Overlapping community detection in dynamic networks

Qinna Wang

THÈSE

en vue d'obtenir le grade de
Docteur de l'Ecole Normale Supérieure de Lyon
Spécialité Informatique

Université de Lyon
Laboratoire de l'Informatique du Parallélisme
École doctorale Informatique et Mathématiques

présentée et soutenue publiquement
le 12 Avril 2012
par

Qinna Wang

Détection de communautés recouvrantes dans des réseaux de terrain dynamiques

---

devant la commission d'examen composée de

Rapporteurs:	Anne-Marie Kermarrec, DR, Inria Rennes-Bretagne Atlantique
	Matthieu Latapy, DR, CNRS / UPMC
Examinateurs:	Pablo Jensen, DR, CNRS / ENS de Lyon
	Bertrand Jouve, Professeur, Université Lumière Lyon 2
	Christophe Prieur, MdC, Université Paris Diderot
Directeur de thèse:	Eric Fleury, Professeur, ENS de Lyon

Abstract

In complex networks, the notion of community structure refers to the presence of groups of nodes in a network. These groups are more densely connected internally than with the rest of the network. The presence of communities inside a network gives an insight on network structural properties. For example, in social networks, communities are based on common interests, location, hobbies.... Generally, a community structure is described by a partition of the network nodes, where each node belongs to a unique community. A more reasonable description seems to be overlapping community structure, where nodes are allowed to be shared by several communities. Moreover, when considering dynamic networks whose interactions between nodes evolve in time, it appears crucial to consider also the evolution of the intrinsic community structure.

This thesis focus on mining dynamic community evolution and overlapping community detection. We have proposed two distinct methods for overlapping community detection. The first one named clique optimization and the second one called fuzzy detection. Our clique optimization aims to identify granular overlaps and it is a fine grain scale approach. Our fuzzy detection is at a coarser grain scale with the strategy of identifying modular overlaps. Their applications in synthetic and real networks indicate that both methods can be used for characterizing overlapping nodes but in distinct and complementary views. We also propose the definition of predecessor and successor in mining community evolution. Such definition describes the relationship between communities at different time steps. We use it to detect community evolution in dynamic networks and show how modular overlaps evolve over time. A visualization tool called lineage diagrams is used to show community evolution by connecting communities in relationship of predecessor and successor. Several cases are studied.

Keywords: community detection, overlapping community structure, complex networks, dynamic networks, community evolution, network science

Résumé

Dans le contexte des réseaux complexes, la structure communautaire du réseau devient un sujet important pour plusieurs domaines de recherche. Les communautés sont en général vues comme des groupes intérieurement denses. La détection de tels groupes offre un éclairage intéressant sur la structure du réseau. Par exemple, une communauté de pages web regroupe des pages traitant du même sujet. La définition de communautés est en général limitée à une partition de l'ensemble des nœuds. Cela exclut par définition qu'un nœud puisse appartenir à plusieurs communautés, ce qui pourtant est naturel dans de nombreux (cas des réseaux sociaux par exemple). Une autre question importante et sans réponse est l'étude des réseaux et de leur structure communautaire en tenant compte de leur dynamique. Cette thèse porte sur l'étude de réseaux dynamiques et la détection de communautés recouvrantes.

Nous proposons deux méthodes différentes pour la détection de communautés recouvrantes. La première méthode est appelée optimisation de clique. L'optimisation de clique vise à détecter les nœuds recouvrants granulaires. La méthode de l'optimisation de clique est une approche à grain fin. La seconde méthode est nommée détection floue (fuzzy detection). Cette méthode est à grain plus grossier et vise à identifier les groupes recouvrants. Nous appliquons ces deux méthodes à des réseaux synthétiques et réels. Les résultats obtenus indiquent que les deux méthodes peuvent être utilisées pour caractériser les nœuds recouvrants. Les deux approches apportent des points de vue distincts et complémentaires. Dans le cas des graphes dynamiques, nous donnons une définition sur la relation entre les communautés à deux pas de temps consécutif. Cette technique permet de représenter le changement de la structure en fonction du temps. Pour mettre en évidence cette relation, nous proposons des diagrammes de lignage pour la visualisation de la dynamique des communautés. Ces diagrammes qui connectent des communautés à des pas de temps successifs montrent l'évolution de la structure et l'évolution des groupes recouvrantes. Nous avons également appliquer ces outils à des cas concrets.

Mots clés: structure communautaire, communautés recouvrantes, réseaux de terrain dynamiques, évolution de communautés, réseaux complexes

Remerciements

Je remercie, en premier lieu, mes rapporteurs, Mme Anne-Marie Kermarrec et Mr. Matthieu Latapy, pour avoir accepté la tâche de rapporter ma thèse. Je remercie également Mr. Christophe Prieur, M. Bertrand Jouve et Mr. Pablo Jensen d'avoir accepté de prendre part à mon jury.

Au cours des trois années qu'a duré ma thèse, j'ai eu le plaisir de rencontrer un grand nombre de chercheurs, d'enseignants et d'étudiants, avec lesquels j'ai eu le très grand plaisir de travailler tout au long de cette thèse. C'est à eux que vont mes plus grands remerciements. Je leur adresse ma gratitude pour l'attention, les conseils, les encouragements, la patience, l'amabilité dont ils ont fait preuve tout au long de mes travaux.

Je remercie, M. Eric Fleury pour m'avoir offert l'opportunité de réaliser cette thèse. Il a dirigé mon travail avec une grande justesse, une profonde humanité et beaucoup de gaieté, ce pour quoi je le remercie infiniment.

Je remercie en particulier les membres de l'équipe D-NET : Guillaume Chelius, Andreea Chis, Christophe Crespelle, Adrien Friggeri, Lucie Martinet, Sèverine Morin et Evelyne Blesle. Je remercie chaleureusement mes collègues de l'Université de Paris 6: Jean-Loup Guillaume et Thomas Aynaud.

Enfin, je souhaiterais remercier mes parents, tous les amis et les personnes qui ont toujours encouragé et accompagné mon travail de leur pensée positive.

Contents

Introduction **9**

1 Community detection in dynamic networks **13**
1.1 Communities in networks 13
1.1.1 Definitions of community 14
1.1.2 Community structure 15
1.2 Community evolution in dynamic networks 18
1.2.1 Communities in dynamic graphs 19
1.2.2 Community dynamics 19
1.3 Community detection in dynamic networks 22
1.3.1 Dynamic community detection challenges 24
1.3.2 Two-stage approaches 27
1.3.3 Evolutionary clustering 31
1.3.4 Coupling graph clustering 34
1.4 Benchmarks 35
1.4.1 Benchmark graphs 35
1.4.2 Real networks 37
1.4.3 Comparing partitions 39
1.5 Conclusion 41

2 Overlapping communities and modularity **42**
2.1 Related work on cover detection 43
2.2 Modified modularity for covers 44
2.2.1 A novel modularity 44
2.2.2 Existing modularity for covers 48
2.3 Clique optimization 50
2.3.1 Our proposed definition 50
2.3.2 The clique optimization algorithm 52
2.3.3 Benchmark graphs 54
2.4 Fuzzy detection 57

2.4.1 Motivation . . . 57
2.4.2 Fuzzy detection algorithm . . . 57
2.4.3 Benchmark graphs . . . 62
2.5 Application to real networks: Complex System Science . . . 65
2.6 Discussion . . . 68
2.6.1 Granular overlaps and the parameter k . . . 69
2.6.2 Community memberships and membership degree . . . 73
2.7 Conclusion . . . 73

