)}(x)\}dx, \tag{6.3}$$

where

$$\widehat{f}^{(u)}(x) = \frac{1}{hN^u} \sum_i^{N^u} \left[n^u(x_i^u) K\left(\frac{x_i^u - x}{h}\right)\right], \tag{6.4}$$

with $u \in \{a, b\}$. In the estimations of this paper, the bandwidth h is selected based on the normal reference rule (Silverman, 1986) to minimize the mean integrated squared error. The only exception is the cases illustrated Figure 6.14 where a higher value has been chosen to get a higher smoothing quality for a better illustration.

Using Equations (6.3) and (6.4) to estimate the similarity between pairs of detection methods on a large dataset will help us discovering different behaviors of community detection methods. Since the accuracy of the estimator depends on the networks of the dataset that we analyze, the result will have obviously to be relativized. However, our large and representative corpus would help to reduce the dependency impact.

6.2.1 Experimental results

From the communities identified in the previous section, we proceed to measure the volumes of communities detected by each method to determine the elements of the corresponding 2-tuples. Finally, we use the similarity function defined by Equation (6.3) to estimate the closeness between each pair of methods. Due to the huge number of experiments, only processes having a reasonable theoretical estimated time and memory consumption are maintained (less than a few days and at most 30 to 40 GBytes of memory). The outcome distributions are illustrated in Figure 6.14.

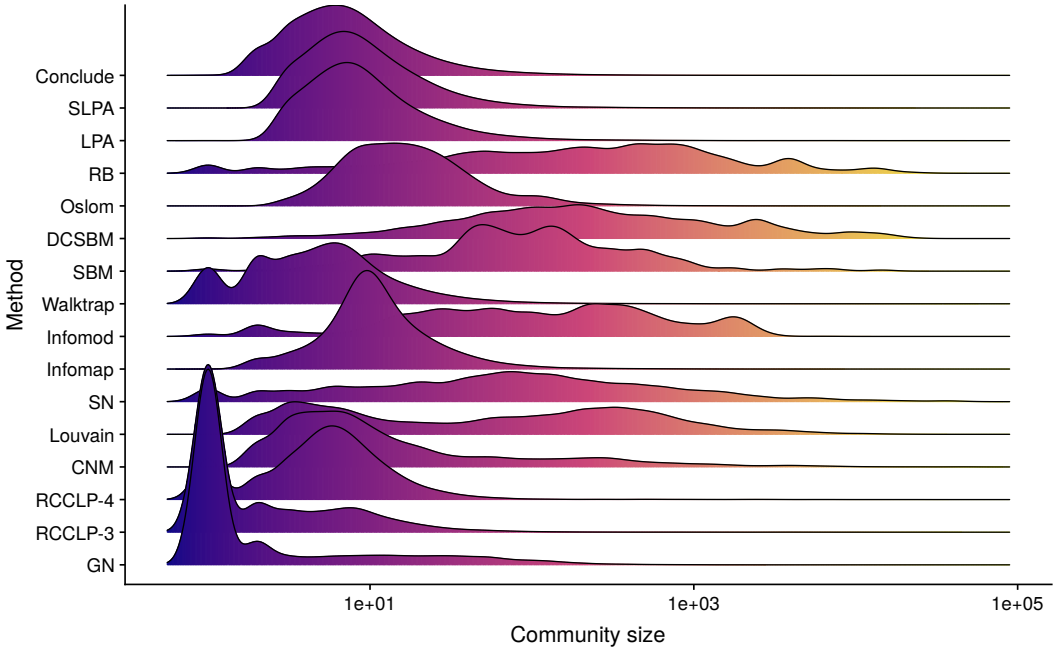


FIGURE 6.14: The distributions of communities by sizes contained in the partitions detected on the networks of the dataset. They are smooth using a Gaussian kernel estimator. The illustrative gradient color is only for the ease of view purpose.

As we can see, there is a clear difference in the densities of community size, showing that these methods have various partitioning strategies. Knowing that methods belonging to the same theoretical group (as shown in Table 6.1) are placed next to each other, we can notice some agreements between the theoretical families with practical outcomes as follows:

Edge removal: *GN* and *RCCLP-3* have very similar distributions where a large number of communities are very small. This is due to the fact that in some highly local centralized networks having star-like structures (as shown in the previous chapter), they have a tendency to remove edges connecting hub and peripheral nodes and create singletons (single node community). This phenomenon is less distinguishable on *RCCLP-4* since there are much less quadrangular than triangular connections in networks.

Modularity optimization: Modularity is known to suffer from resolution limit phenomenon (Fortunato and Barthelemy, 2006), which often aggregates small communities in large scale networks. We can see from Figure 6.14 that *Louvain* and *SN* found very large communities as predicted. In the meanwhile, there are also a comparable number of small communities which are found on small graphs. However, the behavior is a little bit different on *CNM* method, which is an agglomerative clustering algorithm based on modularity optimization.

Dynamic process: Methods in this family show very discernible distributions although all based on dynamic processes. In fact, they make different assumptions about community structure and searching mechanisms. Therefore, belong to the same theoretical family does not lead to a similarity in practical results.

Statistical inference: the Bayesian *SBM* and *DCSBM* uses Monte Carlo sampling process which is very time demanding in order to sweep the solution space. This makes the method unfeasible if the maximum number of clusters is not limited. Indeed, in the default version, the maximum number of communities is limited at 25 making *(DC)SBM* methods find very large communities in large networks. On the other hand, Oslom method use an agglomerative discovery mechanism and identify globally smaller communities.

Other methods: In this group, *LPA*, *SLPA* (both based on label propagation) and *Conclude* display nearly identical distributions. *RB* method, being based on a very close concept with modularity (with a tuning parameter), exhibits a similarity with modularity optimization based methods.

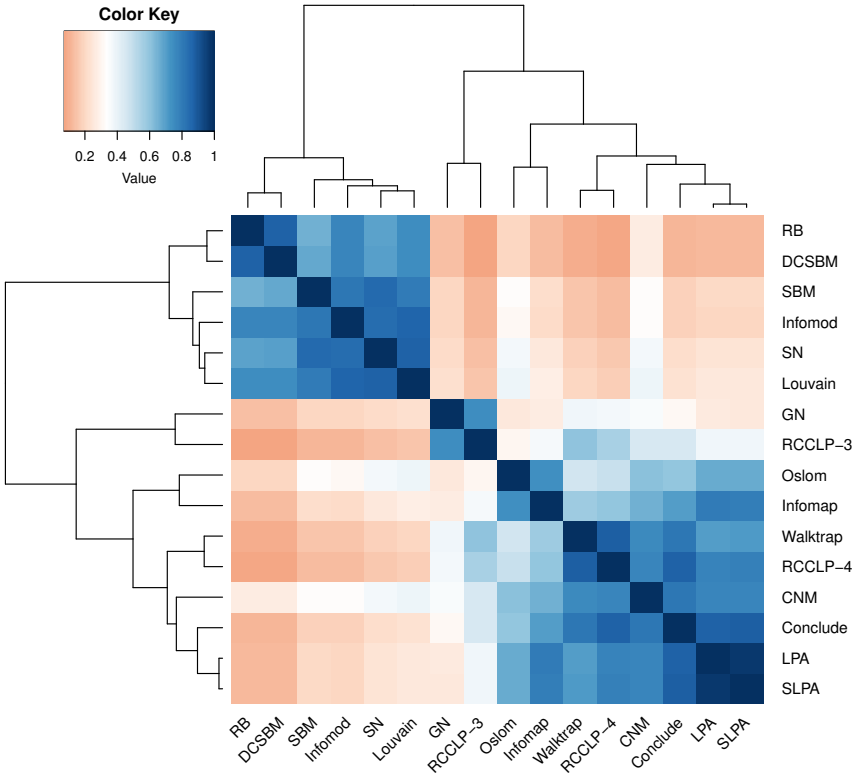


FIGURE 6.15: The similarity between community detection methods in term of size fitting quality. Two methods are considered to be similar if they share a large fraction of same-size communities. Methods are ordered using hierarchical clustering (Joe H. Ward, 1963). The dendrogram proposes a hierarchical structure of the fitting closeness. Blue colors mean high similarity.

Quantitatively, applying the estimator presented in Equation (6.4) to compute pairwise similarities between the methods leads us to the results demonstrated in Figure 6.15. As we can see, according to the community size criterion, these methods can be classified into different classes of partitioning strategy. The separations are very shaped showing that the distinction is very clear between groups. Therefore, we choose to characterize these methods by 3 (possibly 4) principle groups as follows:

1. **Group 1** - *RB*, *DCSBM*, *SBM*, *Infomod*, *SN*, *Louvain*: Methods in this group discover communities whose size vary in wide range of spectrum, from very

small to very large communities. The characterized community size distribution is quite flat, meaning all sizes are nearly equally considered.

2. **Group 2** - *GN* and *RCCLP-3*: These two methods identify a huge number of very small communities including singletons regardless of network size. As a consequence, there are few variations in community volume.

3. **Group 3** - the others: These methods produce communities whose sizes approach bell-shaped distribution. The strategy can be translated as: not left not right, i.e. not too small and not too big communities.

This characterization could help us to identify appropriate group of community detection methods according to different community size fitting strategies. Also, it helps to avoid brute-force tries when a method does not succeed to propose desired partitions by proposing substitute solutions. Moreover, by combining with the previous time computation analysis in Section 6.1.1, one could also choose a group of methods corresponding to size distinction criteria, and then select the fastest method that lead to a desired outcome.

The community distribution (or number of communities) is just one possible quality dimension, even it could possibility be the most important information when choosing a clustering method. In the next part, we demonstrate some techniques that can be used to define other similarity aspects. We show that these notions of similarity can be combined to accentuate the distinction between different community detection methods.

6.3 Detection performance profiling

6.3.1 Fitness functions

As mentioned in the previous chapter, a popular way to evaluate the structure of communities is to design quality goodness metrics in order to measure different expected characteristics from subgraphs that we want to obtain. In Chapter 5, some topological goodness metrics are employed to characterize popular interaction patterns between nodes in communities. However, in practice, other goodness metrics using network models are sometimes preferable when there is an assumption about the underlying generative mechanism. One of the most widely used metric of this class that quantifies the quality of community structure that has been mentioned all along Section 3.2 of Chapter 3 is the *modularity* function. The idea here is to reveal how the quality of identified community structures are different from what would be expected. Although some unexpected phenomena known as resolution limit (Fortunato and Barthelemy, 2006), (Traag, Dooren, and Nesterov, 2011) have been exposed when the scale of community size is too small, modularity remains to be the standard measure of quality.

The advantage of this approach is that one can *"embed"* the assumption of community structure inside quality functions, hence they provide better performance in some cases. However, community structure is quite an open question, such that according to different mechanisms that render the structure of networks, there will be models that are more suitable than others. Modeling networks hence contributes a great impact on the evaluation of network structure as well as community structure.

We present some quality metrics in this class to evaluate community structure. Many of them are initially or gradually employed as objective functions in some community detection methods since they expose good performance in searching

processes. By reusing the notations introduced in Chapter 4, we analyze the following functions:

- *Modularity*: The standard version of modularity (Newman and Girvan, 2004) reflects the difference the fraction of intra community edges of a partition with the expected number of such edges if distributed according to a *null model*. In the standard version of modularity, the null model preserves the expected degree sequence of the graph under consideration. In other words, the modularity compares the real network structure with a corresponding one where nodes are connected without any preference about their neighbors. There are several ways to mathematically express the modularity, in order to compare the standard modularity with other variants, it is convenient to consider the modularity as a sum of contributions from pairs of vertices of the same community:

$$Q_{NG}(P) = \frac{1}{m} \sum_{c \in P} \left[m_c - \frac{(2m_c + l_c)^2}{4m} \right] \tag{6.5}$$

- *ER Modularity*: The Newman-Girvan modularity has attracted much attention in the research literature. Many alternative derivations have been proposed to adapt to different contexts. Some of them use different null models to quantify the modular structure of partitions. For example, one could assume that vertices in a network are connected randomly with a constant probability p as formulated in the Erdős-Rényi (ER) model (Erdős and Rényi, 1959). The connection probability is calculated as $p = \frac{2m}{n(n-1)}$ being the number of presented edges over the total number of edges that could be established. The expected number of edges in a community of size n_c becomes $\langle m_c \rangle = p\binom{n_c}{2}$. This null model leads us to the ER Modularity:

$$Q_{ER}(P) = \frac{1}{m} \sum_{c \in P} \left[m_c - \frac{m n_c (n_c - 1)}{n(n-1)} \right] \tag{6.6}$$

- *Modularity Density (D-value or D-modularity)*: The standard modularity is found to be impacted by resolution limits (Fortunato and Barthelemy, 2006), i.e. it is claimed that the sizes of detected modules depend on the size of the whole network such that optimizing standard modularity can not identify communities having a small number of vertices. The expected number of intra community edges is highly sensitive to the total number of edges in the whole network (Rosvall and Bergstrom, 2007) as can be observed in the second term of Equation (6.5). The modularity density (Li et al., 2008) is one of several propositions that envisioned to palliate this issue. The idea of this metric is to include the information about community size into the expected density of community to avoid the negligence of small and dense communities. For each community C in partition P, it uses the *average modularity degree* calculated by $d(C) = d^{int}(C) - d^{ext}(C)$ where $d^{int}(C)$ and $d^{ext}(C)$ are the average internal and external degrees of C respectively to evaluate the fitness of C in its network. Finally, the modularity density can be calculated as follows:

$$Q_D(P) = \sum_{c \in P} \frac{1}{n_c} \left(\sum_{i \in c} k_{ic}^{int} - \sum_{i \in c} k_{ic}^{ext} \right) \tag{6.7}$$

- *Z-modularity*: This is another variant of the standard modularity proposed to avoid the resolution limit (Miyauchi and Kawase, 2016). The concept of this version is based on an observation that the difference between the fraction of edges inside communities and the expected number of such edges in a null model should not be considered as the only contribution to the final quality of community structure. Specifically, the authors recommend that the statistical rareness of a community should be also taken into consideration. Such that an additive contribution amount of a community to the final modularity of a partition would be more important if its structure is less likely to be happen. Therefore, the variance of the probability distribution of the fraction of the number of edges within each community is included into the quality function throughout a standardization using Z-score. Following the null model of the standard modularity, the probability that an edge in placed inside community C is $p = (\frac{D_C}{2m})^2$, where $D_C = 2m_C + l_C$ is the total degree of community C. The number of edges in each community follows a binomial distribution with the probability p and its normalized value approaches a normal distribution when the number of edges is sufficiently large. The statistical rarity of partition P in terms of the fraction of the number of intra-community edges using Z-score is hence translated into Z-modularity as follows:

$$Q_Z(P) = \left[\sum_{c\in P}\frac{m_c}{m} - \sum_{c\in P}\left(\frac{D_c}{2m}\right)^2\right]\left[\sum_{c\in P}\left(\frac{D_c}{2m}\right)^2\left(1-\sum_{c\in P}\left(\frac{D_c}{2m}\right)^2\right)\right]^{\frac{-1}{2}} \tag{6.8}$$

- *Surprise*: This statistical approach proposes a quality metric assuming that edges between vertices emerge randomly according to a hyper-geometric distribution (Aldecoa and Marín, 2011). Specifically, for a graph of n vertices and m edges, there are $M = \binom{n}{2}$ possible ways of drawing m edges. For a particular partition, there are $M^{int} = \sum_{C\in P}\binom{n_c}{2}$ possible ways of drawing an intra-community edge. Surprise metric computes the (minus logarithm of) probability of observing at least $m^{int} = \sum_{C\in P}\frac{k_C^{int}}{2}$ intra-community edges within m draws without replacement from the population of M possible choices in which consist precisely M^{int} possible intra-community edges. This probability is formalized as follows:

$$S(P) = -\log\sum_{k=m^{int}}^{\min(m,M^{int})}\frac{\binom{M^{int}}{k}\binom{M-M^{int}}{m-k}}{\binom{M}{m}}. \tag{6.9}$$

However, this formulation is not straightforward to work with in large-scale networks due to numerical computational problems. Hence, (Traag, Aldecoa, and Delvenne, 2015) provides an asymptotic approximation for the metric which is a good alternative. By assuming that the relative number of intra-community edges $q = \frac{m^{int}}{m}$ and the relative number of expected intra-community edges $\langle q\rangle = \frac{M^{int}}{M}$ remain fixed, Surprise metric is approximated at:

$$S(P) \approx mD(q||\langle q\rangle), \tag{6.10}$$

where $D(q||\langle q\rangle)$ is the Kullback–Leibler divergence (Kullback and Leibler, 1951):

$$D(q||\langle q\rangle) = p\log\frac{p}{\langle q\rangle} + (1-q)\log\frac{1-q}{1-\langle q\rangle}. \tag{6.11}$$

According to the Surprise metric, the higher the score of a partition, the less likely it is resulted from a random realization, the better the quality of the community structure.

- *Significance*: This metric use a similar approach to Surprise metric. It estimates how likely a partition of dense communities appear in a random graph (Traag, Aldecoa, and Delvenne, 2015). However, Significance estimates the unlikeness of dense communities in a random graph in a different way. While Surprise uses global quantities q and $1-\langle q\rangle$, Significance compares each community density $p_C = \frac{m_C}{\binom{n_C}{2}}$ to the average graph density $p = \frac{m}{M}$. The asymptotic form of Significance can be written as:

$$Z(P) = \sum_{C\in P}\binom{n_C}{2} D(p_C||p). \tag{6.12}$$

Similarly, $D(x||y)$ is the Kullback–Leibler divergence defined in Equation (6.11). Generally, if the number of communities is relatively large or the graph is relatively dense, Significance is more discriminative than Surprise. On the other hand, in case that $\langle q\rangle > p$, Surprise can be better than Significance (Traag, Aldecoa, and Delvenne, 2015).

6.3.2 Detection co-performance index

We devise a new comparative approach using a matrix called community detection *co-performance* matrix. The idea is that, given an expected quality function, one could investigate whether there exist a correlation in the efficiency of enhancing (or aggravating) its scores between different methods. The co-performance matrices reveal how understanding the performance of a method in optimizing a quality would allow us to predict the performance of other methods on the same quality. Therefore, an exhaustive analysis of co-performance matrices on many qualities allows to profile the characteristics of community detection methods in a comparative way. The index could be calculated as follows:

Let methods A and B divide a graph $G_i = (V_i, E_i)$ of dataset $\mathcal{G} = \{G_i | i = 1..N\}$ into α and β communities described by partitions $P^a_{G_i} = \{C^a_{1G_i}, C^a_{2G_i}, ..., C^a_{\alpha G_i}\} \in \mathcal{P}_{G_i}$ and $P^b_{G_i} = \{C^b_{1G_i}, C^b_{2G_i}, ..., C^b_{\beta G_i}\} \in \mathcal{P}_{G_i}$ respectively, we consider solely hard clustering methods, meaning $C^a_{uG_i} \cap C^a_{vG_i} = \varnothing : 1 \le u < v \le \alpha$ and $C^b_{uG_i} \cap C^b_{vG_i} = \varnothing : 1 \le u < v \le \beta$. A function $Q : \mathcal{P}_{G_i} \to \mathbb{R}$ quantifies a quality of a partition of graph G_i according to a particular goodness aspect (or model).

We define a *co-performance index* of two methods A and B on $\mathcal{G}$ by their mutual capacity in discovering community structures showing a particular quality Q. In other words, each couple of methods should be assigned a high index according to a quality Q if knowing the performance of one method reveals significantly the information about the performance of the other. A straightforward solution for defining the index is using Pearson correlation. Denoting $q^a_i = Q(P^a_{G_i})$ and $q^b_i = Q(P^b_{G_i})$, the co-performance index can be calculated as follows:

$$I_{\mathcal{G}}(A, B, Q) = \frac{N\sum q^a_i q^b_i - \sum q^a_i \sum q^b_i}{[N\sum (q^a_i)^2 - (\sum q^a_i)^2]^{1/2}[N\sum (q^b_i)^2 - (\sum q^b_i)^2]^{1/2}}, \tag{6.13}$$

where $0 \leq I_{\mathcal{G}}(A, B, Q) \leq 1$. A high positive (negative) score implies that two methods often find a strong consensus (disagreement) in discovering communities having a particular quality. In order words, given a co-performance index, knowing the quality scores of one method could provide predictive information about the outcomes of the other method on the same dataset. This information in fact could be very useful in a context where alternative solutions must be deployed while maintaining an assumed quality is expected. We present in the following part the mutual performance of the presented detection methods by the previously presented quality functions.

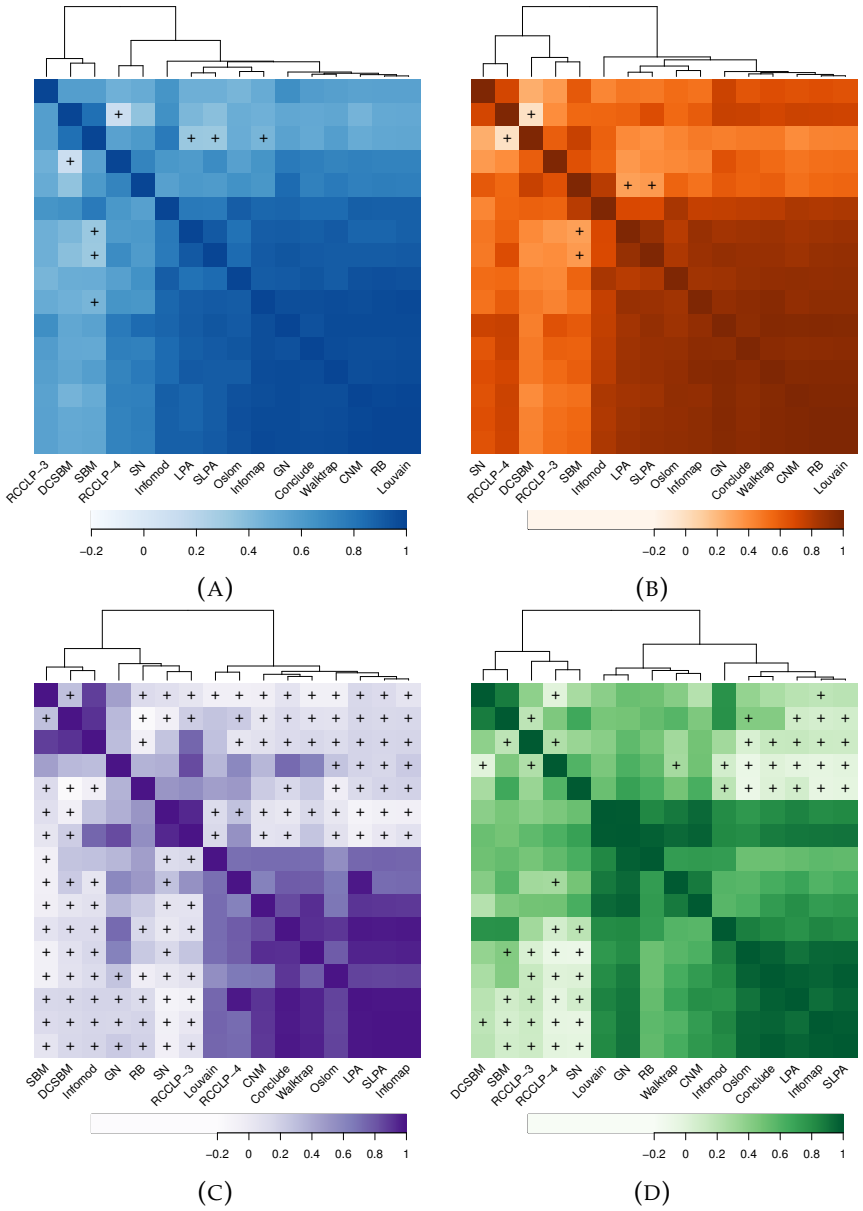
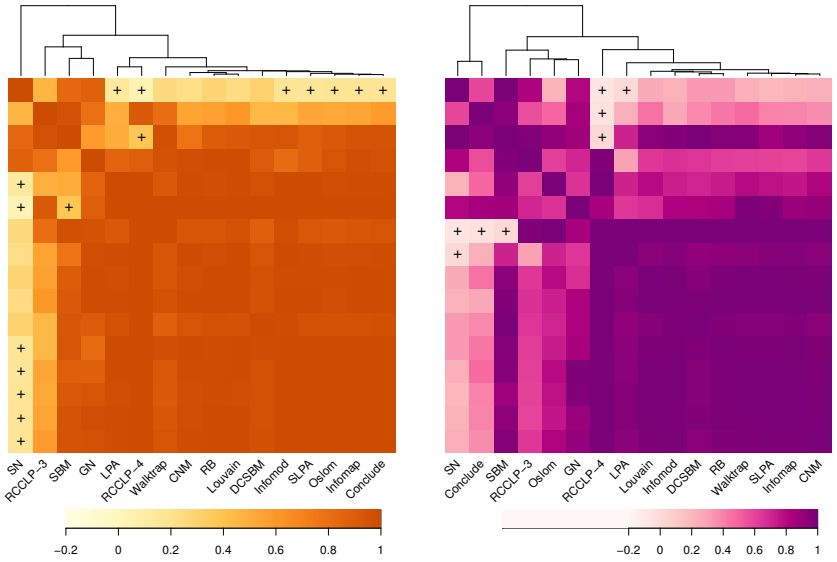


FIGURE 6.16: The *co-performance* matrices of different methods. The "+" marks indicate cases where p-values are larger than 0.05. (A) Newman-Girvan modularity, (B) Erdős-Rényi modularity, (C) Density modularity and (D) Z modularity.

Figure 6.16 illustrates the co-performance matrices according to six different quality goodness criteria. Again, similarly to the previous section, goodness functions



(E) (F)

FIGURE 6.16: The *co-performance* matrices of different methods (cont.)
(D) Surprise and (E) Significance.

with a *close* concept are placed together. For instance, *NG* modularity and *ER* modularity are both based on null models whose concept use an expected fraction of intra-community edges. While the hypothesis of *NG* version is to keep the expected degree sequence of the graph in question, the *ER* version redistributes edges randomly with a constant average degree for every nodes. *D*-modularity and *Z*-modularity attempt to penalize large communities by including community sizes and significance level respectively. One can notice a very slight similarity in the experimental results of the co-performance indexes between different quality functions. Also, it seems that the assumption about the quality of community structure has an impact on the co-performance outcome.

As shown in Figure 6.16, there is a class of methods (*Louvain, GN, CNM, RB, Infomod, Infomap, Walktrap, Oslom, LPA, SLPA, Conclude*) in which all methods show very consistent results, except for the case of *D*-modularity[10]. Besides, there is also a strong relation between *SBM* and *DCSBM*. For the other methods (*RCCLP* and *SN*), no clear tendency could be observed from this experiment. The similarity of a large number of methods by many quality functions imply that, globally, if a method performs well on a given network, there is a signal that the others (from the same group) could also reach good results. In other words, if the community structure in a network is clear, most method will be able to detect it with more or less accuracy and inversely. This is not contradictory with the conclusion stated in the last chapter, indicating that some methods could be better in improving some qualities. As the co-performance indexes also vary significantly (0.2 to 0.3) inside the groups, there will be always a remarkable difference if one go from on method to another.

Within the case of density modularity shown in Figure 6.16(C), we discover that the sizes of detected communities have a great impact on the co-performance. Since density is a measure that penalizes heavily large size communities, especially in sparse networks, *D*-modularity gives very small values of giant communities and

[10] In fact, density modularity is somehow apart from other traditional ways to define the modularity, as it is not defined based on a null model but solely on edge density. The term *D*-modularity is abused in this sense

very high values for small ones. Concretely, the methods *SBM*, *DCSBM*, *Infomod*, *RB*, *SN* discover very large communities (as shown in Section 6.2) and their co-performances in terms of *D*-modularity are very weak, showing that internal densities of communities detected by these methods are not linearly correlated. The reason is that the corresponding densities fluctuate unpredictably around zero. Similarly, *GN* and *RCCLP-3* found many tiny communities making the density either very high or zero (if internal degree is equal to external degree), consequently the co-performance index can not show significant information. On the other hand, we notice a consistency between the similarity of community size and the co-performance when methods identify medium size communities. Specifically, we find high co-performance indexes between *CNM, Conclude, Oslom, Walktrap, LPA, SLPA, Infomap* methods in most of the cases of the six quality fitness functions. This finding exposes a global agreement with our categorization determined by community size distributions.

The co-performance matrices also disclose interesting information about quality functions. As we can see in Figure 6.16(A,B,D), the matrices imply a similarity between *NG* modularity, *ER* modularity and *Z*-modularity in the assumptions of quality. In the same way, Surprise and Significance are quite close in practice as illustrated in 6.16(D,E). This experiment shows again another proof about the closeness between the theoretical assumption of community structure and the practical outcome. Moreover, although being based on different aspects of goodness, the performance of many methods tends to reach agreement on the modular structure of networks in general. This is to say, methods in the same group identify roughly and globally comparable results although there are always significance differences. In order to strengthen and validate our conclusion, we are interested in using other popular approaches in the literature to compare these community detection algorithms, which will be presented in the next section.

6.4 Evaluation using validation metrics

This section is dedicated to using conventional clustering validation metrics from the literature to verify the previous similarity analyses. We employ some popular metrics in the traditional clustering context (and also widely used in community detection context), which measure directly the likeliness of partitions using their corresponding contingency tables. These metrics do not take into consideration the structural information of community structures, but only use the common numbers of nodes that are shared by pairs of communities in two partitions.

6.4.1 Validation metrics

The consensus of two partitions $P_1 = \{c_1^{(1)}, c_2^{(1)}, ..., c_R^{(1)}\}$ and $P_2 = \{c_1^{(2)}, c_2^{(2)}, ..., c_S^{(2)}\}$ can be more practically observed using a contingency table (sometimes called confusion matrix or association matrix) whose elements $n_{ij} = |c_i^{(1)} \cap c_j^{(2)}|$ corresponds to the number of common vertices between the i-th community of P_1 and the j-th community of P_2 as shown in Table 6.4.