3 Overlapping communities and community evolution 75
3.1 Tracking community evolution in dynamic networks . . . 76
3.1.1 Group persistence two-stage method . . . 77
3.1.2 Motivation . . . 78
3.1.3 Community dynamics . . . 79
3.1.4 Mapping method . . . 81
3.1.5 Visualizing community evolution . . . 83
3.2 Experimental results . . . 87
3.2.1 Synthetic datasets . . . 87
3.2.2 Blogs . . . 88
3.3 Application to a dynamic co-citation network . . . 90
3.3.1 Building a dynamic graph . . . 92
3.3.2 Detecting and visualizing community evolution . . . 92
3.3.3 Evaluating the results . . . 93
3.3.4 Study of the community evolution . . . 95
3.3.5 Robust cluster evolution and overlaps evolution . . . 97
3.3.6 Discussion and conclusion . . . 98

4 Conclusion 99
4.1 Summary . . . 99
4.2 Future works . . . 100

Bibliography 100

Annexes 112

A List of publications 113
A.1 International Conferences . . . 113
A.2 Poster . . . 113
A.3 Journals . . . 113
A.4 Seminars . . . 114

B Past history complex system science DATA set & keywords 115

Introduction

Modular organization of complex networks

Complex networks are obtained by modeling real systems with graphs. This paradigm is used to represent a wide variety of systems in different areas, such as the Internet [45], World Wide Web, citation networks [47], coauthorship networks [64], metabolic networks [56]. Each citizen, as an individual, can construct a social network whose nodes are connected by one or more specific types of relations, like friendship, kinship, common interest [6, 13].

Studies in complex networks become a popular interest of research area. It was triggered by two seminal papers: Watts and Strogatz on small-world networks [130] and Barabasi and Albert on scale-free networks [9]. These studies have introduced common non-trivial properties, which do not occur in simple networks such as lattices or random graphs. It induced a large development of work on the studies of properties of real networks.

The massive and comparative analysis of networks from several fields has produced a series of unexpected and impressive results. One important issue is community structure. Empirical studies on different networks such as World Wide Web, protein interaction networks, email networks, *etc.* find their degree distributions different from each other. Studies also find that the distribution of node degrees is not only globally, but also locally heterogeneous. In another words, networks can be characterized by communities, with dense connections within them and sparse connections between them.

The community structure of a real network is not only the result of the topology, but also refers to system functions: in protein-protein interaction networks, communities correspond to specific functions [21]; in the World Wide Web, they may relate to topics [30]; in food webs they correspond to compartments [66], *etc.* Studies in community structure should lead to a better understanding of complex systems.

Community detection

In order to detect community structures, diverse techniques are proposed and are applied to real networks. As early as 1955, Weiss and Jacobson [107] carried out the first analysis of community structure, which was at the basis of *graph partitioning*. *Graph partitioning* divides nodes into predefined communities, such that the number of edges lying between the groups is minimal. In a seminal paper appeared in 2001, Girvan and Newman [48] proposed a new algorithm, which identified edges lying between communities for successive removal until the isolation of communities. This paper triggered a big activity in the field, and many new modern methods have been proposed. For example, modularity optimization is the most popular method for community detection on large graphs [16, 91, 126], dynamic algorithms are based on physical techniques: spin models [116], random walks [102] and synchronization [2], and others like methods based on statistical inference: Bayesian inference [132], blockmodeling [10], model selection [17] and information theory.

These methods provide good performance in community detection, and have been applied to real networks for analysis. Is the subject of community detection deserving another report? At least two reasons have deeply motivated our work.

The first is that current complex networks become more complex, with the main focus moving from the analysis of small static networks to that of systems with thousands or millions of nodes, and with a renewed attention to the properties of networks of dynamical units. For instance, the network of communications of millions of users is changing its interactions across time. The structure of a real network is the result of the continuous evolution of interactions which correspond to system functions. So that the research on communities in dynamic networks would lead to a better knowledge of system evolutionary mechanisms, and to a better cottoning on dynamical and functional behaviours. Most of community detection methods are proposed for static networks. There is a crucial need for algorithms that detect communities in dynamic networks.

The second is that overlapping community structure is still a problem. Most of community detection methods are proposed to detect disjoint communities without *overlapping nodes*. *Overlapping nodes* are shared by several communities in overlapping community structure. They are interesting to investigate since they play a key role as intermediate between communities, with a special effect in predicting dynamic behaviors of individuals in networks. Studies [125] in histories of personnel ties among the largest enterprises in Hungary showed that overlapping nodes were possible mixing or recombining memberships of groups. The membership of a long duration community changed year by year. Some communities were built up through splitting and reuniting in an ongoing pattern. This phenomenon, indeed, represents a crucial feature of overlapping nodes in understanding structural organization of complex systems. Studying overlapping community structure of networks will be helpful to understand system dynamic mechanisms and predict future trends.

We explore this thesis to deal with the analysis of overlapping community structure

in different networks and their dynamics. For this, methods for overlapping community structure are proposed as well as approaches to track the evolution of these structures over time. To verify their applicability, the presented methods are applied to different real work data sets and the obtained results are evaluated.

Main contributions

The main contributions of this thesis are briefly summarized in the following.

- Two different views on overlapping node detection: In order to detect overlapping community structure and characterize overlapping nodes, we have proposed two definitions of overlapping nodes: granular overlaps and modular overlaps. Granular overlaps are a set of nodes, each of which connects several communities with high cohesion. Modular overlaps are a set of groups, each of which is a group of nodes having high community membership degree (how strong the group of nodes belongs to the community) with at least two communities.

  For the detection of granular overlaps, we have proposed clique optimization, which detects cliques k-adjacent to communities (A clique which does not belong to the community but shares at least $k-1$ common nodes). A granular overlapping node in a weak sense is the member of one clique, which is adjacent to other communities different from its community membership in the partition. A granular overlapping node in a strong sense is the member of one clique, which is adjacent to at least two communities simultaneously.

  By running the Louvain algorithm several times, we can compute the probability that pair of nodes appear in the same community. It allows us to detect robust clusters, which have high stability against random impacts as every pair of connected nodes has a high co-appearance probability. Furthermore, we are able to detect community cores and modular overlaps. The community core is the maximum robust cluster within one community. The modular overlaps is one robust cluster has the high co-appearance probability with several communities.

  The applications of both methods to benchmark graphs have a high agreement with the known community structure. We also apply them to a real network. In the experiments, we observe that both methods provide meaningful but different results in characterizing overlapping nodes.

- Tracking community evolution and identifying community dynamics: In order to track community evolution and identify community dynamics, we have proposed a two-stage method: we firstly apply our fuzzy community detection to detection community structure at each time step, and secondly establish the relationship between communities at different time steps through the definition of group persistence. As the definition of group persistence is used to establish the relationship between predecessor community and successor community, we are able to characterize community dynamics even if parts of the membership fluctuate. To further

analyze and explore community dynamics, we introduced a visualization technique called lineage diagrams. The lineage diagrams allow us to observe how stable communities hold their members over time and how structure changes in the evolution of communities. This approach has been applied to a dynamic co-citation network called historic complex system science. In the experiments, we have applied citation analysis to understand the history of complex system science over time.

An important advantage of our method is its efficiency in detecting and characterizing community dynamics in highly dynamic networks. Therefore, our method is desirable to detect and analyse the evolution of communities in large, noisy networks that exhibit a high number of changes over time.

Outline of this thesis

The thesis is organized as follows. Chapter 1 is the survey of community detection in dynamic networks. We describe the definition of community structure and how a community changes over time. Community detection in dynamic networks becomes a popular issue. This problem is very hard and not yet satisfactorily solved. We review the main algorithms designed for dynamic networks, which are based on techniques for static networks. We also discuss crucial issues like how methods should be tested and compared against each other.

Chapter 2 concerns on overlapping community detection. We discuss the importance of overlapping community structure in network analysis and limits of existing algorithms in practice. Then, we transform the problem of overlapping community detection to overlapping node detection, with the developed concept of overlapping nodes into overlapping granularity and overlapping clusters. Therefore, we proposed two distinct methods: clique optimization and fuzzy detection. One is to detect overlapping granularity and the other is to detect overlapping clusters. Applications of the both methods in synthetic networks and real networks have good performances. Particularly, applications in the network between articles, describing the common references of articles relevant to complex systems provide an impressive result: the both methods provide knowledge of intermediate between communities but different characteristics.