In the evaluation of community structure using a validation metric, some following validation metrics are often used in the context of community detection to define the matching coefficient between two arbitrary partitions of a network:

		Partition P_2				
		$c_1^{(2)}$	$c_2^{(2)}$	$\cdots$	$c_S^{(2)}$	$\sum$
Partition P_1	$c_1^{(1)}$	n_{11}	n_{12}	$\cdots$	n_{1S}	$n_{1\cdot}$
	$c_2^{(1)}$	n_{21}	n_{22}	$\cdots$	n_{2S}	$n_{2\cdot}$
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	$c_R^{(1)}$	n_{R1}	n_{R2}	$\cdots$	n_{RS}	$n_{R\cdot}$
	$\sum$	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot S}$	n

TABLE 6.4: Contingency table of P_1 and P_2 on the same graph provides information about the similarity between the two partitions.

Rand Index (RI)

The rand index is a pair-counting based measure, defined as the ratio of the number of vertex pairs correctly classified (either in the same community or in different communities) by the total number of pairs (Rand, 1971). The RI penalizes both false positive and false negative decisions of the clusterings. When the false positive need to be neglected, we can refer to the *Jaccard index* (Kuncheva and Hadjitodorov, 2004). The rand index value of two partitions can be calculated by the following:

$$RI(P_1, P_2) = \frac{\binom{n}{2} + 2\sum_i \sum_j \binom{n_{ij}}{2} - \left[\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}\right]}{\binom{n}{2}} \tag{6.14}$$

The value varies between 1 (meaning the two partitions are identical) and 0 (indicating that the two partitions do not agree on any pair of vertices). However, this value is only observed in the scenario when one partition consists in one community and the other consists in n community of 1 vertex, which has little practical value. Another shortcoming of the rand index is that its expected value for two randomly chosen partitions does not take a constant value which is normally expected for a good matching index (Vinh, Epps, and Bailey, 2010). Therefore, a modified version of RI has been suggested, taking into consideration the expected value of randomness (Hubert and Arabie, 1985), which is introduced in the following.

Adjusted Rand Index (ARI)

The corrected version of rand index takes the form:

$$Adjust\_index = \frac{Index - Expected\_Index}{Max\_Index - Expected\_Index} \tag{6.15}$$

It quickly becomes a replacement recommended for measuring agreement between two partitions in the analysis of clusterings. Its values ranges from -1 to 1 indicating completely different and identical partitions respectively. It is known to be less sensitive to the difference of the number of communities between two partitions. An ARI value of 0 indicates that the similarity is equal to the expected value from randomly chosen partitions. It can be calculated as:

$$ARI(P_1, P_2) = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}\right] - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{n}{2}} \tag{6.16}$$

Normalized Mutual Information (NMI)

Information theoretic based metrics constitute another approach for validating community structure with a given reference partition. Using the same notations as previously presented, the Mutual Information (MI) between two partitions quantifying the mutual dependence is calculated as:

$$I(P_1, P_2) = \sum_{ij} p(c_i^{(1)}, c_j^{(2)}) \log \frac{p(c_i^{(1)}, c_j^{(2)})}{p(c_i^{(1)}) p(c_j^{(2)})} = \sum_{ij} \frac{n_{ij}}{n} \log \frac{n_{ij} n}{n_{i.} n_{.j}} \tag{6.17}$$

It measures how much knowing a repartition of vertices in one way would reduce the uncertainty about the other way. In order words, it could be considered as an indicator of information *closeness* expressing by the joint distribution between two variables. Therefore, the mutual information can be used as similarity measure between two partitions. However, it need to be normalized to reflect a consistency between different measures. The normalization is applied by using the entropy of each partition as:

$$H(P) = -\sum_{k} \frac{n_k}{n} \log \frac{n_k}{n} \tag{6.18}$$

Several variants of normalization can be considered, for instance taking the average, the root or the maximum of entropy of the two partitions as the denominator (Ana and Jain, 2003). In this document, we use the average version which is widely used in the context of community analysis (Danon et al., 2005), (Chakraborty et al., 2017). The closed form of NMI is hence defined from Equation (6.17) and (6.18) as follows:

$$NMI(P_1, P_2) = \frac{2I(P_1, P_2)}{H(P_1) + H(P_2)} = \frac{-2 \sum_{ij} n_{ij} \log \left(\frac{n_{ij} n}{n_i n_j} \right)}{\sum_i n_i \log \left(\frac{n_i}{n} \right) + \sum_j n_j \log \left(\frac{n_j}{n} \right)} \tag{6.19}$$

Likewise, the NMI similarity between two partitions varies between 0 corresponding to independent relation and 1 when two partitions are identical. The NMI does not follow triangle inequality.

Adjusted Mutual Information (AMI)

Similarly to the Rand Index, the Mutual Information is also subject to the effect of randomness, i.e. there is not a constant baseline value between random partitions of a graph. This issue raises many difficulties in the comparison mechanism since it is expected that a comparative index should preserve the relativity between different clusterings and enhance intuitiveness about the mutual agreement. For that reason, the traditional Mutual Information is proposed to be normalized with a supplementary correction for chance and recently attracted attentions for comparing graph partitions. It is calculated as follows (Vinh, Epps, and Bailey, 2010):

$$AMI(P_1, P_2) = \frac{I(P_1, P_2) - E\{I(M)|n_{i.} n_{.j}\}}{\frac{1}{2}(H(P_1) + H(P_2)) - E\{I(M)|n_{i.} n_{.j}\}}, \tag{6.20}$$

where $I(P_1, P_2)$ and $H(P)$ are introduced in Equation (6.17) and (6.18) respectively. $E\{I(M)|n_{i.} n_{.j}\}$ is the expected mutual information value of all feasible contingency tables constructed from the actual table M with the same marginals $n_{i.}$, $n_{.j}$.

Normalized Variation of Information (NVI)

Another popular metric that is often used in the context of comparing community partition similarity is the Variation of Information (VI) (Meilă, 2003), which is defined as:

$$VI(P_1, P_2) = H(P_1) + H(P_2) - 2I(P_1, P_2) \tag{6.21}$$

The VI metric can be interpreted as an index of shared information distance between two partitions. Its lower bound is 0 and is occurred when the two partitions are identical whether the upper bound $\log(n)$ happens when they are completely different. It is also preferable to use a normalized version with chance corrected to avoid the effect of randomness. Similarly to the construction of the Adjusted Mutual Information, with the same notation, the Normalized Variation of Information is calculated as follows:

$$NVI(P_1, P_2) = \frac{H(P_1) + H(P_2) - 2I(P_1, P_2)}{H(P_1) + H(P_2) - 2E\{I(M)|n_i.n_{.j}\}}. \tag{6.22}$$

However, it turns out that NVI discloses the same information with AMI since from Equation (6.20) and (6.22), one has $NVI(P_1, P_2) = 1 - AMI(P_1, P_2)$. By consequent, calculating VI and NVI is unnecessary. We will be interested in using RI, ARI, NMI and AMI in our experiment. A summary of these validation metrics are shown in Table 6.5.

Label	Range	Measure
RI	$[0, 1]$	Fraction of commonly grouped and separated vertices in two partitions.
ARI	$[0, 1]$	Rand index with a chance correction, less sensitive to differences of community sizes.
NMI	$[0, 1]$	Information theoretic approach, indicate how much information knowing one partition will help to guess the other.
AMI	$[0, 1]$	Similar with NMI, with a chance correction to set a constant baseline for random partitions.
VI	$[0, \log(n)]$	Shared information distance measures the amount of mutual information. The higher the value, the less resembling the two partitions.
NVI	$[0, 1]$	Normalized version of shared information distance with chance correction.

TABLE 6.5: Some popular validation metrics for comparing community partitions

Validation metrics are often used in the context of community structure evaluation to measure the difference between the partition identified by a method with an expected partition of the network under consideration (*ground-truth*). The more similar the discovered partition to the ground-truth, the higher the performance of the method. However, in this section, validation metrics are exploited as a tool to compare community structures of different detection methods. They estimate the practical proximity of different algorithms through detected partitions, which constitutes a supplement source of information for evaluating their performance in a comparative approach.

6.4.2 Empirical results

Once again, the experimental process is the same as those of the previous sections. From the partitions detected by the methods on the dataset, we calculate pairwise scores quantified by each validation metric on each network. Figure 6.17 illustrates pairwise average scores of the 4 metrics over the networks of the dataset[11].

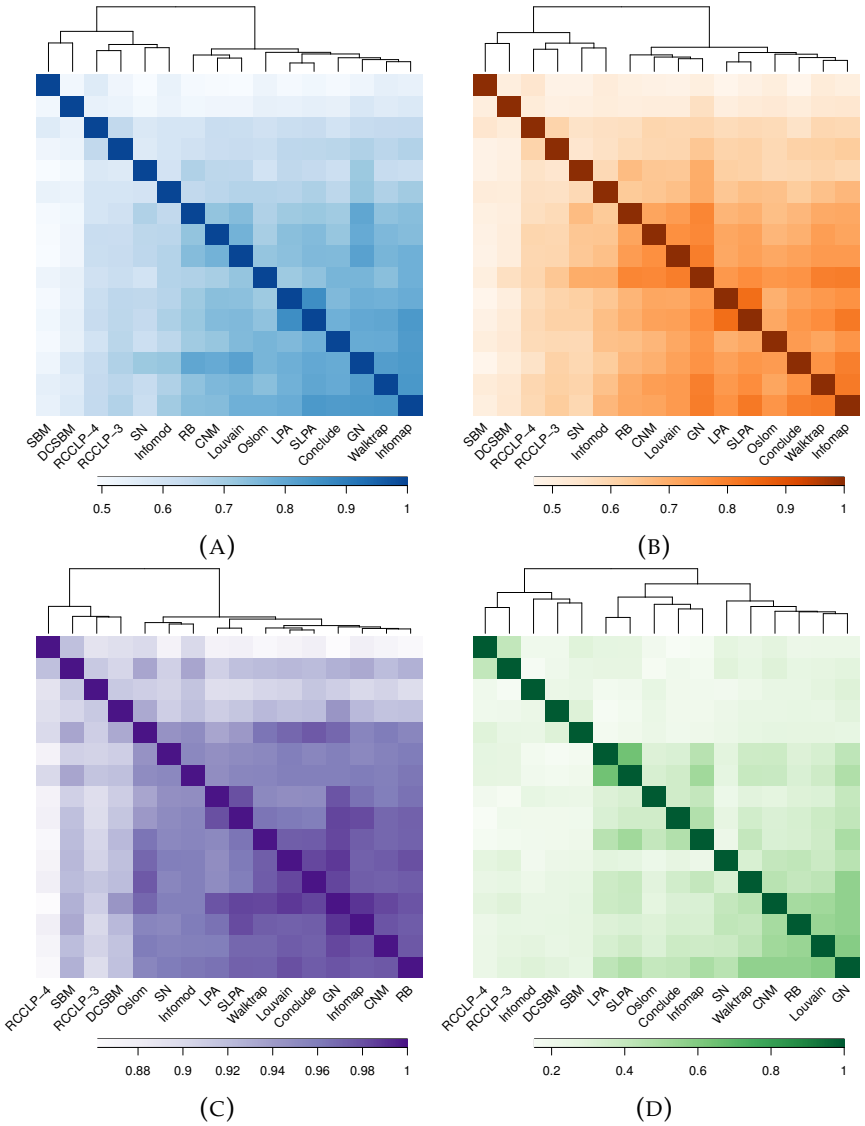


FIGURE 6.17: The similarity between community detection methods quantified by different validation metrics based on partitions discovered on networks of the dataset. Rows and columns are ordered according an hierarchical clustering method (Joe H. Ward, 1963). In the order, the average score of (A). NMI, (B). AMI, (C). RI, (D). ARI.

Again, by observing the dendrograms in Figure 6.17, one can see that all of the 4 metrics classify methods into two principle groups in a similar way that the co-performance matrices exposed in the previous section. The group of methods *CNM, Conclude, Oslom, Walktrap, LPA, SLPA, Infomap* mentioned in the last section also show very strong similarities in this experiment. Especially, *LPA* and *SLPA* being

[11] Where the corresponding methods are able to finish using a reasonable amount of time and memory as mentioned in the previous experiments.

based on label propagation mechanism show nearly identical results in many cases. Besides, one could also discern another group including *RB, CNM, GN* and *Louvain* (modularity based), which show a high consistency in general. Additionally, even with weaker scores, *SBM* and *DCSBM* are often found in the same group as well as *RCCLP-3* and *RCCLP-4*. In a global view, it seems that methods with a close theoretical approach tend to provide more similar results, which is also noticed in the previous sections.

Another information that could be extracted from this experiment is that RI should not be used as validation metrics for evaluating detection performance. Since its average values vary generally in a small range (0.9 to 1.0), it is more difficult to see the different between partitions. On the other hand, NMI and AMI shows very close results in our experiment, which are between 0.5 and 1.0 meaning that structural communities detected by different methods are quite comparable as concluded in the previous section. Finally, ARI seems to magnify the differences between methods, however there is no major difference in the similarity evaluation in comparison with the other metrics.

6.5 Related work

Orman *et al.* publish a comparative evaluation of eight community detection algorithms which most of them are also studied in this chapter (Orman, Labatut, and Cherifi, 2012). Different validation metrics are also used to compare detection performance and they also find that these metrics (RI, ARI, NMI) *"agree with each other with small differences when considering the way they rank algorithms"*, as also illustrated in Section 6.4. Beside, the authors also focus on analyzing many topological aspects of community structure including also community size, transitivity, density, etc. These topological qualities are then used to inspect community structures detected by different algorithms. The analyses allow the authors to conclude that these two approaches (topological metrics and validation metrics) to evaluate community structures are *"complementary and needed to perform a relevant and complete analysis of community detection results"*. They also propose that the *"traditional approach (RI, ARI, NMI) is much faster and easier to apply"*, and hence is proposed to be used first. However, in practice, reference community structures (ground-truths) are not usually available[12]. Therefore, from these above notices, our analyses in this chapter could be an important support dispensing additional information about the closeness between methods both in terms of topological aspect and partition-based aspect.

Agreste *et al.* evaluate different community detection algorithms in a empirical and comparative approach, especially for the context of web data analytic (Agreste et al., 2017). The authors find that *"time complexity is a crucial factor in the selection of a community detection algorithm"* and recommend that the label propagation method (LPA) *"has outstanding performance in scalability on both artificial and real graphs"*, which is also in a global agreement with our analysis in Section 6.1.1 providing predictions about required time of each method in function of network size. They also conclude that *"Infomap algorithm showcased the best trade-off between accuracy and computational performance"* based on NMI score. The conclusion could be valid in some specific

[12] In the context where a new algorithm is invented, one normally uses networks whose community structures are well known in order to validate the proposed method. In reality, since community detection is often employed to discover structures of new networks, hence it is not likely that reference community structures always exist.

cases when the reference community structure is well understood. Otherwise, as demonstrated in Section 4.2 of Chapter 4, some additional analyses should be done to obtain other structural aspects. Our conclusion is not as precise as the above ones, but are expected to give a more specific and quantitative information.

Ghasemian *et al.* present in a recent publication that an evaluation of overfitting and underfitting of several community detection models (Ghasemian, Hosseinmardi, and Clauset, 2018). The authors study the number of communities detected in practice by many methods and the maximum number of detectable clusters according to a theoretical model. Some conclusions are drawn about fitting qualities of methods in comparison to theoretical estimates. This study provides evidences that help to choose an appropriate method in function of fitting quality. Community detection methods are also grouped in distinct families based on their outputs on many real-world networks (similarity to our analysis in Section 6.4.1) using AMI metric. The authors also find that *"what an algorithm finds in a network depends strongly on the assumptions it makes about what to look for"*, which is aligned with our results through several analyses.

6.6 Conclusion

Finally, it is quite challenging to say which method is better in which scenario. It is at least as much demanding as defining all possible scenarios in the reality that could happen. Our experiments in this chapter provide several experiments demonstrating different aspects of community structure quality, which can be combined together in a flexible way to assist network analysts to find appropriate methods according to their context. Some questions could be sequentially asked during decision making processes:

1. What is the size of the network of interest and what is the acceptable computation time?
2. What are the expectancy about the number of communities as well as the community size distribution?
3. Is there any fitness function that should be optimized?
4. In case where the targeted method can not be deployed, is there any alternative solution?

The experiments and results in this chapter could help to identify quickly suitable method(s) if one is able to response the previous questions. Even still very far from being an exclusive analysis, our experiments cover a wide range of popular aspects of community structure that are studied in the state-of-the-art. Some primary conclusions that could be extracted from the analyses in this chapter can be cited:

- A consideration of computation time is very crucial in the process of choosing a community detection method for a problem at hand. As such, a well performed method in the literature can help one to reduce approximately 10^4 times of required computation time, which is significantly important in discovering large graph. Theoretical estimate of time complexity is important and reveals the scalability of a community detection method. On the other hand, practical computation time is worth being studied in practice for specific applications. Our estimates shown in Figure 6.6 provide detailed information

of practical time required by many popular community detection methods in function of network size. It could help network analysts to determine a suitable method(s) according to time availability.

- The expected number of communities to be obtained is another important criterion in choosing a community detection methods. According to the context, one would prefer different granularity levels to discover a network. Our study show that there are globally three main strategies that community detection methods decompose a network. Specifically, some identify communities whose size vary regularly in a wide range of values from very small to very large communities, some others divide networks into a huge number of very small communities and very few large communities, the last ones distribute nodes into similar-size communities. Therefore, knowing how a network should be broken down is very useful in order to end up with an appropriate community discovery method.

- In cases where one can determine a targeted objective function, designing new algorithms (or employing existing algorithms) that optimize the function would be the most evident. Since improving an objective function usually means expending more computation time, a compromise between getting higher fitness score and using less time needs to be considered. However, finding a good method to optimize an objective function satisfying a time constraint condition in the problem of community detection is not straightforward and needs many investigations. Our approach presented in the co-performance analysis provides network practitioners a quick glance about how different methods perform in improving some widely-used quality functions. This predictive information about the effect of using alternative methods in achieving good fitness scores would suggest network analysts multiple solutions for a certain objective function. This scenario is specifically useful when the desired method is too expensive in terms of computation time. Therefore, a combination of analyses presented in Section 6.1.1 and Section 6.3 is expected to eligible community detection method(s) for specific cases.

- Finally, we find that using some validation metrics to estimate the similarity between community detection methods would also provide interesting information that could help the decision process of network analysts. However, the distinction between the performance of different methods is less significant than that of the previous analyses. In case when one know exactly what should be found (ground-truth information), an analysis in this direction is important as it provides useful information about how a method is able to reach the desired result. However, this scenario is normally not popular in practice since community detection is often used to discover the structures of networks of which we do not a priori information. In these cases, validation metrics should be used as complementary evaluation to verify the appropriateness of using a method. For example, methods such as *SBM* or *RCCLP-4* detect partitions which are very discernible from that of the others. This means there could be significance differences in the way that nodes are distributed into communities in these two methods in comparison to those of the others. Hence, the use of these methods need to be examined in specific cases.

Chapter 7

Conclusion and perspectives

In this thesis, we study many aspects of the problem of community detection, which is an important exploratory task to discover large scale structure of networks without knowing any additional information, such as number of communities or size distribution. Thanks to a promising perspective that community detection could help to understand the mesoscopic structure of complex networks in many domains, a plethora of methods has have been invented in the last decades. However, community detection is an ill-defined problem, i.e. there is no consensus on what should be considered as a good community structure. Therefore, although some protocolaire procedures exist in the evaluation of detection accuracy, the disagreement on community structure goodness still provokes many discussions in the literature. There is a need to revisit essential notions of the community detection problem in order to better understand community structure in real world networks as well as community detection methods. For that reasons, we conduct the following analyses:

Firstly, we provided a novel analysis process to characterize community structure of networks into many topological groups which show different node organization patterns. Each representative group is then associated to a corresponding graph generative model that produces a high similarity in connection patterns including star-like, tree-like, grid-like, string-like. Our empirical study uncovers that networks across different categories including communication, technological, information, biological and social networks might have different community structures and can be described by distinguishable characterized topologies. The difference of modular topology between networks in various categories could help to construct network profiles or network signatures by domain of study, and hence open a possibility for creating adapted network generative models, network class prediction algorithms, etc. Specifically, since networks in each domain reveal some particular modular structures, the mechanisms which are responsible for their creation, evolution, degradation are also discernible. Hence, different simulation or analysis strategies will generate different impacts on the networks in a predictable way if their structures are well understood. In other words, the network structure profiling assists to achieve suitable network analysis processes and to interpret obtained results without requiring expensive brute force analysis.

On the other hand, we study many state-of-the-art community detection methods. However, the evaluation of these methods still presents many challenges. Specifically, a conventional way to evaluate the accuracy of an algorithm is to design an objective function reflecting how a solution is *"close"* to the expected one. If the expected community structure can be defined clearly, evaluating the performance of a method would be straightforward and universal via a unique function. On the contrary, it is also possible that in each domain or each specific application, one will expect to discover modular structures of one kind or another due to some reasons. In

this case, modular structure is context dependent and corresponding objective functions must be designed accordingly. These two approaches are principally different and the way that we define what community detection is will crucially impact the way algorithms are designed and evaluated.

- Global approach: If a universal notion of community structure could be defined by an objective function, then evaluating the quality of a method will become simply inspecting its corresponding scores measured by the function. The problem of community detection becomes similar to the traveling salesman problem, where the notion of *optimal result* is clearly defined. In this scenario, other algorithms whose objective are different from the standard one should not be called community detection since they are looking for something else. Consequently, evaluating the performance of a community detection algorithm is simply analyzing the expected fitness scores of its outcomes.
- Contextual approach: Community detection is a problem-dependent task, algorithms to detect community structures should be attached with specific contexts. In other words, there will be an objective function for each context (based on network model for example) and the term *community detection* should not be considered as a specific task, but a class of methodologies. In this scenario, the evaluation of any community detection algorithm should take into consideration corresponding hypotheses which were *embedded* into the algorithm. For example, a method which is designed to discover community structure on bipartite network should not be validated on a *traditional* ground-truth partition. In the same way that a method designed to discover core-periphery community structure should not be verified on other kinds of modular structures.

There is no consensus in the literature on how the problem of community detection should be considered. Reaching a global agreement for the task of community detection is unrealistic as it is somehow irrational to impose a unique objective function on this exploratory task. Recent approaches in finding community structure try to search for solutions that optimize likelihood functions that are regulated by different underlying network models based on stochastic block models. Here, a context can be considered as a network model that controls the nascency of a network (configuration) by some certain probability functions. In this scenario, defining contextual network models that explain well network structures in specific problems are vital. That is the reason why many studies need to be conducted to understand the underlying mechanisms that are responsible for the interactions of nodes in different kind of networks.

Which method then?

Since there is no agreement on the expected community structure that should be found on networks, proposing a comprehensive recommendation of community detection methods is not straightforward and is dependent on specific context. Nevertheless, the experiments and analyses presented in this thesis provoke some guidelines that could support network analysts to determine eligible detection methods as follows:

- Time complexity is an important factor to be examined when one needs to consider different solutions. For example, a fast method can be 10^4 times

faster than a slow method, even on very small networks of less than a hundred nodes. A detailed time computation analysis could be found in Section 6.1.1

- In the scenarios where no information about the expected structure or partition is available, *Louvain* method and *LPA* are very good points to start. These methods are among the highest scalable methods and can be accomplished in a few seconds for networks with some million nodes and edges. The modularity of their clustering results are relatively high. In general, *Louvain* provides larger communities and higher modularity with respect to *LPA*. However, one could choose lower-levels of hierarchical clusters to acquire small communities in *Louvain* or aggregate several clusters to acquire large communities in *LPA* if another resolution needs to be examined.

- In cases where one needs to investigate networks at a certain resolution, further inspections on community size distribution as introduced in Section 6.1.2 and/or size distribution similarity as shown in Section 6.2 could be beneficial. They provide informative indications on the expected granularity that could be produced by different community detection methods.

- When one has a specific objective function, community detection is reduced to an optimization problem. In this case, it is better focus on methods whose objective functions correspond to the desired function. However, when the optimization process of an objective function is too expensive, alternative solutions could be envisaged based on a co-performance analysis as presented in Section 6.3.

- Validation metrics (such as AMI, ARI, etc.) should be used to verify the outcomes of detection methods when several solutions need to be compared as mentioned in Section 6.4. This kind of analysis is especially valuable when objective function (or reference modular structure) is not available. Specifically, it could help to identify groups of methods whose outcomes are relatively comparable. Hence, one can reduce significantly the number of necessary calculations by focusing only on some representative methods.

- Finally, if quality metrics could not help to get insight into the differences of community structures detected by different methods, it is worth investigating into details information about topological structure. Specifically, it is advantageous to examine the distribution of different interaction patterns of nodes in communities detected by each method as described in Chapter 5 and Chapter 6. Indeed, choosing a community detection method is also about how to explain and interpret its results which is also very challenging. The introduced characterization contributes a practice that helps to explore community structure in an intuitive and informative way.

Limitations and perspectives

Although we try to cover as much as possible several aspects of community structure, there are still many important tasks to be done which were not addressed in our study:

- The quality of network data needs to be examined in more details. In a conventional manner, we include many well-studied networks in the literature in

our experiments. However, there must be considerable impact from the characteristic of the involved dataset on the final analytical outcomes. Analyzing the detection performance of community detection methods with respect to the variation of input data could provide insightful information and is worth studying. There is a need of developing a framework to deal with the problem of data reliability (Guimerà and Sales-Pardo, 2009).

- In Chapter 5, we characterize many topological patterns showing how nodes interact with each other in networks of different categories. The difference in topology implies that algorithms could be highly specialized to get better performance on some specific cases. Therefore, there should be specialized algorithms and associating evaluation benchmarks in order to validate detection performance.

- We concentrated on the most traditional notions of community detection, i.e. networks considered are unweighted, undirected and communities are disjointed. Another breath analysis direction into directed and/or weighted networks as well as overlapping communities could be addressed in order to cover other scenarios of community detection in practice.

Finally, we conclude this dissertation by some important notices that would be beneficial in community detection analysis:

- Determining what is called *community structure* is very indispensable in constructing appropriate analytical process. Since the term *community* may relate to many things in practice (ground-truth, semantic, structural, communities with attributes, etc.) and many models in network science, understanding what we are looking for helps to identify appropriate analysis tools.

- Understanding various characteristics of the network under consideration provides valuable information for the analysis process. For example, one should investigate how large and how dense is the network; what is the degree distribution; what could be the underlying mechanisms that are responsible for the connection of nodes; how many communities could be appropriate and how large are they, etc.

- There are many reasons why one would like to discover community structure on a network in, practice. Clarifying the main purpose that motivates us to analyze community structure could lead to suitable objective functions, quality metrics and eventually detection methods.

Appendix A

Supporting Information

A.1 Modularity

Given a graph $\mathcal{G}$ of m edges, the expected probability of having an edge between a vertex i of degree k_i and vertex j of degree k_j can be calculated by rewiring disconnected edges (stubs) while conserving the number of edges leaving each vertex. The probability p_i of choosing a stub incident with i is $p_i = \frac{k_i}{2m}$ as there are k_i stubs connecting i to other parts of the graph out of $2m$ stubs (total degree). Hence, the expected number of edges between i and j is $p_{ij} = 2mp_ip_j = \frac{k_ik_j}{2m}$ as there are $2m$ times of choosing a stub randomly. The modularity can be calculated as:

$$Q = \frac{1}{2m} \sum_{i,j \in \mathcal{V}} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j), \tag{A.1}$$

where c_i is the community of vertex i, $\delta(c_i, c_j)$ is the Kronecker delta function which is equal to 1 when $c_i = c_j$ and to 0 otherwise. This quality measure can be literally translated as:

$$Q = \text{(number of edges inside communities)} - \text{(expected number of such edges)}. \tag{A.2}$$

Because only intra-community edges contribute to the final score of modularity, the equation A.1 can be rewritten by grouping the amount of contributions of edges in the same communities in the same factions as:

$$Q = \sum_{c=1}^{n_c} \left[\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right], \tag{A.3}$$

where n_c is the number of communities, l_c is the total number of edges connecting vertices of community c and d_c is the total degree of vertices in community c. The modularity function varies between $-1 < Q < 1$ where negative (positive) values indicate a partition in which there are more inter-community (intra-community) edges than the expectancy. Values approaching 1 represent very strong community structure and accordingly -1 for heterophily structure while $Q \approx 0$ implies random regrouping. In practice, modularity scores often fall in the range between 0.3 and 0.7.

Methods that produce multiple partitions such as hierarchical clustering often use the modularity quality function to rank partitions' quality. The partition corresponding to the highest community score is often taken as the final outcome. Besides the utilization as a quality metric, the modularity function is also used as an objective

function for many clustering optimization problems, such as approaches presented in Section 3.2.3.

A.2 Edge betweenness centrality

Going into the calculation of edge betweenness centrality, calculating the shortest path between a pair of vertices in a graph using breadth first search takes $\mathcal{O}(m)$ time and there are $\mathcal{O}(n^2)$ pairs of vertices. In total, it would take $\mathcal{O}(mn^2)$ time to calculate all edge betweenness scores for a graph, which is really impractical. The authors proposed to use a faster algorithm, invented independently by (Brandes, 2001) and (Newman, 2001a) that performs the calculation in $\mathcal{O}(mn)$ time for each removal. The algorithm uses a first-in first out queue Q similarity to the classical breath first search algorithm in order to discover the other vertices in the graph from a source vertex s. Additionally, a last-in first out stack S is also in use to stock vertices in order of non-increasing distance from the source vertex s. These ordered vertices are served to accumulate betweenness centrality scores from the most distant vertices (leaves) towards the source vertex s. We maintain for each vertex v a list of direct *predecessors*[1] $v.P$ that is either null or a list of other vertex/vertices. In addition, each vertex v contains its geodesic distance $v.d$ to source s, the number of shortest paths $v.\sigma$ from the source s to v, the vertex betweenness centrality $v.\delta$ representing the sum of the centrality contributions of its direct successor edge(s) for its direct predecessor edge(s). Assuming that graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is represented by an adjacency list $\mathcal{G}.Adj[v], v \in \mathcal{V}$, the processes to calculate edge betweenness centrality are presented in Algorithm 3.

[1] The shortest distance path(s) going from the source vertex s to vertex v must go through the predecessor(s) of v before reaching v.