In Chapter 3 we consider overlapping community structure on dynamic networks and propose a method based on our previous work. The applications in dynamic networks such as the past history of complex system science, reveal overlapping nodes are important for structural functions and interactions between modules.

Finally, we end in Chapter 4 by concluding our work in community detection with the discussion in future work.

Chapter 1

A survey of community detection in dynamic networks

The material in this chapter is intended to serve as a brief description of recent developments in community detection for dynamic network description. In Section 1.1, we first introduce the concept of community, and discuss the basic quantities of community structure. Then in Section 1.2, we introduce the description of community evolution in dynamic networks. Next, we describe existing algorithms designed for dynamic networks in Section 1.3. The evaluation of the obtained clusterings is an important task, therefore, Section 1.4 is devoted to the discussion of benchmarks for testing the reliability of algorithms. Section 1.5 ends this chapter with a discussion about future research directions in this issue.

1.1 Communities in networks

It appears natural and common to model the topology structure of a complex system by a graph (or network). Many real world problems (biological, social, web) can be effectively modeled as networks or graphs where nodes represent entities of interest and edges mimic the interactions or relationships among them. A graph $G = (V, E)$ consists of two sets V and E, where $V = \{v_1, v_2, \ldots, v_n\}$ are the nodes (or vertices, or points) of the graph G and $E \subseteq V \times V$ are its links (or edges, or lines). The number of elements in V and E are denoted by n and m, respectively.

In the context of graph theory, an adjacency (or connectivity) matrix $\mathbf{A}$ is often used to describe a graph G. Specifically, the adjacency matrix of a finite graph G on n vertices is the $n \times n$ matrix $\mathbf{A} = [A_{ij}]_{n \times n}$, where an entry A_{ij} of $\mathbf{A}$ is equal to 1 if the link $e_{ij} = (v_i, v_j) \in E$ exists, and zero otherwise.

In the study of networks, such as computer, information networks, social networks or biological networks, finding underlying community structure is common. Social networks often include community groups based on common location, interests, hobbies, *etc.* Metabolic networks have communities based on modular functions [105]. Citation

networks form communities by research topic. In each context, communities are groups of nodes in a network with more edges inside than edges linking the rest of the network.

In the following, we introduce the definition of community, which depends on the context. Social network analysts have devised many definitions of communities with various degrees of internal cohesion among nodes [61, 111]. Many other definitions have been introduced by computer scientists and physicists. We distinguish three main classes of definitions: local, global and based on vertex similarity. We review the notion of community structure and hierarchies of communities. We also discuss the definition of the modularity function, derived to measure the quality of a graph partition into communities.

1.1.1 Definitions of community

Local definitions

Communities are parts of the graph (group of nodes), within which the connections are dense and between which the connections are sparse. In some specific systems or applications, they can be considered as separate entities with their own autonomy, which do not depend on the whole graph. For instance in [80], communities are defined in a very strict sense and require that all pairs of nodes are connected. In other words, this corresponds to a clique, *i.e.*, a subset whose nodes are all adjacent to each other. However, such a criterion is too strict. A relaxable extended definition is *k-clique community*, which is the basis of CPM (Clique Percolation Method) [98]. A *k-clique community* is a series of adjacent cliques, where two k-cliques are *adjacent* if they share k-1 nodes.

Another criterion for community cohesion is the difference between the internal and external cohesion of the community. This idea is also used to define communities. For instance, Radicchi *et al.* [104] proposed the definitions of *strong communities* and *weak communities*. A set of nodes is a community in a strong sense if the internal degree of each node is greater than its external degree. This definition seems too strict. Its relaxable definition is the community in a weak sense: the internal degree of the community (sum of all its node internal degree) should exceed its external degree. Note that a community in a strong sense is also a weak community, while the converse is not generally true.

Global definitions

Communities can be defined with respect to the graph as a whole. This seems to be reasonable when the community structure is exactly the division of the graph into several groups of nodes. In such a context, many global criteria are used to identify communities, which are all based on the intrinsic idea that a graph offers a community structure if it is not a random graph. Random networks such as Erdös-Renyi's graphs do not display community structure. Indeed, as any pair of nodes are linked with the same probability, there should be no preferential wiring involving special groups of nodes. Therefore, one may define a *null model*, *i.e.*, a random graph that shares some structural properties of the original graph such as its degree distribution. The null model

is the basic element in the conception of the notion of *modularity*. The modularity is a quality function that evaluates the partition of a graph into disjoint communities. The most popular modularity is proposed by Newman and Girvan [91], which compares the number of edges inside the community to the expected number of internal edges in the null model. A series of algorithms using modularity maximization heuristics [16, 93] for finding communities are proposed and developed.

Definitions based on node similarity

It seems also natural to assume that communities are groups of nodes similar to each other. One can compute the similarity between each pair of nodes with respect to some reference properties. An important class of node similarity measures is based on properties of random walks on graphs, such as *commute-time*. The *commute-time* between a pair of nodes is the average number of steps needed for a random walker, starting at either node, to reach the other node for the first time and to come back to the starting node. Saerens *et al.* [109] have studied and used the commute-time as a similarity measure: the larger the commute-time is, the less similar nodes are.

1.1.2 Community structure

Basics

A *partition* is a division of a graph into disjoint communities, such that each node belongs to a unique community. A division of a graph into overlapping (or fuzzy) communities is called a *cover*. We use $\mathcal{P} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{n_c}\}$ to denote the partition, which is composed of n_c communities. In $\mathcal{P}$, the community to which the node v belongs to is denoted by σ_v. By definition we have $V = \cup_1^{n_c} C_i$ and $\forall i \neq j, C_i \cap C_j = \emptyset$. We denote by $\mathcal{S} = \{S_1, \ldots, S_{n_c}\}$ a cover composed of n_c communities. In $\mathcal{S}$, we may find a pair of community S_i and S_j such that $S_i \cap S_j \neq \emptyset$.

Given a community $\mathcal{C} \subseteq V$ of a graph $G = (V, E)$, we define the internal degree k_v^{int} (respectively the external degree k_v^{ext}) of a node $v \in \mathcal{C}$, as the number of edges connecting v to other nodes belonging to $\mathcal{C}$ (respectively to the rest of the graph). If $k_v^{\text{ext}} = 0$, the node v has only neighbors within $\mathcal{C}$: assigning v to the current community $\mathcal{C}$ is likely to be a good choice. If $k_v^{\text{int}} = 0$ instead, the node is disjoint from $\mathcal{C}$ and it should better be assigned to a different community. Classically, we note $k_v = k_v^{\text{int}} + k_v^{\text{ext}}$ the degree of node v. The internal degree k^{int} of $\mathcal{C}$ is the sum of the internal degrees of its nodes. Likewise, the external degree k^{ext} of $\mathcal{C}$ is the sum of the external degrees of its nodes. The total degree $k_{\mathcal{C}}$ is the sum of the degrees of the nodes of $\mathcal{C}$. By definition: $k_{\mathcal{C}} = k_{\mathcal{C}}^{\text{int}} + k_{\mathcal{C}}^{\text{ext}}$.

Modularity

One may want to measure the quality of a partition through a *quality function*, which assigns a score to each partition of a graph. In this way, partitions can be ranked based

on their score given by the quality function. Partitions with high scores are "*good*", so the one with the highest score is by definition the best.