Algorithm 3: Edge betweenness centrality

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Output: $C_B[u, v]$; $e = (u, v) \in \mathcal{E}$; $u, v \in \mathcal{V}$

$C_B[u, v] = 0; (u, v) \in \mathcal{E}$
for $s \in \mathcal{V}$ **do**
  $S \leftarrow$ empty stack
  $w.P \leftarrow$ empty list, $w \in \mathcal{V}$
  $t.\sigma \leftarrow 0, t \in \mathcal{V}, s.\sigma \leftarrow 1$
  $t.d \leftarrow -1, t \in \mathcal{V}, s.d \leftarrow 0$
  $Q \leftarrow$ empty queue
  $ENQUEUE(Q, s)$
  while $Q \neq \emptyset$ **do**
    $v \leftarrow DEQUEUE(Q)$
    $PUSH(S, v)$
    for *each* $w \in \mathcal{G}.Adj[v]$ **do**
      // If w is found for the first time
      if $w.d < 0$ **then**
        $ENQUEUE(Q, w)$
        $w.d \leftarrow v.d + 1$
      end
      // If v is a direct predecessor of w
      if $w.d = v.d + 1$ **then**
        $w.\sigma \leftarrow w.\sigma + v.\sigma$
        $APPEND(w.P, v)$
      end
    end
  end
  $v.\delta \leftarrow 0, v \in \mathcal{V}$
  while $S \neq \emptyset$ **do**
    $w \leftarrow POP(S)$
    if $w \neq s$ **then**
      for *each* $v \in w.P$ **do**
        $C_B[v, w] \leftarrow C_B[v, w] + \frac{v.\sigma}{w.\sigma}(1 + w.\delta)$
        $v.\delta \leftarrow v.\delta + \frac{v.\sigma}{w.\sigma}(1 + w.\delta)$
      end
    end
  end
end
return $C_B[u, v]$

A.3 Graph spectral partitioning

Given a graph $\mathcal{G}$ with an associated adjacency matrix A. It is possible to represent the number of inter-cluster edges called cut size of a graph bisection into two clusters as:

$$R = \frac{1}{2}\sum_{ij} A_{ij}(1 - \delta_{c_i c_j}), \tag{A.4}$$

where c_i and c_j represents the cluster of vertex i and vertex j receptively; $\delta_{c_i c_j} = 1$ if $c_i = c_j$ and 0 elsewhere. Every partition of graph $\mathcal{G}$ of n vertices into two clusters can be represented using an *index vector* $\mathbf{s} = [s_1, s_2, ..., s_n]^T$, whose components is $+1$ if the associated vertex belongs to cluster 1 and -1 if the associated vertex belongs to cluster 2:

$$s_i = \begin{cases} +1 \text{ if vertex } i \text{ belongs to group 1,} \\ -1 \text{ if vertex } i \text{ belongs to group 2.} \end{cases} \tag{A.5}$$

Then we have:

$$\frac{1}{2}(1 - s_i s_j) = (1 - \delta_{c_i c_j}) = \begin{cases} 1 \text{ if } c_i \neq c_j, \\ 0 \text{ if } c_i = c_j. \end{cases} \tag{A.6}$$

The partition cut size presented in Equation (A.4) can be rewritten as:

$$R = \frac{1}{4}\sum_{ij} A_{ij}(1 - s_i s_j) \tag{A.7}$$

The first term in Equation (A.7) can be represented as:

$$\sum_{ij} A_{ij} = \sum_i \sum_j A_{ij} = \sum_i k_i = \sum_i s_i^2 k_i = \sum_{ij} s_i s_j k_i \delta_{ij}, \tag{A.8}$$

Equation (A.7) is then:

$$R = \frac{1}{4}\sum_{ij} s_i s_j (k_i \delta_{ij} - A_{ij}), \tag{A.9}$$

which can be represented in matrix form as:

$$R = \frac{1}{4}\mathbf{s}^T(\mathbf{D} - \mathbf{A})\mathbf{s} = \frac{1}{4}\mathbf{s}^T\mathbf{L}\mathbf{s}, \tag{A.10}$$

where $\mathbf{D} = diag(k_1, k_2, ..., k_n)$ is the diagonal *degree matrix* of graph $\mathcal{G}$ and $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is called *Laplacian matrix*. The minimization of cut size quality is equivalent to choosing a partition, hence the vector $\mathbf{s}$, in such a way that R in minimized. The index vector $\mathbf{s}$ can be written as a linear combination of the normalized eigenvectors $\mathbf{v}_i = [v_1^{(i)}, v_2^{(i)}, ..., v_n^{(i)}]^T$ of the Laplacian matrix:

$$\mathbf{s} = \sum_{i=1}^{n} a_i \mathbf{v}_i, \tag{A.11}$$

where $a_i = \mathbf{v}_i^T \mathbf{s}$. Since $\mathbf{s}^T \mathbf{s} = n$, the linear combination coefficients are constrained by:

$$\sum_{i=1}^{n} a_i^2 = n. \tag{A.12}$$

The cut size hence follows:

$$R = \sum_i a_i \mathbf{v}_i^T L \sum_j a_j \mathbf{v}_j = \sum_{ij} a_i a_j \lambda_j \delta_{ij} = \sum_i a_i^2 \lambda_i, \tag{A.13}$$

where λ_i is the eigenvalue corresponding to eigenvector $\mathbf{v}_i$[2] of matrix $\mathbf{L}$.

Equation (A.13) shows that the cut size of a graph partition can be represented as a linear combination of all eigenvalues of the corresponding Laplacian matrix. Thus, the task of minimizing cut size is equivalent to the task of choosing the quantities a_i^2, hence $\mathbf{s} = \sum_i a_i \mathbf{v}_i$, so as to place as much as possible of the weight a_i^2 to the lowest eigenvalues λ_i provided that Equation (A.12) requires a normalization constraint $\sum_{i=1}^{n} a_i^2 = n$. Without loss of generality, these eigenvalues can be labeled in an increasing order as: $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. The task is minimizing a_i^2 corresponding to small values of i.

The Laplacian matrix is symmetric and the sum of every row (and column) is zero:

$$\sum_j L_{ij} = \sum_j (k_i \delta_{ij} - A_{ij}) = k_i - k_i = 0, \tag{A.14}$$

Thus the vector $[1, 1, 1, ...]^T$ is always an eigenvector corresponding to eigenvalue $\lambda_1 = 0$ of the Laplacian matrix (all eigenvalues are nonnegative, which makes 0 always the smallest eigenvalue). Following the normalization condition of the eigenvectors, one has $\mathbf{v}_1 = [1, 1, 1, ...]^T / \sqrt{n}$. Hence, it is straightforward to see that the cut size R can be minimized by choosing $\mathbf{s} = [1, 1, 1, ...]$ paralleling to $\mathbf{v}_1$, then all the weights of a_i^2 in Equation (A.13) are concentrated in $a_1^2 = n$ corresponding to eigenvalue $\lambda_1 = 0$ and all other terms are zero. This choice leads us to the result $R = 0$ which is the smallest possible value of cut size. The partition of this solution, in fact a division of the graph into one big cluster containing all vertices and the other having no vertex, is trivial and not helpful for the graph partitioning task. Several approaches are proposed to eliminate this trivial solution, but the most common is to define the sizes of two clusters being n_1 and n_2, where $n_1 + n_2 = n$. Since the value of $\mathbf{v}_1$ is constant, this imposition fixes the coefficient corresponding to λ_1:

$$a_1^2 = (\mathbf{v}_1^T \mathbf{s})^2 = \frac{(n_1 - n_2)^2}{n} \tag{A.15}$$

The coefficient a_1 gets the lowest value when $n_1 = n_2 = \frac{n}{2}$. The minimization of R leads us to distributing other coefficients a_i^2, for $i > 1$ of Equation (A.13) to small eigenvalues. A straightforward solution would be choosing the index vector s proportional to the second eigenvector $\mathbf{v}_2$ called *Fiedler vector* (Fiedler, 1973) corresponding to the second smallest eigenvalue known as *algebraic connectivity*. This choice implies a value of zero to other coefficients a_i for $i > 2$, since eigenvectors are orthogonal $\mathbf{v}_i^T \mathbf{s} = 0, \forall i > 2$. However, since index vector elements s_i can not receive every value but only $+1$ or -1, $\mathbf{s}$ can not be choosen to be parallel with $\mathbf{v}_2$. Hence, one try to choose $\mathbf{s}$ to be as close to parallel with $\mathbf{v}_2$ as possible. Since

[2]Eigenvectors of matrix L are orthogonal, hence for all i and $j < n$, we can write $\mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}$

$$|a_2| = |\mathbf{v}_2^T \mathbf{s}| = |\sum_{i=1}^{n} v_i^{(2)} s_i| \leq \sum_{i=1}^{n} |v_i^{(2)}|, \tag{A.16}$$

the indexes s_i can be chosen so as to make the terms $v_i^{(2)} s_i$ all positive or all negative:

$$s_i = \begin{cases} \pm 1 \text{ if } v_i^{(2)} \geq 0, \\ \mp 1 \text{ if } v_i^{(2)} < 0. \end{cases} \tag{A.17}$$

Taking into account the constraint about the sizes of clusters stated previously, it is not possible to attain the equality sign in Equation (A.16) unless the Fiedler vector has the proportion of nonnegative values being n_1/n or n_2/n. In the most optimist scenario, one can rarely obtain this optimal solution. The best solution is achieved by distributing vertices into two clusters in order of the values of the Fiedler vector. In other words, vertices corresponding to n_1 largest (or smallest) values of $\{v_i^{(2)}\}$ are assigned to cluster 1 and the rest are assigned to cluster 2. The solution that produce the smaller cut size R among the two solutions will be preferable. The time complexity of the algorithm depends principally on the step of calculation of eigenvectors. Since the Laplacian matrix is often sparse, it requires $\mathcal{O}(n+m)$ time for each of $\mathcal{O}(n)$ operations using the iterative power method or the Lanczos method (Lanczos, 1950). In total, the time complexity is $\mathcal{O}(n(n+m))$.

Bibliography

Agreste, Santa, Pasquale De Meo, Giacomo Fiumara, Giuseppe Piccione, Sebastiano Piccolo, Domenico Rosaci, Giuseppe M. L. Sarne, and Athanasios V. Vasilakos (2017). "An Empirical Comparison of Algorithms to Find Communities in Directed Graphs and Their Application in Web Data Analytics". In: *IEEE Transactions on Big Data* 3.3, pp. 289–306. DOI: 10.1109/tbdata.2016.2631512. URL: https://doi.org/10.1109/tbdata.2016.2631512.

Albert, R., H. Jeong, and A.-L. Barabási (Sept. 1999). "Internet: Diameter of the World-Wide Web". In: *Nature* 401, pp. 130–131. DOI: 10.1038/43601.

Aldecoa, Rodrigo and Ignacio Marín (Sept. 2011). "Deciphering Network Community Structure by Surprise". In: *PLoS ONE* 6.9. Ed. by Eshel Ben-Jacob, e24195. DOI: 10.1371/journal.pone.0024195. URL: https://doi.org/10.1371/journal.pone.0024195.

Amaral, L.A.N., A. Scala, M. Barthélémy, and H.E. Stanley (2000). "Classes of small-world networks". In: *Proceedings of the National Academy of Sciences* 97.21, pp. 11149–11152. ISSN: 0027-8424. DOI: 10.1073/pnas.200327197. eprint: http://www.pnas.org/content/97/21/11149.full.pdf. URL: http://www.pnas.org/content/97/21/11149.

Ames, Brendan P. W. (2013). "Guaranteed clustering and biclustering via semidefinite programming". In: *Mathematical Programming* 147.1-2, pp. 429–465. DOI: 10.1007/s10107-013-0729-x. URL: https://doi.org/10.1007/s10107-013-0729-x.

Ana, L.N.F. and A.K. Jain (2003). "Robust data clustering". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Proceedings.* DOI: 10.1109/cvpr.2003.1211462. URL: https://doi.org/10.1109/cvpr.2003.1211462.

Andrei, Broder, Kumar Ravi, Maghoul Farzin, Raghavan Prabhakar, Rajagopalan Sridhar, Stata Raymie, Tomkins Andrew, and Wiener Janet (2000). "Graph structure in the Web". In: *Computer Networks* 33.1, pp. 309 –320. ISSN: 1389-1286. DOI: https://doi.org/10.1016/S1389-1286(00)00083-9. URL: http://www.sciencedirect.com/science/article/pii/S1389128600000839.

Arifin, S, Zulkardi, R I I Putri, Y Hartono, and E Susanti (2017). "Developing Ill-defined problem-solving for the context of South Sumatera". In: *Journal of Physics: Conference Series* 943, p. 012038. DOI: 10.1088/1742-6596/943/1/012038. URL: https://doi.org/10.1088/1742-6596/943/1/012038.

Barabási, Albert-László and Réka Albert (1999). "Emergence of Scaling in Random Networks". In: *Science* 286.5439, pp. 509–512. ISSN: 0036-8075. DOI: 10.1126/science.286.5439.509. URL: http://science.sciencemag.org/content/286/5439/509.

Barrat, A. and M. Weigt (Jan. 2000). "On the properties of small-world network models". In: *European Physical Journal B* 13, pp. 547–560. eprint: cond-mat/9903411.

Barrat, A., M. Barthélemy, R. Pastor-Satorras, and A. Vespignani (2004). "The architecture of complex weighted networks". In: *Proceedings of the National Academy of Sciences* 101.11, pp. 3747–3752. ISSN: 0027-8424. DOI: 10.1073/pnas.0400087101.

eprint: http://www.pnas.org/content/101/11/3747.full.pdf. URL: http://www.pnas.org/content/101/11/3747.

Blondel, Vincent D, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre (Oct. 2008). "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10, P10008. DOI: 10.1088/1742-5468/2008/10/p10008. URL: https://doi.org/10.1088/1742-5468/2008/10/p10008.

Bohlin, Ludvig, Daniel Edler, Andrea Lancichinetti, and Martin Rosvall (2014). "Community Detection and Visualization of Networks with the Map Equation Framework". In: *Measuring Scholarly Impact*. Springer International Publishing, pp. 3–34. DOI: 10.1007/978-3-319-10377-8_1. URL: https://doi.org/10.1007/978-3-319-10377-8_1.

Bosch, Antal van den, Toine Bogers, and Maurice de Kunder (May 2016). "Estimating search engine index size variability: a 9-year longitudinal study". In: *Scientometrics* 107.2. ISSN: 1588-2861. DOI: 10.1007/s11192-016-1863-z. URL: https://doi.org/10.1007/s11192-016-1863-z.

Bothorel, Cécile, Juan David Cruz, Matteo Magnani, and Barbora Micheková (2015). "Clustering attributed graphs: Models, measures and methods". In: *Network Science* 3.03, pp. 408–444. DOI: 10.1017/nws.2015.9. URL: https://doi.org/10.1017/nws.2015.9.

Brandes, U., D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner (2008). "On Modularity Clustering". In: *IEEE Transactions on Knowledge and Data Engineering* 20.2, pp. 172–188. DOI: 10.1109/tkde.2007.190689. URL: https://doi.org/10.1109/tkde.2007.190689.

Brandes, Ulrik (June 2001). "A faster algorithm for betweenness centrality". In: *The Journal of Mathematical Sociology* 25.2, pp. 163–177. DOI: 10.1080/0022250x.2001.9990249. URL: https://doi.org/10.1080/0022250x.2001.9990249.