The widest accepted quality function is the modularity introduced by Newman and Girvan [91, 95]. Let e_{ij} be the fraction of edges in the network that connect nodes in community i to those in community j, and $a_i = \sum_j e_{ij}$. The modularity measure is defined as:

$$Q = \sum_i \left(e_{ii} - a_i^2\right). \tag{1.1}$$

This quantity measures the fraction of the within-community edges in the network minus the expected value in a network with the same community division but when connections between nodes are random. If the number of within-community edges is less than the expected number of edges in a random graph, we will get $Q = 0$. Values approaching $Q = 1$, which is the maximum, indicate networks with strong community structure. In practice, values for real networks typically fall in the range from 0.3 to 0.7. Higher values are rare.

Suppose we have a division of a network into communities. Let σ_i be the community to which node i is assigned. The fraction of the edges in the graph that fall within communities, *i.e.*, that connect nodes that both lie in the same community, is

$$\frac{\sum_{ij} A_{ij}\delta(\sigma_i, \sigma_j)}{\sum_{ij} A_{ij}} = \frac{1}{2m}\sum_{ij} A_{ij}\delta(\sigma_i, \sigma_j)$$

where the function $\delta(\sigma_i, \sigma_j)$ is 1 if $\sigma_i = \sigma_j$ and 0 otherwise. At the same time, the expected number of edges between nodes i and j if edges are placed at random is $k_i k_j / 2m$, where k_i and k_j are the degrees of the nodes and m is the total number of edges in the network. Thus the modularity [90], as defined above, is given by:

$$Q = \frac{1}{2m}\sum_{i \neq j}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(\sigma_i, \sigma_j)\,. \tag{1.2}$$

Note that the modularity is always smaller than one but can be negative as well. For instance, the partition where each node represents a single community is always negative. When considering the whole graph as a single community, the modularity is zero as the two terms in this case are equal and cancels each other out. There are also other types of modularity, some of which are motivated by specific classes of clustering problems or graphs [43].

Modularity has been employed as quality function in many algorithms, like some division algorithms [93] which give a tradeoff between high accuracy and low complexity. In addition, modularity optimization is the most popular method for community detection. Heuristic proposed in [16] runs fast and handles very large-scale networks. Modularity also allows to assess the stability of partitions [84].

However, the applicability and reliability of modularity for the problem of graph clustering may be limited. An important issue concerning the limits of modularity is

raised by Fortunato and Barthelemy [44]. The study shows that a large value for the maximum modularity does not necessarily mean that a graph has a clear community structure. In a random graph, such as the Erdös-Rényi model, the distribution of edges among the nodes is highly homogeneous. For instance, the distribution of the number of neighbours of a node, or degree, is binomial, so most nodes have equal or similar degree. The random graph is supposed to have no community structure, as the link probability between nodes is either constant or a function of the node degrees, so there is no bias a priori towards special groups of nodes. Still, random graphs may have partitions with large modularity values [55, 106]. This is due to fluctuations in the distribution of edges in the graph, which determine concentrations of links in some subsets of the graph, which then appear as communities.

Moreover, Fortunato and Barthelemy [44] have found that modularity optimization has a resolution limit. It may prevent from detecting communities which are comparatively small with respect to the graph as a whole. Given two communities $\mathcal{A}$ and $\mathcal{B}$, with a total degree $k_{\mathcal{A}}$ and $k_{\mathcal{B}}$ respectively and where the number of edges connecting $\mathcal{A}$ and $\mathcal{B}$ is $l_{\mathcal{AB}}$. The difference of modularity determining the merge of two communities with respect to the whole graph partition is:

$$\Delta Q = \left[\frac{k_{\mathcal{A}}^{int} + k_{\mathcal{B}}^{int} + 2l_{\mathcal{AB}}}{2m} - \left(\frac{k_{\mathcal{A}} + k_{\mathcal{B}}}{2m}\right)^2\right] - \left[\frac{k_{\mathcal{A}}^{int} + k_{\mathcal{B}}^{int}}{2m} - \left(\frac{k_{\mathcal{A}}}{2m}\right)^2 - \left(\frac{k_{\mathcal{B}}}{2m}\right)^2\right].$$

If $l_{\mathcal{AB}} = 1$, *i.e.*, there is a single edge joining $\mathcal{A}$ to $\mathcal{B}$, we expect that the two communities should be separated. If $k_{\mathcal{A}}k_{\mathcal{B}}/2m^2 < \frac{1}{m}$, we have $\Delta Q_{\mathcal{AB}} > 0$. For simplicity, let us suppose that $k_{\mathcal{A}} \sim k_{\mathcal{B}} = k$, *i.e.*, that the two subgraphs have roughly the same number of edges. We conclude that when $k < \sqrt{2m}$ and the two communities $\mathcal{A}$ and $\mathcal{B}$ are connected, then the modularity is higher if they are in the same cluster [44]. So, if the partition with maximum modularity includes clusters with total degree of the order of $\mathcal{O}(\sqrt{m})$ (or smaller), one can not know a priori whether the clusters are composed of single communities or are in fact a combination of smaller weakly interconnected communities. This resolution problem may have important impacts in practical applications.

Hierarchy

An important aspect related to community structure is the hierarchical organization. A community structure can be hierarchically ordered, when the graph has several levels of organization/structure at different scales. In this case, the community structure is hierarchically composed of small communities at each level that are nested within large communities at higher levels. As an example, in a social network of children living in the same town, one could group the children according to schools they attend, but within each school one can make a subdivision into classes.

The hierarchical form of organization is often represented as a tree or dendrogram, as shown, for example, in Fig. 1.1. The hierarchy allows efficient analysis of several specific functions using modules, such as *majority consensus*. Majority consensus is widely used in the reconstruction of phylogenetic trees [25].

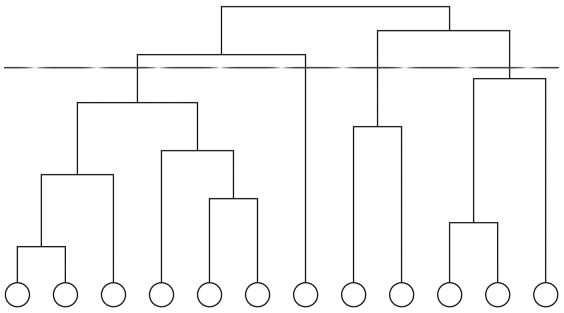


Figure 1.1: A hierarchical tree or dendrogram illustrating the hierarchical form of organization described here. The circles at the bottom of the figure represent the individual nodes of the network. As we move up the tree, the nodes join together to form larger and larger communities, as indicated by the lines, until we reach the top, where all are joined together in a single community. Alternatively, the dendrogram depicts an initially connected network splitting into smaller and smaller communities as we go from top to bottom. A cross section of the tree at any level, such the one indicated by a dotted line, will give the communities at that level. The vertical height of the split points in the tree are indicative only of the order in which the splits or joins take place, although it is possible to construct more elaborate dendrograms in which these heights contain other information. The figure is obtained from Ref. [95].

The presence of hierarchy motivates hierarchical clustering [95], which is a well-known technique in social network analysis [129], biology [34] and finance [83]. Starting from a partition in which each node is its own community, or all nodes are in the same community, one merges or splits clusters according to a topological measure of similarity between nodes. Though this method naturally produces a hierarchy of partitions, nothing is known a priori about their qualities. The modularity is a good quality function to identify a single partition, *i.e.*, the selected partition corresponds to the largest value of the modularity.

1.2 Community evolution in dynamic networks

In complex networks, the interactions between entities dynamically evolve over time [8]. Lets take Facebook[1] as an example: users add or delete "friends" [35]. Similarly, new forms of social contacts can be observed in phone calls, e-mail exchanges [78] or other communications on the Internet.

Traditional analysis treats networks as *static* graphs, which is either derived from an aggregation of data over the whole network life (experiment measure), or from a snapshot of data at a particular time step. Although this study provides meaningful results, the dynamic features are neglected. Dynamic features are also important in the study of complex networks.