Brun, Armelle and Anne Boyer (2012). "Détection de communautés d'intérêt et recommandation sociale par leaders". In: *Ingénierie des systèmes d'information* 17.6, pp. 91–113. DOI: 10.3166/isi.17.6.91-113. URL: https://doi.org/10.3166/isi.17.6.91-113.

Burt, Ronald (1992). *Structural holes : the social structure of competition*. Cambridge, Mass: Harvard University Press. ISBN: 9780674843714.

Cano, P., O. Celma, M. Koppenberger, and J. M. Buldú (Mar. 2006). "Topology of music recommendation networks". In: *Chaos* 16.1, p. 013107. DOI: 10.1063/1.2137622.

Chakraborty, Tanmoy, Ayushi Dalmia, Animesh Mukherjee, and Niloy Ganguly (Aug. 2017). "Metrics for Community Analysis: A Survey". In: *ACM Comput. Surv.* 50.4, pp. 1–37. ISSN: 0360-0300. DOI: 10.1145/3091106. URL: http://doi.acm.org/10.1145/3091106.

Chaoming, Song, Havlin Shlomo, and Makse Hernán A. (Jan. 2005). "Self-similarity of complex networks". In: *Nature* 433. DOI: doi:10.1038/nature03248.

Clauset, A., C. Rohilla Shalizi, and M. E. J. Newman (June 2009). "Power-law distributions in empirical data". In: *SIAM Review* 51. arXiv: 0706.1062 [physics.data-an].

Clauset, Aaron, M. E. J. Newman, and Cristopher Moore (Dec. 2004). "Finding community structure in very large networks". In: *Physical Review E* 70.6. DOI: 10.1103/physreve.70.066111. URL: https://doi.org/10.1103/physreve.70.066111.

Cleveland, William S. (1979). "Robust Locally Weighted Regression and Smoothing Scatterplots". In: *Journal of the American Statistical Association* 74.368, pp. 829–836.

DOI: 10.1080/01621459.1979.10481038. URL: https://doi.org/10.1080/01621459.1979.10481038.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Stein Clifford (2009). *Introduction to algorithms*. The MIT Press. ISBN: 978-0-262-03384-8.

Cortes, Corinna, Daryl Pregibon, and Chris Volinsky (2001). "Communities of Interest". In: *Advances in Intelligent Data Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 105–114. ISBN: 978-3-540-44816-7.

Coscia, Michele, Fosca Giannotti, and Dino Pedreschi (Sept. 2011). "A classification for community discovery methods in complex networks". In: *Statistical Analysis and Data Mining* 4.5, pp. 512–546. DOI: 10.1002/sam.10133. URL: https://doi.org/10.1002/sam.10133.

Creusefond, Jean, Thomas Largillier, and Sylvain Peyronnet (2015). "Finding compact communities in large graphs". In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM 15*. ACM Press. DOI: 10.1145/2808797.2808868.

Creusefond, Jean, Thomas Largillier, and Sylvain Peyronnet (2016). "On the Evaluation Potential of Quality Functions in Community Detection for Different Contexts". In: *Advances in Network Science*, pp. 111–125. DOI: 10.1007/978-3-319-28361-6_9. URL: https://doi.org/10.1007/978-3-319-28361-6_9.

Daly, Elizabeth M., Werner Geyer, and David R. Millen (2010). "The Network Effects of Recommending Social Connections". In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys '10, pp. 301–304. ISBN: 978-1-60558-906-0. DOI: 10.1145/1864708.1864772.

Danon, Leon, Albert Díaz-Guilera, Jordi Duch, and Alex Arenas (Sept. 2005). "Comparing community structure identification". In: *Journal of Statistical Mechanics: Theory and Experiment* 2005.09, P09008–P09008. DOI: 10.1088/1742-5468/2005/09/p09008. URL: https://doi.org/10.1088/1742-5468/2005/09/p09008.

Dao, Vinh-Loc, Cécile Bothorel, and Philippe Lenca (2017a). "Community detection methods can discover better structural clusters than ground-truth communities". In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017 - ASONAM 17*. ACM Press. DOI: 10.1145/3110025.3110053.

Dao, Vinh-Loc, Cécile Bothorel, and Philippe Lenca (2017b). "Community Structures Evaluation in Complex Networks: A Descriptive Approach". In: *3rd International Winter School and Conference on Network Science*, pp. 11–19. DOI: 10.1007/978-3-319-55471-6_2.

Dao, Vinh-Loc, Cécile Bothorel, and Philippe Lenca (2018a). "An empirical characterization of community structures in complex networks using a bivariate map of quality metrics". In: *ArXiv e-prints*. eprint: 1806.01386.

Dao, Vinh-Loc, Cécile Bothorel, and Philippe Lenca (2018b). "Community structure: A comparative evaluation of community detection methods". In: *arXiv e-prints*. eprint: 1812.06598 (cs.SI).

Dao, Vinh-Loc, Cécile Bothorel, and Philippe Lenca (2018c). "Estimating the similarity of community detection methods based on cluster size distribution". In: *The 7th International Conference on Complex Networks and Their Applications*. Vol. 812. Springer International Publishing, pp. 183–194. DOI: 10.1007/978-3-030-05411-3_15.

Davis, Allison, Burleigh B Gardner, and Mary R Gardner (1941). *Deep South, a social anthropological study of caste and class*. University of Chicago Press, IL, US.

Dijkstra, E.W. (Dec. 1959). "A Note on Two Problems in Connexion with Graphs". In: *Numer. Math.* 1.1, pp. 269–271. ISSN: 0029-599X. DOI: 10.1007/BF01386390. URL: http://dx.doi.org/10.1007/BF01386390.

Dongen, Stijn van (2000). *A Cluster Algorithm for Graphs*. Tech. rep.

Dorogovtsev, S. N. and J. F. F. Mendes (June 2002). "Evolution of networks". In: *Advances in Physics* 51, pp. 1079–1187. DOI: 10.1080/00018730110112519. eprint: cond-mat/0106144.

Duch, Jordi and Alex Arenas (2005). "Community detection in complex networks using extremal optimization". In: *Physical Review E* 72.2.

Eckmann, Jean-Pierre, Elisha Moses, and Danilo Sergi (2004). "Entropy of dialogues creates coherent structures in e-mail traffic". In: *Proceedings of the National Academy of Sciences* 101.40, pp. 14333–14337. ISSN: 0027-8424. DOI: 10.1073/pnas.0405728101. URL: http://www.pnas.org/content/101/40/14333.

Elias, P., A. Feinstein, and C. Shannon (Dec. 1956). "A note on the maximum flow through a network". In: *IEEE Transactions on Information Theory* 2.4, pp. 117–119. DOI: 10.1109/tit.1956.1056816. URL: https://doi.org/10.1109/tit.1956.1056816.

Erdős, P. and A. Rényi (1959). "On random graphs, I". In: *Publicationes Mathematicae (Debrecen)* 6, pp. 290–297. URL: http://www.renyi.hu/\~{}p_erdos/Erdos.html\#1959-11.

Erzsébet, Ravasz and Barabási Albert-László (Feb. 2003). "Hierarchical organization in complex networks". In: *Physical Review E* 67.2. DOI: 10.1103/physreve.67.026112. URL: https://doi.org/10.1103/physreve.67.026112.

Estrada, Ernesto (Dec. 2010). "Quantifying network heterogeneity". In: *Physical Review E* 82.6. DOI: 10.1103/physreve.82.066102. URL: https://doi.org/10.1103/physreve.82.066102.

Estrada, Ernesto, Desmond J. Higham, and Naomichi Hatano (Mar. 2009). "Communicability betweenness in complex networks". In: *Physica A: Statistical Mechanics and its Applications* 388.5, pp. 764–774. DOI: 10.1016/j.physa.2008.11.011. URL: https://doi.org/10.1016/j.physa.2008.11.011.

Estrada, Ernesto (2011). *The structure of complex networks: Theory and applications*. Oxford University Press. ISBN: 978-0-19-959175-6. DOI: 10.1007/978-0-387-84858-7.

Faloutsos, Michalis, Petros Faloutsos, and Christos Faloutsos (Aug. 1999). "On Power-law Relationships of the Internet Topology". In: *SIGCOMM Comput. Commun. Rev.* 29.4, pp. 251–262. ISSN: 0146-4833. DOI: 10.1145/316194.316229. URL: http://doi.acm.org/10.1145/316194.316229.

Fiedler, Miroslav (1973). "Algebraic connectivity of graphs". In: *Czechoslovak Mathematical Journal* 3.

Flaviano, Morone, Min Byungjoon, Bo Lin, Mari Romain, and A. Makse Hernán (July 2016). "Collective Influence Algorithm to find influencers via optimal percolation in massively large social media". In: *Scientific Reports* 6.1. DOI: 10.1038/srep30062. URL: https://doi.org/10.1038/srep30062.

Fortunato, S. and M. Barthelemy (Dec. 2006). "Resolution limit in community detection". In: *Proceedings of the National Academy of Sciences* 104.1, pp. 36–41. DOI: 10.1073/pnas.0605965104. URL: https://doi.org/10.1073/pnas.0605965104.

Fortunato, Santo (Feb. 2010). "Community detection in graphs". In: *Physics Reports* 486.3-5, pp. 75–174. DOI: 10.1016/j.physrep.2009.11.002. URL: https://doi.org/10.1016/j.physrep.2009.11.002.

Fortunato, Santo and Darko Hric (Nov. 2016). "Community detection in networks: A user guide". In: *Physics Reports* 659, pp. 1–44. DOI: 10.1016/j.physrep.2016.09.002. URL: https://doi.org/10.1016/j.physrep.2016.09.002.

Fouss, Francois, Alain Pirotte, Jean michel Renders, and Marco Saerens (Mar. 2007). "Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation". In: *IEEE Transactions on Knowledge and Data Engineering* 19.3, pp. 355–369. DOI: 10.1109/tkde.2007.46. URL: https://doi.org/10.1109/tkde.2007.46.

Francois, Fouss, Pirotte Alain, Renders Jean-Michel, and Saerens Marco (2004). "A Novel Way of Computing Dissimilarities between Nodes of a Graph, with Application to Collaborative Filtering". In: *Workshop on Statistical Approaches for Web Mining*.

Freeman, Linton C. (Mar. 1977). "A Set of Measures of Centrality Based on Betweenness". In: *Sociometry* 40.1, p. 35. DOI: 10.2307/3033543. URL: https://doi.org/10.2307/3033543.

Fronczak, Agata, Piotr Fronczak, and Janusz A. Hołyst (Oct. 2003). "Mean-field theory for clustering coefficients in Barabási-Albert networks". In: *Physical Review E* 68.4. DOI: 10.1103/physreve.68.046126. URL: https://doi.org/10.1103/physreve.68.046126.

Fruchterman, Thomas M. J. and Edward M. Reingold (Nov. 1991). "Graph Drawing by Force-directed Placement". In: *Softw. Pract. Exper.* 21.11, pp. 1129–1164. ISSN: 0038-0644. DOI: 10.1002/spe.4380211102. URL: http://dx.doi.org/10.1002/spe.4380211102.

Ghasemian, A., H. Hosseinmardi, and A. Clauset (Feb. 2018). "Evaluating Overfit and Underfit in Models of Network Community Structure". In: *ArXiv e-prints*. arXiv: 1802.10582 [stat.ML].

Gilbert, E. N. (Dec. 1959). "Random Graphs". In: *The Annals of Mathematical Statistics* 30.4, pp. 1141–1144. DOI: 10.1214/aoms/1177706098. URL: https://doi.org/10.1214/aoms/1177706098.

Girvan, M. and M. E. J. Newman (June 2002). "Community structure in social and biological networks". In: *Proceedings of the National Academy of Sciences* 99.12, pp. 7821–7826. DOI: 10.1073/pnas.122653799. URL: https://doi.org/10.1073/pnas.122653799.

Goldschmidt, O. and D. S. Hochbaum (Oct. 1988). "Polynomial algorithm for the k-cut problem". In: *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pp. 444–451. DOI: 10.1109/SFCS.1988.21960.

González, Marta C., César A. Hidalgo, and Albert-László Barabási (June 2008). "Understanding individual human mobility patterns". In: *Nature* 453.7196, pp. 779–782. DOI: 10.1038/nature06958. URL: https://doi.org/10.1038/nature06958.

Green, R.A. et al. (Apr. 2011). "A high-resolution C. elegans essential gene network based on phenotypic profiling of a complex tissue". In: *Cell* 45.

Guimerà, R., M. Sales-Pardo, and L. A. N. Amaral (Aug. 2004). "Modularity from fluctuations in random graphs and complex networks". In: *Phys. Rev. E* 70.2, p. 025101. DOI: 10.1103/PhysRevE.70.025101. eprint: cond-mat/0403660.

Guimerà, Roger and Marta Sales-Pardo (2009). "Missing and spurious interactions and the reconstruction of complex networks". In: *Proceedings of the National Academy of Sciences* 106.52, pp. 22073–22078. DOI: 10.1073/pnas.0908366106. URL: https://doi.org/10.1073/pnas.0908366106.

Guimerà, Roger, Marta Sales-Pardo, and Luís A. N. Amaral (Dec. 2006). "Classes of complex networks defined by role-to-role connectivity profiles". In: *Nature*

Physics 3.1, pp. 63–69. DOI: 10.1038/nphys489. URL: https://doi.org/10.1038/nphys489.

Haijun, Zhou and Lipowsky Reinhard (2004). "Network Brownian Motion: A New Method to Measure Vertex-Vertex Proximity and to Identify Communities and Sub-communities". In: *International Conference on Computational Science*.

Harel, David and Yehuda Koren (2001). "On Clustering Using Random Walks". In: *FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science*. Springer Berlin Heidelberg, pp. 18–41. DOI: 10.1007/3-540-45294-x_3. URL: https://doi.org/10.1007/3-540-45294-x_3.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009a). *The Elements of Statistical Learning*. Springer-Verlag New York. ISBN: 978-0-387-84857-0. DOI: 10.1007/978-0-387-84858-7.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (Dec. 2009b). *Unsupervised Learning - The Elements of Statistical Learning*. Springer New York, pp. 485–585. DOI: 10.1007/978-0-387-84858-7_14.

Hizanidis, Johanne, Nikos E. Kouvaris, Gorka Zamora-López, Albert Diaz-Guilera, and Chris G. Antonopoulos (Jan. 2016). "Chimera-like States in Modular Neural Networks". In: *Scientific Reports* 19845. DOI: http://dx.doi.org/10.1038/srep19845.

Hric, Darko, Richard K. Darst, and Santo Fortunato (Dec. 2014). "Community detection in networks: Structural communities versus ground truth". In: *Physical Review E* 90.6. DOI: 10.1103/physreve.90.062805. URL: https://doi.org/10.1103/physreve.90.062805.

Hubert, Lawrence and Phipps Arabie (Dec. 1985). "Comparing partitions". In: *Journal of Classification* 2.1, pp. 193–218. DOI: 10.1007/bf01908075. URL: https://doi.org/10.1007/bf01908075.

Jeong, H., S. P. Mason, A.-L. Barabási, and Z. N. Oltvai (May 2001). "Lethality and centrality in protein networks". In: *Nature* 411, pp. 41–42. DOI: 10.1038/35075138.