[1] `http://www.facebook.com/`

During the last decade, the availability of large data set (thanks to Open Data initiative), the optimized rating of computing facilities, as well as the development of powerful and reliable data analysis tools, have constituted a better and better machinery to explore the topological properties of several networked systems from the real world. This has allowed to study the topology of the dynamic interactions in a large variety of Big Data [27] as diverse as communication [100, 99], social [32, 94] and biological systems [63, 18].

The goal of community detection in dynamic networks is to track community evolution and to identify their dynamics. In the following, we first describe the definitions and notations of a community which is observed at different time steps. Second, we present community dynamics which are used to describe community changes.

1.2.1 Communities in dynamic graphs

A *dynamic graph* $\mathcal{G}(V, \mathcal{E})$ on a finite time sequence $1 \dots \Delta$ is a sequence of graph snapshots $\{G(1), \dots, G(\Delta)\}$. There is a set $V = \{v_1, \dots, v_n\}$ of nodes. Each node $v_i \in V$ appears at least one during the dynamic graph lifetime, *i.e.*, $\exists t \; s.t. \; v_i \in G(t)$.

At each time step t where $1 \leq t \leq \Delta$, the corresponding snapshot $G(t)$ describes interactions between active nodes at time t, where the edges of a snapshot graph is a set of active dynamic links. $G(t)$ is partitioned into a set of *temporal clusters* $\mathcal{P}(t) = \{C_1(t), \dots, C_{n_c^t}(t)\}$, where n_c^t denotes the number of temporal clusters in $G(t)$. In some definitions of communities in dynamic networks [40, 41], the number of temporal clusters may be not equal to the number of communities at the same time step t . One community $\mathcal{C}_i$ at time step t is possibly represented by a set of temporal clusters such that $\mathcal{C}_i(t) = \{C_1(t), \dots\}$.

The problem of tracking community evolution can be resolved by the identification of a set of *community evolution paths* (or *community evolution traces* [128], *dynamic communities* [51]).

Definition 1 (Community evolution path)**.** *For a given time window* $[\delta, \delta+\Delta]$*, an evolution path* $\mathrm{Evol}(\mathcal{C}_i)$ *is a time-series of temporal clusters:* $\mathrm{Evol}(\mathcal{C}_i) := \{C_i(\delta), \dots, C_i(\delta + \Delta)\}$ *where each temporal cluster* $C_i(t) \in \mathrm{Evol}(\mathcal{C}_i), t \in [\delta, \delta + \Delta]$ *is the observation of the community* $\mathcal{C}_i$*.*

In the definition of Wang *et al.* [128], the observation of the community $\mathcal{C}_i$ at time t can be the union of several temporal clusters. When a community appears for the first time, it should be a unique temporal cluster.

1.2.2 Community dynamics

When we track community evolution, one problem is to characterize community dynamics. How does a community change over time? Palla *et al.* have introduced the main phenomena occurring during the lifetime of a community (See Fig. 1.2): creation, growth, reduction, fusion, split and death (or removal). Moreover, Chakrabarti *et al.* [19]

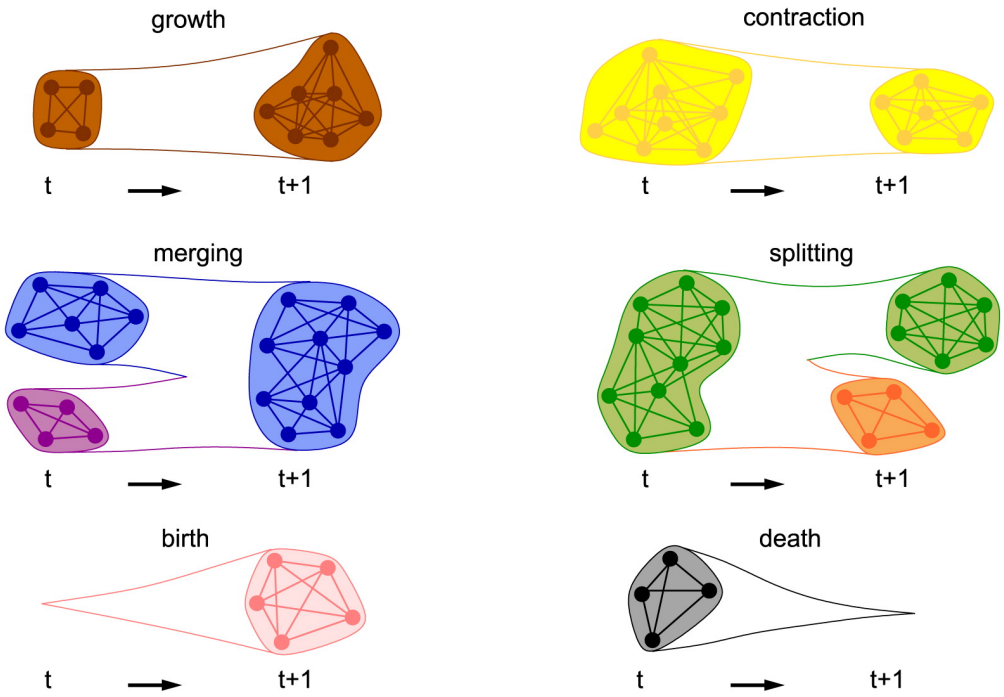


Figure 1.2: Possible scenarios in the evolution of communities. The figure is gained from [97].

proposed the definition of *change point* to describe a significant change in community structure. In the following, we describe them in details.

Community changes

We show six community changes in Fig. 1.2, which are used to describe the main events occurring in dynamic graphs. In order to identify them, Asur *et al.* [3] have proposed a definition.

Definition 2. *Let $G(t)$ and $G(t+1)$ be snapshots of $\mathcal{G}$ at two consecutive time steps with the cluster $C_i(t)$ and $C_i(t+1)$ denoting the observations of the community $\mathcal{C}_i$ at time step t and $t+1$, respectively.*

Continue: *$C_i(t+1)$ is the continuation of $C_i(t)$ if $C_i(t+1)$ is the same as $C_i(t)$:*

$$C_i(t) = C_i(t+1)$$

κ−Merge: *two clusters $C_i(t)$ and $C_j(t)$ merge into $C_i(t+1)$ if $C_i(t+1)$ contains at least κ% of nodes belonging to the union of $C_i(t)$ and $C_j(t)$ and the renewal of*

$C_i(t)$ and $C_j(t)$ is at least 50%:

$$\begin{aligned} \frac{|(C_i(t) \cup C_j(t)) \cap C_i(t+1)|}{\max(|C_i(t) \cap C_j(t)|, |C_i(t+1)|)} &> \kappa \\ |C_i(t) \cap C_i(t+1)| &> |C_i(t)|/2 \\ |C_j(t) \cap C_i(t+1)| &> |C_j(t)|/2 \end{aligned}$$

$\kappa-$Split: *$C_i(t)$ is split into $C_i(t+1)$ and $C_j(t+1)$ if κ% of nodes belonging to $C_i(t)$ are in two different clusters at time $t+1$, such as*

$$\begin{aligned} \frac{|(C_i(t+1) \cup C_j(t+1)) \cap C_i(t)|}{|\max(|C_i(t+1) \cap C_j(t+1)|, |C_i(t)|)} &> \kappa \\ |C_i(t) \cap C_i(t+1)| &> |C_i(t+1)|/2 \\ |C_i(t) \cap C_j(t+1)| &> |C_j(t+1)|/2 \end{aligned}$$

Emerge: *a new cluster $C_i(t+1)$ emerges at time $t+1$ if none of the nodes in the cluster $C_i(t+1)$ are grouped together at time t,* i.e., *$\nexists\, C_i(t)$, such that $|C_i(t) \cap C_i(t+1)| > 1$;*

Disappear: *$C_i(t)$ disappears if none of the nodes in the cluster $C_i(t)$ are grouped at time $t+1$,* i.e.,*$\nexists\, C_i(t+1)$, such that $|C_i(t) \cap C_i(t+1)| > 1$.*

This definition has several limits. First, the definition of one continuation is so strict that almost all communities do not have any continuation at the next time step. Second, the value of κ needs be set to determine when a community is merged or when a community is split. Varying κ may lead to different results. Finally, the definition of emerging community or disappearing community has weaknesses. Some clusters may be generated only by the fluctuation of degree distribution. This artificial clusters will not share a strong common interest. For the disappearance, the process may be too slow: a community may lose its core nodes but still have node attached to it. In this case, the observed community does not share a strong common interest anymore. It is difficult to determine whether a community exists.

There are also other types of definitions [22, 49, 51]. For example, Chen *et al.*. [22] characterize community dynamics by tracking community core evolution. Greene *et al.*. [51] use the definition of dynamic communities described above but require that if several dynamic communities share the same temporal cluster at time t, then these dynamic communities should merge.

In Fig. 1.3, we have shown examples of community evolution. There are four dynamic communities over the total three time steps, whose evolution paths are expressed as following:

$$\begin{aligned} \text{Evol}(\mathcal{C}_1) &\leftarrow \{C_1(t), C_1(t+1), C_1(t+2)\} \\ \text{Evol}(\mathcal{C}_2) &\leftarrow \{C_2(t+1), C_2(t+2)\} \\ \text{Evol}(\mathcal{C}_3) &\leftarrow \{C_3(t), C_3(t+1), C_3(t+2)\} \\ \text{Evol}(\mathcal{C}_4) &\leftarrow \{C_4(t+2)\} \end{aligned}$$

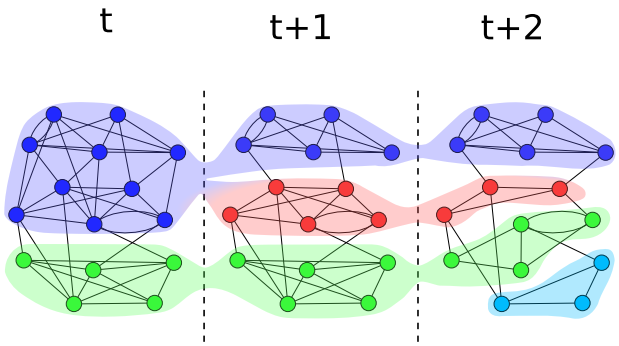


Figure 1.3: Examples of community evolution over three snapshot graphs by matching temporal clusters to dynamic communities. We observe 4 dynamic communities, indicated by colours: $\mathcal{C}_1$ in dark blue, $\mathcal{C}_2$ in red, $\mathcal{C}_3$ in green and $\mathcal{C}_4$ in light blue. During their evolution, we observe the community $\mathcal{C}_1$ is split into $\mathcal{C}_1$ and $\mathcal{C}_2$ between t and $t+1$.

Through these evolution paths, we observe two new communities appearing during network evolution: the community $\mathcal{C}_2$ is the branch of $\mathcal{C}_1$ and the community $\mathcal{C}_2$ emerges at time $t = 2$.

This is an example to illustrate the relationship between community dynamics and community evolution paths. We conclude that the problem of identifying and characterizing community dynamics can be revealed by community evolution paths, whereas the problem of tracking community evolution in dynamic networks can be reformulated as a problem of constructing community evolution paths across one or more time steps.

Change point

There is another definition about community dynamics. Chakrabarti *et al.* [19] have detected *change point*, which represents a significant time point when the system evolves, *i.e.*, a major change (or critical event) occurs in the graph structure during a short period. The approach called GraphScope[19] applied the MDL (Minimum Description Length) principle [53] to compute the encoding cost of assigning nodes into communities. A *segment* presents a sequence of graphs without any change in its community structure. So the graphs of each segment are characterized by the same partition with the lowest encoding cost. If the cost for encoding a graph into the existing segment is higher than the cost for encoding the graph into a new segment, a significant change of community structure occurs. The change point offers one important benefit of detecting community evolution using information theory.

1.3 Community detection in dynamic networks

In order to track community evolution, it is necessary to identify communities at different time steps. In [58], Hopcroft *et al.* have detected the partition of each snapshot graph

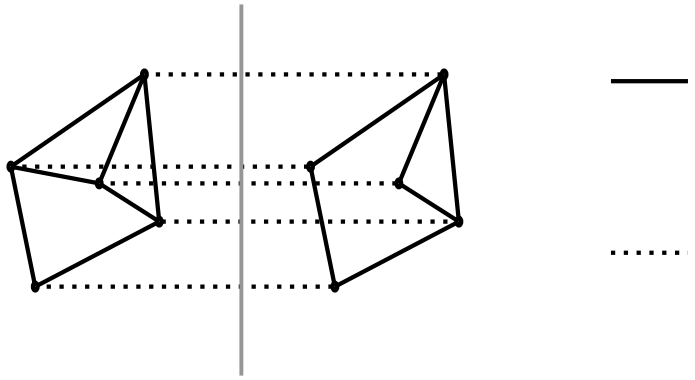


Figure 1.4: An example of a coupling graph, where graphs at different time steps are connected through couplings. The real interactions between nodes are shown in solid lines while the coupling interactions are denoted by dotted lines. The figure is gained from Ref. [62].

by hierarchical clustering [60], and then matched communities at different time steps through *natural communities*. *Natural communities* are groups of nodes having high stability against perturbations of interactions. In analysing citation networks, natural communities can be used to denote topics of communities. Tracking natural community evolution allows us to understand the history of topics, such as the emergence of new topics. The idea of detecting *time-independent* communities at different time steps and then matching them, becomes the basis for several algorithms. They are called *two-stage approaches*. Each *time-independent* community is detected independent of the results at other time steps.

Another method [119, 123, 124] called *evolutionary clustering* is proposed to detect *time-dependent* temporal clusters. The principle of *evolutionary clustering* [19] is to simultaneously optimize two potentially conflicting criteria: first, the clustering at any time step should remain faithful to the current data as much as possible; and second, the clustering should not shift dramatically from one time step to the next.

There are also many other methods, such as *coupling graph clustering*. The *coupling graph clustering* is a framework which detects community structure of a *coupling graph*. A *coupling graph* is a graph linking a sequence of graphs over several time steps by adding coupling edges between the same nodes at different time steps (See Fig. 1.4). Given a coupling graph, a subgraph which describes all interactions at a specific time step is call a *slice*.

In the following, we begin by listing the challenges raised by community detection in dynamic communities. Next, we review current techniques proposed for community detection in dynamic networks.

1.3.1 Dynamic community detection challenges

Quality function: the cost

Reliable algorithms are supposed to provide results having a high quality value. In the case of community detection in dynamic networks, a famous function named α-cost has been used by several algorithms [68, 118, 124] for measuring the quality of the found dynamic communities. This α-cost is motivated by the principle of evolution clustering: the community structure at each time step is the evolution of the community structure at the previous time step. Therefore, it is a combination of a *snapshot cost* and a *past history cost*. The parameter α controls the relative weight of recent and past history:

$$\text{cost} = \alpha\, \mathcal{CS} + (1 - \alpha)\, \mathcal{CT} \tag{1.3}$$

where the snapshot cost $\mathcal{CS}$ measures how a community structure fits the graph interactions at time t and the past history cost $\mathcal{CT}$ qualifies how consistent the community structure is with the past history community structure at time $t-1$.

Let X represent the current community structure, Y represent the community structure at the previous time step, W denote current graph interaction, Λ be an non-negative diagonal matrix, and $D(\bullet)$ be the function for measuring the cost such that $D(\bullet)$ computes the similarity between the network structure and the community structure and the similarity between the current community structure and the previous community structure.