Jerome, Kunegis (2013). "The Koblenz Network Collection". In: *Proceedings Conference on World Wide Web Companion*, pp. 1343–1350. URL: http://konect.uni-koblenz.de.

Joe H. Ward, Jr. (1963). "Hierarchical Grouping to Optimize an Objective Function". In: *Journal of the American Statistical Association* 58.301, pp. 236–244. DOI: 10.1080/01621459.1963.10500845.

Karrer, Brian and M. E. J. Newman (Jan. 2011). "Stochastic blockmodels and community structure in networks". In: *Physical Review E* 83.1. DOI: 10.1103/physreve.83.016107. URL: https://doi.org/10.1103/physreve.83.016107.

Katz, Leo (Mar. 1953). "A new status index derived from sociometric analysis". In: *Psychometrika* 18.1, pp. 39–43. DOI: 10.1007/bf02289026. URL: https://doi.org/10.1007/bf02289026.

Kernighan, B. W. and S. Lin (Feb. 1970). "An Efficient Heuristic Procedure for Partitioning Graphs". In: *Bell System Technical Journal* 49.2, pp. 291–307. DOI: 10.1002/j.1538-7305.1970.tb01770.x. URL: https://doi.org/10.1002/j.1538-7305.1970.tb01770.x.

Kevin, Lewis, Kaufman Jason, and Christakis Nicholas (Nov. 2008). "The Taste for Privacy: An Analysis of College Student Privacy Settings in an Online Social Network". In: *Journal of Computer-Mediated Communication* 14.1, pp. 79–100. DOI: 10.1111/j.1083-6101.2008.01432.x.

Khan, B. S. and M. A. Niazi (Aug. 2017). "Network Community Detection: A Review and Visual Survey". In: *ArXiv e-prints*. arXiv: 1708.00977.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (May 1983). "Optimization by Simulated Annealing". In: *Science* 220.4598, pp. 671–680. DOI: 10.1126/science.220.4598.671. URL: https://doi.org/10.1126/science.220.4598.671.

Kleinberg, Jon M., Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins (1999). "The Web As a Graph: Measurements, Models, and Methods". In: *Proceedings of the 5th Annual International Conference on Computing and Combinatorics*. COCOON'99. Berlin, Heidelberg: Springer-Verlag, pp. 1–17. ISBN: 3-540-66200-6. URL: http://dl.acm.org/citation.cfm?id=1765751.1765753.

Klemm, Konstantin and Víctor M. Eguíluz (May 2002). "Growing scale-free networks with small-world behavior". In: *Physical Review E* 65.5. DOI: 10.1103/physreve.65.057102. URL: https://doi.org/10.1103/physreve.65.057102.

Kullback, S. and R. A. Leibler (Mar. 1951). "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1, pp. 79–86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

Kuncheva, L.I. and S.T. Hadjitodorov (2004). "Using diversity in cluster ensembles". In: *IEEE International Conference on Systems, Man and Cybernetics*. DOI: 10.1109/icsmc.2004.1399790. URL: https://doi.org/10.1109/icsmc.2004.1399790.

Labatut, V. and G. K. Orman (2017). "Community Structure Characterization". In: *Encyclopedia of Social Network Analysis and Mining*. Springer New York, pp. 1–13. DOI: 10.1007/978-1-4614-7163-9_110151-1. URL: https://doi.org/10.1007/978-1-4614-7163-9_110151-1.

Lambiotte, R. (2010). "Multi-scale modularity in complex networks". In: *8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pp. 546–553.

Lancichinetti, Andrea, Santo Fortunato, and Filippo Radicchi (Oct. 2008). "Benchmark graphs for testing community detection algorithms". In: *Physical Review E* 78.4. DOI: 10.1103/physreve.78.046110. URL: https://doi.org/10.1103/physreve.78.046110.

Lancichinetti, Andrea, Mikko Kivelä, Jari Saramäki, and Santo Fortunato (Aug. 2010). "Characterizing the Community Structure of Complex Networks". In: *PLoS ONE* 5.8. Ed. by Olaf Sporns, e11976. DOI: 10.1371/journal.pone.0011976. URL: https://doi.org/10.1371/journal.pone.0011976.

Lancichinetti, Andrea, Filippo Radicchi, José J. Ramasco, and Santo Fortunato (Apr. 2011). "Finding Statistically Significant Communities in Networks". In: *PLoS ONE* 6.4. Ed. by Eshel Ben-Jacob, e18961. DOI: 10.1371/journal.pone.0018961. URL: https://doi.org/10.1371/journal.pone.0018961.

Lanczos, C. (Oct. 1950). "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators". In: *Journal of Research of the National Bureau of Standards* 45.4, p. 255. DOI: 10.6028/jres.045.026. URL: https://doi.org/10.6028/jres.045.026.

László, Lovász (1993). "Random Walks on Graphs: A Survey". In: *Combinatorics, Paul Erős is Eighty* 2, pp. 1–46.

Leicht, E. A., Petter Holme, and M. E. J. Newman (Feb. 2006). "Vertex similarity in networks". In: *Physical Review E* 73.2. DOI: 10.1103/physreve.73.026120. URL: https://doi.org/10.1103/physreve.73.026120.

Leskovec, J., J. Kleinberg, and C. Faloutsos (Mar. 2007). "Graph Evolution: Densification and Shrinking Diameters". In: *ACM Transactions on Knowledge Discovery from Data (ACM TKDD)*.

Leskovec, Jure and Andrej Krevl (June 2014). "SNAP Datasets: Stanford Large Network Dataset Collection". In: URL: http://snap.stanford.edu/data.

Leskovec, Jure, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney (2008). "Statistical properties of community structure in large social and information networks". In: *Proceeding of the 17th international conference on World Wide Web - WWW 08*. DOI: 10.1145/1367497.1367591. URL: https://doi.org/10.1145/1367497.1367591.

Li, Zhenping, Shihua Zhang, Rui-Sheng Wang, Xiang-Sun Zhang, and Luonan Chen (Mar. 2008). "Quantitative function for community detection". In: *Physical Review E* 77.3. DOI: 10.1103/physreve.77.036109. URL: https://doi.org/10.1103/physreve.77.036109.

Lichtenberg, Ulrik de, Lars Juhl Jensen, Søren Brunak, and Peer Bork (2005). "Dynamic Complex Formation During the Yeast Cell Cycle". In: *Science* 307.5710, pp. 724–727. ISSN: 0036-8075. DOI: 10.1126/science.1105103. eprint: http://science.sciencemag.org/content/307/5710/724.full.pdf. URL: http://science.sciencemag.org/content/307/5710/724.

Magoni, Damien and Jean Jacques Pansiot (July 2001). "Analysis of the Autonomous System Network Topology". In: *SIGCOMM Comput. Commun. Rev.* 31.3, pp. 26–37. ISSN: 0146-4833. DOI: 10.1145/505659.505663. URL: http://doi.acm.org/10.1145/505659.505663.

Malliaros, Fragkiskos D. and Michalis Vazirgiannis (Dec. 2013). "Clustering and community detection in directed networks: A survey". In: *Physics Reports* 533.4, pp. 95–142. DOI: 10.1016/j.physrep.2013.08.002. URL: https://doi.org/10.1016/j.physrep.2013.08.002.

Marcus, Kaiser (2008). "Mean clustering coefficients: the role of isolated nodes and leafs on clustering measures for small-world networks". In: *New Journal of Physics* 10.8, p. 083042. URL: http://stacks.iop.org/1367-2630/10/i=8/a=083042.

Mary, McGlohon, Akoglu Leman, and Faloutsos Christos (2011). "Statistical Properties of Social Networks". In: *Social Network Data Analytics*. Springer US, pp. 17–42. DOI: 10.1007/978-1-4419-8462-3_2.

Masuda, Naoki and Renaud Lambiotte (Apr. 2016). *A Guide to Temporal Networks*. WORLD SCIENTIFIC (EUROPE). DOI: 10.1142/q0033. URL: https://doi.org/10.1142/q0033.

Matthew, Jackson (2008). *Social and economic networks*. Princeton, NJ: Princeton University Press. ISBN: 978-0691148205.

McCallum, Andrew, Xuerui Wang, and Andrés Corrada-Emmanuel (Oct. 2007). "Topic and Role Discovery in Social Networks with Experiments on Enron and Academic Email". In: *Journal of Artificial Intelligence Research* 30.1, pp. 249–272. ISSN: 1076-9757. URL: http://dl.acm.org/citation.cfm?id=1622637.1622644.

Meilă, Marina (2003). "Comparing Clusterings by the Variation of Information". In: *Learning Theory and Kernel Machines*, pp. 173–187. DOI: 10.1007/978-3-540-45167-9_14. URL: https://doi.org/10.1007/978-3-540-45167-9_14.

Meo, Pasquale De, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti (Feb. 2014). "Mixing local and global information for community detection in large networks". In: *Journal of Computer and System Sciences* 80.1, pp. 72–87. DOI: 10.1016/j.jcss.2013.03.012. URL: https://doi.org/10.1016/j.jcss.2013.03.012.

Miritello, Giovanna, Rubén Lara, Manuel Cebrian, and Esteban Moro (June 2013). "Limited communication capacity unveils strategies for human interaction". In: *Scientific Reports* 3.1950. DOI: doi:10.1038/srep01950.

Mislove, Alan, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee (2007). "Measurement and analysis of online social networks". In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM Press. DOI: 10.1145/1298306.1298311. URL: https://doi.org/10.1145/1298306.1298311.

Miyauchi, Atsushi and Yasushi Kawase (Jan. 2016). "Z-Score-Based Modularity for Community Detection in Networks". In: *PLOS ONE* 11.1. Ed. by Frederic Amblard, e0147805. DOI: 10.1371/journal.pone.0147805. URL: https://doi.org/10.1371/journal.pone.0147805.

Moreno, Jacob Levy and Helen Hall Jennings (1934). *Who shall survive?: A new approach to the problem of human interrelations*. Beacon House, Beacon, NY.

Nadler, Boaz, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis (July 2006). "Diffusion maps, spectral clustering and reaction coordinates of dynamical systems". In: *Applied and Computational Harmonic Analysis* 21.1, pp. 113–127. DOI: 10.1016/j.acha.2005.07.004. URL: https://doi.org/10.1016/j.acha.2005.07.004.

Nascimento, Mariá C.V. and André C.P.L.F. de Carvalho (June 2011). "Spectral methods for graph clustering – A survey". In: *European Journal of Operational Research* 211.2, pp. 221–231. DOI: 10.1016/j.ejor.2010.08.012. URL: https://doi.org/10.1016/j.ejor.2010.08.012.

Newman, M. E. J. (June 2001a). "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality". In: *Physical Review E* 64.1. DOI: 10.1103/physreve.64.016132. URL: https://doi.org/10.1103/physreve.64.016132.

Newman, M. E. J. (2001b). "The structure of scientific collaboration networks". In: *Proceedings of the National Academy of Sciences* 98.2, pp. 404–409. ISSN: 0027-8424. DOI: 10.1073/pnas.98.2.404. URL: http://www.pnas.org/content/98/2/404.

Newman, M. E. J. (Jan. 2003). "The Structure and Function of Complex Networks". In: *SIAM Review* 45, pp. 167–256. DOI: 10.1137/S003614450342480. eprint: cond-mat/0303516.

Newman, M. E. J. (June 2004). "Fast algorithm for detecting community structure in networks". In: *Physical Review E* 69.6. DOI: 10.1103/physreve.69.066133. URL: https://doi.org/10.1103/physreve.69.066133.

Newman, M. E. J. (Sept. 2005). "Power laws, Pareto distributions and Zipf's law". In: *Contemporary Physics* 46, pp. 323–351. DOI: 10.1080/00107510500052444. eprint: cond-mat/0412004.

Newman, M. E. J (2006). "Modularity and community structure in networks". In: *Proceedings of the National Academy of Sciences* 103.23, pp. 8577–8582. ISSN: 0027-8424. DOI: 10.1073/pnas.0601602103. URL: http://www.pnas.org/content/103/23/8577.

Newman, M. E. J. (2008). "Mathematics of Networks". In: *The New Palgrave Dictionary of Economics*. Palgrave Macmillan UK, pp. 1–8. DOI: 10.1057/978-1-349-95121-5_2565-1. URL: https://doi.org/10.1057/978-1-349-95121-5_2565-1.

Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford University Press. DOI: 10.1093/acprof:oso/9780199206650.001.0001.

Newman, M. E. J. and M. Girvan (Feb. 2004). "Finding and evaluating community structure in networks". In: *Physical Review E* 69.2. DOI: 10.1103/physreve.69.026113. URL: https://doi.org/10.1103/physreve.69.026113.

Nils, Johnsson (2014). "Analyzing protein–protein interactions in the post-interactomic era. Are we ready for the endgame?" In: *Biochemical and Biophysical Research Communications* 445.4. Advances in OMICs-based disciplines, pp. 739–745. ISSN: 0006-291X. DOI: https://doi.org/10.1016/j.bbrc.2014.02.023. URL: http://www.sciencedirect.com/science/article/pii/S0006291X14002757.

Onnela, J.-P., J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabási (2007). "Structure and tie strengths in mobile communication networks". In: *Proceedings of the National Academy of Sciences* 104.18, pp. 7332–7336. ISSN: 0027-8424. DOI: 10.1073/pnas.0610245104. eprint: http://www.pnas.org/content/104/18/7332.full.pdf. URL: http://www.pnas.org/content/104/18/7332.

Orman, Günce Keziban, Vincent Labatut, and Hocine Cherifi (Aug. 2012). "Comparative evaluation of community detection algorithms: a topological approach". In: *Journal of Statistical Mechanics: Theory and Experiment* 2012.08, P08001. DOI: 10.1088/1742-5468/2012/08/p08001. URL: https://doi.org/10.1088/1742-5468/2012/08/p08001.

Page, Larry, Sergey Brin, R. Motwani, and T. Winograd (1998). *The PageRank Citation Ranking: Bringing Order to the Web*.

Papadopoulos, Symeon, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos (June 2011). "Community detection in Social Media". In: *Data Mining and Knowledge Discovery* 24.3, pp. 515–554. DOI: 10.1007/s10618-011-0224-z. URL: https://doi.org/10.1007/s10618-011-0224-z.

Papadopoulos, Symeon, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos (May 2012). "Community Detection in Social Media". In: *Data Min. Knowl. Discov.* 24.3, pp. 515–554. ISSN: 1384-5810. DOI: 10.1007/s10618-011-0224-z. URL: http://dx.doi.org/10.1007/s10618-011-0224-z.

Pareto, Vilfredo (1964). *Cours d'Économie Politique*. Librairie Droz, Geneva.

Pastor-Satorras, Romualdo and Alessandro Vespignani (2004). *Evolution and Structure of the Internet: A Statistical Physics Approach*. New York, NY, USA: Cambridge University Press. ISBN: 0521826985.

Peel, Leto, Daniel B. Larremore, and Aaron Clauset (May 2017). "The ground truth about metadata and community detection in networks". In: *Science Advances* 3.5, e1602548. DOI: 10.1126/sciadv.1602548. URL: https://doi.org/10.1126/sciadv.1602548.

Peixoto, Tiago P. (2018). "Bayesian stochastic blockmodeling". In: *Advances in Network Clustering and Blockmodeling*.

Pons, Pascal and Matthieu Latapy (2005). "Computing Communities in Large Networks Using Random Walks". In: *Computer and Information Sciences - ISCIS 2005*. Ed. by pInar Yolum, Tunga Güngör, Fikret Gürgen, and Can Özturan. Springer Berlin Heidelberg, pp. 284–293. ISBN: 978-3-540-32085-2.

Pons, Pascal and Matthieu Latapy (2011). "Post-processing hierarchical community structures: Quality improvements and multi-scale view". In: *Theoretical Computer Science* 412.8-10, pp. 892–900. DOI: 10.1016/j.tcs.2010.11.041. URL: https://doi.org/10.1016/j.tcs.2010.11.041.

Porta, S., P. Crucitti, and V. Latora (Sept. 2006). "The Network Analysis of Urban Streets: A Dual Approach". In: *Science* 369 (2). DOI: https://doi.org/10.1016/j.physa.2005.12.063.

Porter, M. A., J.-P. Onnela, and P. J. Mucha (Feb. 2009). "Communities in Networks". In: *Notices of the American Mathematical Society* 9.

Pothen, Alex, Horst D. Simon, and Kang-Pu Liou (July 1990). "Partitioning Sparse Matrices with Eigenvectors of Graphs". In: *SIAM Journal on Matrix Analysis and*

Applications 11.3, pp. 430–452. DOI: 10.1137/0611030. URL: https://doi.org/10.1137/0611030.

Prim, R.C. (Nov. 1957). "Shortest connection networks and some generalizations". In: *The Bell System Technical Journal* 36.6, pp. 1389–1401. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1957.tb01515.x.

Rabbat, M. G., J. R. Treichler, S. L. Wood, and M. G. Larimore (Mar. 2005). "Understanding the topology of a telephone network via internally-sensed network tomography". In: *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* Vol. 3, iii/977–iii/980 Vol. 3. DOI: 10.1109/ICASSP.2005.1415875.

Radicchi, F., C. Castellano, F. Cecconi, V. Loreto, and D. Parisi (Feb. 2004). "Defining and identifying communities in networks". In: *Proceedings of the National Academy of Sciences* 101.9, pp. 2658–2663. DOI: 10.1073/pnas.0400054101. URL: https://doi.org/10.1073/pnas.0400054101.

Raghavan, Usha Nandini, Réka Albert, and Soundar Kumara (Sept. 2007). "Near linear time algorithm to detect community structures in large-scale networks". In: *Physical Review E* 76.3. DOI: 10.1103/physreve.76.036106. URL: https://doi.org/10.1103/physreve.76.036106.

Rand, William M. (Dec. 1971). "Objective Criteria for the Evaluation of Clustering Methods". In: *Journal of the American Statistical Association* 66.336, pp. 846–850. DOI: 10.1080/01621459.1971.10482356. URL: https://doi.org/10.1080/01621459.1971.10482356.

Ravasz, E., A.L. Somera, D.A. Mongru, Z.N. Oltvai, and A.L. Barabasi (Aug. 2002). "Hierarchical organization of modularity in metabolic networks". In: *Science* 297.5586.

Reichardt, Jörg, Roberto Alamino, and David Saad (2011). "The Interplay between Microscopic and Mesoscopic Structures in Complex Networks". In: *PLoS ONE* 6.8. Ed. by Olaf Sporns, e21282. DOI: 10.1371/journal.pone.0021282. URL: https://doi.org/10.1371/journal.pone.0021282.

Reichardt, Jörg and Stefan Bornholdt (July 2006). "Statistical mechanics of community detection". In: *Physical Review E* 74.1. DOI: 10.1103/physreve.74.016110. URL: https://doi.org/10.1103/physreve.74.016110.

Riolo, Maria A., George T. Cantwell, Gesine Reinert, and M. E. J. Newman (Sept. 2017). "Efficient method for estimating the number of communities in a network". In: *Physical Review E* 96.3. DOI: 10.1103/physreve.96.032310. URL: https://doi.org/10.1103/physreve.96.032310.

Rossi, Ryan A. and Nesreen K. Ahmed (2015). "The Network Data Repository with Interactive Graph Analytics and Visualization". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. URL: http://networkrepository.com.

Rosvall, M., D. Axelsson, and C. T. Bergstrom (Nov. 2009). "The map equation". In: *European Physical Journal Special Topics* 178, pp. 13–23. DOI: 10.1140/epjst/e2010-01179-1. arXiv: 0906.1405.

Rosvall, M. and C. T. Bergstrom (Apr. 2007). "An information-theoretic framework for resolving community structure in complex networks". In: *Proceedings of the National Academy of Sciences* 104.18, pp. 7327–7331. DOI: 10.1073/pnas.0611034104. URL: https://doi.org/10.1073/pnas.0611034104.

Rosvall, Martin and Carl T. Bergstrom (2008). "Maps of random walks on complex networks reveal community structure". In: *Proceedings of the National Academy of Sciences* 105.4, pp. 1118–1123. ISSN: 0027-8424. DOI: 10.1073/pnas.0706851105. URL: http://www.pnas.org/content/105/4/1118.

Rosvall, Martin and Carl T. Bergstrom (Apr. 2011). "Multilevel Compression of Random Walks on Networks Reveals Hierarchical Organization in Large Integrated

Systems". In: *PLoS ONE* 6.4. Ed. by Fabio Rapallo, e18209. DOI: 10.1371/journal.pone.0018209. URL: https://doi.org/10.1371/journal.pone.0018209.

Sabidussi, Gert (Dec. 1966). "The centrality index of a graph". In: *Psychometrika* 31.4, pp. 581–603. DOI: 10.1007/bf02289527. URL: https://doi.org/10.1007/bf02289527.

Saitou, N. and M. Nei (July 1987). "The neighbor-joining method: a new method for reconstructing phylogenetic trees." In: *Molecular Biology and Evolution*. DOI: 10.1093/oxfordjournals.molbev.a040454. URL: https://doi.org/10.1093/oxfordjournals.molbev.a040454.

Schaeffer, Satu Elisa (Aug. 2007). "Graph clustering". In: *Computer Science Review* 1.1, pp. 27–64. DOI: 10.1016/j.cosrev.2007.05.001. URL: https://doi.org/10.1016/j.cosrev.2007.05.001.

Schaub, Michael T., Jean-Charles Delvenne, Martin Rosvall, and Renaud Lambiotte (Feb. 2017). "The many facets of community detection in complex networks". In: *Applied Network Science* 2.1. DOI: 10.1007/s41109-017-0023-6. URL: https://doi.org/10.1007/s41109-017-0023-6.

Sekara, Vedran, Arkadiusz Stopczynski, and Sune Lehmann (2016). "Fundamental structures of dynamic social networks". In: *Proceedings of the National Academy of Sciences* 113.36, pp. 9977–9982. ISSN: 0027-8424. DOI: 10.1073/pnas.1602803113. URL: http://www.pnas.org/content/113/36/9977.

Sen, Parongama, Subinay Dasgupta, Arnab Chatterjee, P. A. Sreeram, G. Mukherjee, and S. S. Manna (Mar. 2003). "Small-world properties of the Indian railway network". In: *Phys. Rev. E* 67 (3), p. 036106. DOI: 10.1103/PhysRevE.67.036106. URL: https://link.aps.org/doi/10.1103/PhysRevE.67.036106.

Shannon, C. E. (July 1948). "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x. URL: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. London New York: Chapman and Hall. ISBN: 0412246201.

Sozio, Mauro and Aristides Gionis (2010). "The community-search problem and how to plan a successful cocktail party". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD10*. ACM Press. DOI: 10.1145/1835804.1835923. URL: https://doi.org/10.1145/1835804.1835923.

Stanley, Milgram (Dec. 1967). "The Small World Problem". In: *Psychology today*, pp. 515–554.

Stelling, J., S. Klamt, K. Bettenbrock, S. Schuster, and E.D. Gilles (Nov. 2002). "Metabolic network structure determines key aspects of functionality and regulation". In: *Nature* 420.6912.

Su, Jessica, Aneesh Sharma, and Sharad Goel (2016). "The Effect of Recommendations on Network Structure". In: *Proceedings of the 25th International Conference on World Wide Web*. WWW '16, pp. 1157–1167. ISBN: 978-1-4503-4143-1. DOI: 10.1145/2872427.2883040. URL: https://doi.org/10.1145/2872427.2883040.

Subbian, Karthik, Charu C. Aggarwal, and Jaideep Srivastava (2016). "Querying and Tracking Influencers in Social Streams". In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. WSDM '16. ACM, pp. 493–502. ISBN: 978-1-4503-3716-8. DOI: 10.1145/2835776.2835788. URL: http://doi.acm.org/10.1145/2835776.2835788.

Traag, V. A., R. Aldecoa, and J.-C. Delvenne (Aug. 2015). "Detecting communities using asymptotical surprise". In: *Physical Review E* 92.2. DOI: 10.1103/physreve.92.022816. URL: https://doi.org/10.1103/physreve.92.022816.

Traag, V. A., P. Van Dooren, and Y. Nesterov (July 2011). "Narrow scope for resolution-limit-free community detection". In: *Physical Review E* 84.1. DOI: 10.1103/physreve.84.016114. URL: https://doi.org/10.1103/physreve.84.016114.

Treichler, J. R., M. G. Larimore, S. L. Wood, and M. Rabbat (Aug. 2004). "Determining the topology of a telephone system using internally sensed network tomography". In: *3rd IEEE Signal Processing Education Workshop. 2004 IEEE 11th Digital Signal Processing Workshop, 2004.* Pp. 259–262. DOI: 10.1109/DSPWS.2004.1437954.

Van Laarhoven, Twan and Elena Marchiori (Jan. 2014). "Axioms for Graph Clustering Quality Functions". In: *J. Mach. Learn. Res.* 15.1, pp. 193–215. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=2627435.2627441.

Vázquez, Alexei, Romualdo Pastor-Satorras, and Alessandro Vespignani (June 2002). "Large-scale topological and dynamical properties of the Internet". In: *Phys. Rev. E* 65 (6), p. 066130. DOI: 10.1103/PhysRevE.65.066130. URL: https://link.aps.org/doi/10.1103/PhysRevE.65.066130.

Vinh, Nguyen Xuan, Julien Epps, and James Bailey (Dec. 2010). "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance". In: *J. Mach. Learn. Res.* 11, pp. 2837–2854. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1756006.1953024.

Von Luxburg, Ulrike, Agnes Radl, and Matthias Hein (Jan. 2014). "Hitting and Commute Times in Large Random Neighborhood Graphs". In: *J. Mach. Learn. Res.* 15.1, pp. 1751–1798. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=2627435.2638591.

Wasserman, Stanley (1994). *Social network analysis : methods and applications*. Cambridge New York: Cambridge University Press. ISBN: 9780521387071.

Watts, Duncan J. (1999). *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton, NJ, USA: Princeton University Press. ISBN: 0-691-00541-9.

Watts, Duncan J. and Steven H. Strogatz (June 1998). "Collective dynamics of 'small-world' networks". In: *Nature* 393.6684, pp. 440–442. DOI: 10.1038/30918. URL: https://doi.org/10.1038/30918.

White, J.G., E. Southgate, J.N. Thomson, and S. Brenner (1986). "The structure of the nervous system of the nematode Caenorhabditis elegans". In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 314.1165, pp. 1–340. ISSN: 0080-4622. DOI: 10.1098/rstb.1986.0056.

Wunderlich, Z. and L.A. Mirny (Sept. 2006). "Using the topology of metabolic networks to predict viability of mutant strains". In: *Biophysical Journal* 91.2304.

Xie, Jierui and Boleslaw K. Szymanski (June 2011). "Community detection using a neighborhood strength driven Label Propagation Algorithm". In: *2011 IEEE Network Science Workshop*. IEEE. DOI: 10.1109/nsw.2011.6004645. URL: https://doi.org/10.1109/nsw.2011.6004645.

Xie, Jierui and Boleslaw K. Szymanski (2012). "Towards Linear Time Overlapping Community Detection in Social Networks". In: *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, pp. 25–36. DOI: 10.1007/978-3-642-30220-6_3. URL: https://doi.org/10.1007/978-3-642-30220-6_3.

Yang, J. and J. Leskovec (May 2012). "Defining and Evaluating Network Communities based on Ground-truth". In: *Proceedings of 2012 IEEE International Conference on Data Mining (ICDM)*.

Yang, J., J. McAuley, and J. Leskovec (Dec. 2013). "Community Detection in Networks with Node Attributes". In: *2013 IEEE 13th International Conference on Data Mining*, pp. 1151–1156. DOI: 10.1109/ICDM.2013.167.

Yang, Jaewon and Jure Leskovec (Oct. 2013). "Defining and evaluating network communities based on ground-truth". In: *Knowledge and Information Systems* 42.1, pp. 181–213. DOI: 10.1007/s10115-013-0693-z. URL: https://doi.org/10.1007/s10115-013-0693-z.

Yang, Zhao, René Algesheimer, and Claudio J. Tessone (2016). "A Comparative Analysis of Community Detection Algorithms on Artificial Networks". In: *Scientific Reports* 6.1. DOI: 10.1038/srep30750. URL: https://doi.org/10.1038/srep30750.

Zachary, Wayne W. (Dec. 1977). "An Information Flow Model for Conflict and Fission in Small Groups". In: *Journal of Anthropological Research* 33.4, pp. 452–473. DOI: 10.1086/jar.33.4.3629752. URL: https://doi.org/10.1086/jar.33.4.3629752.

Zipf, George Kingsley (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Reading, MA.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Évaluation de structures de communautés

Mot clés : Structure communautaire, Evaluation, Caractérisation, Aide à la décision

Resumé : La détection de communauté est une technique qui décompose des graphes en sous-graphes densément connectés, ce qui est particulièrement utile dans le cas de (très) grands réseaux complexes dont la visualisation est difficile. De très nombreuses méthodes, très variées, ont été proposées ces dernières années. Dans un contexte où aucun consensus n'émerge autour de la notion même de communauté, ces méthodes provoquent de multiples discussions scientifiques autour de la qualité de leur résultat. Dans cette thèse, nous proposons plusieurs types d'évaluation comparative et approfondie de 16 méthodes bien connues de l'état de l'art ainsi que la caractérisation exhaustive des structures communautaires découvertes dans des réseaux réels variés provenant de domaines différents. Nos résultats — méthodes et analyses — constituent un début de boîte à outils pour l'analyste bien en peine de choisir la méthode adaptée à son étude.

Title : Evaluation of community structures

Keywords : Community structure, Evaluation, Characterization, Decision Aid

Abstract : Community detection is a technique used to separate graphs into several densely connected groups of vertices, especially powerful when visualization techniques are infeasible for large-scale structures of networks. Thanks to a plethora of potential applications in the golden age of social interaction, many detection techniques have been invented in the last decades. Their performance in discovering significant structures has been a hot topic in the network science community since there is still no consensus on what good communities are. In this dissertation, we invite readers to go through several comprehensive analyses of various state-of-the-art community detection methods as well as modular structures of real networks belonging to a large variety of domains. Our results provide intuitive illustrations of community structures and useful information that helps readers to choose their context-based rule-of-thumb solution.