In [124], authors defined $D(\bullet)$ as a KL-divergence between two objects such that: $\mathcal{CS} = D(W \parallel X\Lambda X^T)$ and $\mathcal{CT} = D(Y \parallel X\Lambda)$. Given two objects A and B, $D(A \parallel B) = \sum_{ij} \left(a_{ij} \log \frac{a_{ij}}{b_{ij}} - a_{ij} + b_{ij} \right)$. Through this cost definition, the snapshot cost is high when the approximate community structure fails to fit the graph interactions at time t while the past history cost is high when there is a dramatic change of community structure from time $t-1$ to t.

There exist also other definitions of $D(\bullet)$. In [68], two definitions of the cost are introduced: one is the distance between all pairs of objects in an agglomerative hierarchical clustering, and the other is associated with the centroid of the community in k-means clustering [15]. In k-means clustering, community memberships are measured by the membership degrees of nodes, *i.e.*, the distance between the node to the centroid of its community. Then, in the cost of the community structure of a dynamic graph, the snapshot cost is associated with the distance between the node and the centroid of its community, and the past history cost is computed by the difference between the current community centroid and the community centroid at the previous time step.

In the case of multi-mode networks, Tang *et al.* [119] have suggested the resolution by transforming the problem in multi-mode networks into the problem of two-mode. Most of existing work concentrates on *one-mode network*. That is, there is only one type of social actors (nodes) involved in the network and the ties (interactions) between actors are all of the same type. This is common in a broad sense such as friendship network, Internet, phone call network, *etc.* . However, some applications such as web

mining, collaborative filtering, and online targeted marketing involve more than one type of actors and multiple heterogeneous interactions between different types of actors. Such a network is called *multi-mode network* [129].

Given an m-mode network, for each mode i, let $\mathbb{X}_i$ denote this mode of nodes, such as $\mathbb{X}_i = \{x_1^i, \ldots, x_{n_i}^i\}$, where n_i is the number of nodes for $\mathbb{X}_i$. Then, for each pair of modes, we use $\mathbf{R}_{ij}^t \subseteq \mathbb{X}_i \times \mathbb{X}_j$ to represent interactions between two modes of nodes $\mathbb{X}_i, \mathbb{X}_j$ at time t. Ideally, the interaction between nodes can be approximated by:

$$\mathbf{R}_{ij}^t \approx \mathbf{C}_i^t \mathbf{A}_{ij}^t (\mathbf{C}_j^t)^T$$

where $\mathbf{C}_i^t$ is the cluster membership for $\mathbb{X}_i$ at time t and A_{ij}^t represents the group interaction. The group interaction is computed by $A_{ij}^t = (\mathbf{C}_i^t)^T \mathbf{R}_{ij}^t \mathbf{C}_j^t$. Therefore, for each temporal m-mode graph at time t, its snapshot cost $\mathcal{CS}$ can be formulated as:

$$\sum_{1 \leqslant i < j \leqslant m} w_a^{(i,j)} D\left(\mathbf{R}_{ij}^t \parallel \mathbf{C}_i^t \mathbf{A}_{ij}^t (\mathbf{C}_j^t)^T\right) ,$$

and its history cost $\mathcal{CT}$ is expressed as:

$$\sum_{1 \leqslant i \leqslant m} w_b^i D\left(\mathbf{C}_i^t \parallel \mathbf{C}_i^{t-1}\right) ,$$

where w_a^{ij} is an importance factor for every pair of modes i and j, and w_b^i is a relative importance factor for each mode i.

The optimal value of the cost corresponds to a good community structure which incorporates the deviation from the past history. There exist several algorithms detecting community evolution by optimizing α-cost, such as community model (See Section 1.3.3). However, the value of the parameter α is *a priori* unknown, which is a major limitation. Since the parameter α controls the relative weight of recent and past history, the obtained results [68] depend on the value of α: a lower value of α yields to a less change of community structure. If $\alpha = 0$, the obtained community structure is exactly the same as applying the community detection algorithm independently on each snapshot. A good quality function for dynamic graphs should find the perfect compromise and accommodate past history without compromising the snapshot quality.

Matching metric

A matching metric is a similarity function, which measures how similar two communities are. It is often used in two-stage approaches to connect similar communities. Of course, we can measure the similarity between two temporal clusters at different time steps. Then, we obtain how one community evolves from one time step to the following time steps.

Hopcroft *et al.* [58] defined a *match* function. Let C and C' be two clusters, their match value is written as follows:

$$\text{match}(C, C') = \min\left(\frac{|C \cap C'|}{|C|}, \frac{|C \cap C'|}{|C'|}\right) \tag{1.4}$$

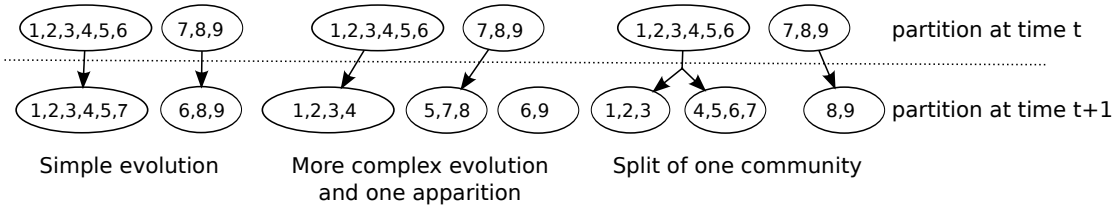


Figure 1.5: Examples of community evolution in a time period $[t, t+1]$. We match clusters at time t to the clusters at time $t+1$. Given a cluster at time t, it remains stable if it is matched to a unique cluster at time $t+1$; it splits if it is matched to more than one cluster at time $t+1$. In addition, one new cluster appears at time $t+1$, if no cluster at time t is matched to it. The figure is obtained from [5].

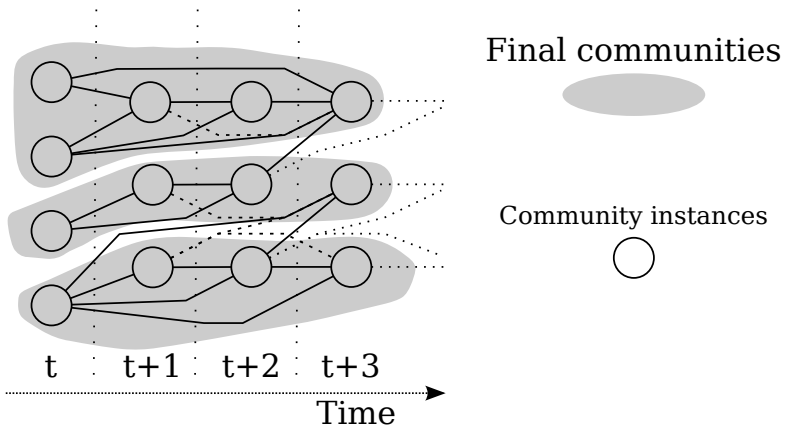


Figure 1.6: An example of dynamic networks with community instances (nodes) and final communities (in grey). At each time step, we may match several community instances to the same temporal community.

The definition ensures that a high matching value (close to 1) occurs when two clusters have many common nodes and are roughly of the same size. The best match value for C at time t, is the highest match(C, C') value for any cluster C' at time t.

Palla *et al.* [97] defined *relative overlap*, which is a Jaccard index. The relative overlap value between two communities X and Y is written as follows:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \,. \tag{1.5}$$

By definition, the cluster $C(t+1)$ at time $t+1$ is matched to the cluster $C(t)$ which has the largest overlap at time t.

Another bipartite mapping metric is *dynamic Jaccard's index*, whose definition is:

$$JacD'(X,Y) = \frac{J(X,Y)}{|t-t'|} \tag{1.6}$$

where $|t - t'|$ represents the time interval duration between communities X and Y. It allows a temporal cluster matched to an old one ($|t - t'| > 1$) which disappeared during several time steps.

Two communities are matched if they share the highest matching value. The matching metric is a natural resolution to connect temporal clusters over time. So it is often used in two-stage methods [58, 97, 121]. Its another advantage is to characterize community dynamics (See Section 1.2.2). However, there is no standard definition of matching metric. In Hopcroft *et al.* [58]'s *match* function (Eq. 1.4), the minimum size of communities is important for the comparison. Instead, the size of the union of communities is essential in the relative overlap (Eq. 1.5). Furthermore, a minimum intersection size threshold needs to set, *i.e.*, the minimum number of common nodes shared by the matching communities.

1.3.2 Two-stage approaches

The basic idea of two-stage approaches is to detect temporal clusters at each time step, and then establish relationships between clusters for tracking community evolution over time. Figure 1.3 illustrates the result of applying a two-stage approach to a dynamic network across three time steps. In a first phase, clusters at each time step are detected: at time t, there are two clusters, then there are three clusters at time $t + 1$ and four clusters at time $t + 2$. In a second phase, the relationship between clusters at different time steps are established, which is shown by colours. Through the above results, we learn how the community structure of this graph evolves from the time step t to the time step $t + 2$. For the first phase, we apply a graph clustering algorithm [48]. For the second phase, we can use a matching metric (See Section 1.3.1). However, it may lead to noisy results where some nodes often change their community memberships. Therefore, many advanced resolutions are proposed to resolve this matching problem.

Core-based methods

If a partition is significant, it will be recovered even if the structure of the graph is modified, as long as the modification is not too extensive. Instead, if a partition is not significant, we may observe that minimal perturbations of the graph will suffice to disrupt its group memberships. A *significant cluster*, *i.e.*, a significant group of nodes, is often defined as a *community core*. We can reduce noisy results by matching community cores. This is the main principle of core-based methods. The matching metric (See Section 1.3.1) is often applied. Two temporal clusters are matched if their community cores share the highest similarity value.

Hopcroft *et al.* have proposed the concept of natural communities, which are significant clusters that have high stability against modification of graph structure. Given a temporal graph, by applying 5% of perturbations, a set of modified graphs are produced, each of which has 95% of core nodes. Each natural community is identified by the partitions corresponding to these modified graphs, which has the best match value with clusters in those partitions.

Rosvall *et al.* [108] used a bootstrap method [33] to detect significance of clusters. The bootstrap method assesses the accuracy of an estimate by resampling from the empirical distribution of observations. Each graph can be resampled by assigning to each edge a weight taken from a Poisson distribution with mean equal to the original edge weight. A graph clustering method is applied to the original graph and the samples. For each community in the original graph's partition, they define its largest subset of nodes that are classified in the same community in at least 95% of all bootstrap samples, as the significant cluster.

In some methods, core nodes are identified through their roles within their communities. Given a community, there are core nodes and peripheral nodes. Guimerá and Amaral [54] have classified community members into different roles according to intra- and inter-module connection patterns. With respect to core node identification, Wang *et al.* [128] defined core nodes, where each core node v satisfies $\sum_{u \in \text{neighbours}} (k_v - k_u) > 0$. In [12], k-cores nodes [1] are detected with a threshold k where k-core decomposition is used for filtering out peripheral nodes.

Although core-based approach can smooth variances caused by peripheral nodes, its results still suffer from some limits such as the parameters used in matching metrics. In additional, if we only track evolution of community cores, there is a risk of missing important structural changes which are related to peripheral nodes.

Union-graph-based methods

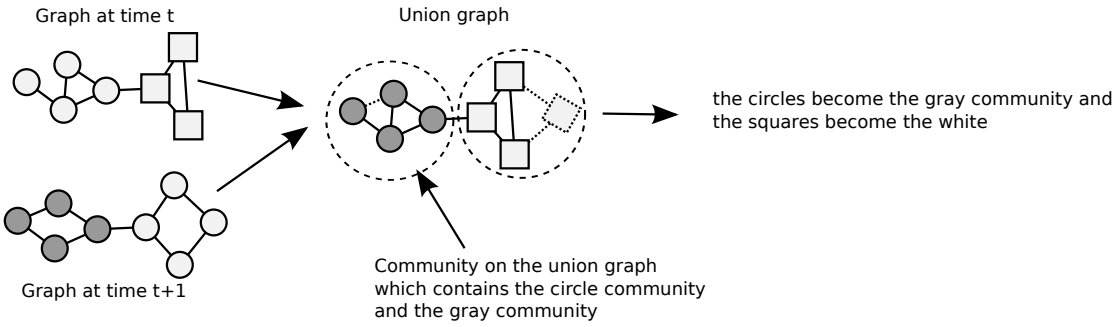


Figure 1.7: An example of an union graph which is constructed by jointing two graphs at time t and $t+1$. The figure is obtained from Ref. [97].

Another important early work [97] for detecting community evolution is related to the *union graph*. Each union graph merges two graphs (union of their links) present at contiguous time steps. Let $G(t, t+1)$ denote the union graph resulting from the union of two graphs at time t and $t+1$. We have $E_{t,t+1} = E_t \cup E_{t+1}$. Figure 1.7 gives an example of an union graph. Any community present at t or $t+1$ is contained in exactly one community in the union graph. Thus, communities in the union graph provide a natural connection between communities at t and $t+1$. If a community in

the joined graph contains a single community from t and a single community from $t+1$, then they are matched. If the joined group contains more than one community from both time steps, the communities are matched in decreasing order of their relative node overlap (Eq. 1.5). The technique is validated by applying it to two social systems: a graph of phone calls between customers of a mobile phone company over one year and a collaboration network between scientists spanning a period of 142 months.

The union graph smooths the change between every pair of consecutive time steps. This property can reduce the fluctuation caused by noisy data. In addition, the union graph allows us to directly determine the links between temporal clusters at consecutive time steps. It simplifies the problem of tracking community evolution.

The main disadvantage of this technique is that the CPM algorithm used only detects communities in certain contexts, *i.e.*, CPM algorithm fails to detect community structure of networks with few cliques. In addition, some parameters are used to determine how community change due to the application of similarity metric.

Survival-graph-based methods

Given a dynamic graph, its *community survival graph* is constructed by representing community instances as nodes which are linked via edges based on their similarity. One can divide this community survival graph into final communities. Each final community groups a set of temporal clusters and spans several time steps as shown in Fig. 1.6.

Algorithm 1 Hierarchical edge betweenness clustering

Input: $G = (V, E)$
Output: A dendrogram
  repeat
    Compute edge betweenness for all edges
    Remove edge with highest betweenness
  until no more edges in graph
  Return a dendrogram // *The dendrogram is produced from a top down approach: the network is split into different communities with successive removals of links. The leaves of the dendrogram are individual nodes.*

The first approach associated with survival graph is proposed by Falkowski *et al.* [40, 41]: first cluster each temporal graph to find community instances at each time step, then construct a community survival graph, and finally cluster the community survival graph to find final communities by using a hierarchical edge betweenness clustering [48].

To construct a community survival graph, a time window is set to compare the similarity between community instances and connect the similar community instances with edges. In another words, this time window size is the largest time distance between every pair of connected community instances in a community survival graph. The applied hierarchical edge betweenness clustering (see Algorithm 1) contains an iteration, which eliminates edges to separate subgraphs. In Falkowski *et al.*'s method, a parameter k is applied to determine the number of iterations. The connected subgraphs retained after

k iterations correspond to the final communities. A connected subgraphs consists of similar community instances.

Chi *et al.* [23] have detected final communities through a soft clustering [131], after detecting community instances [93, 113, 131] at each time step. At a time step i, the graph interaction is denoted by $\mathbf{A}^i \subseteq V \times V$ with l_i *basis subgraphs* $\mathcal{B}^i = [\mathbf{B}^i_1, \ldots, \mathbf{B}^i_{l_i}]$. Each *basis subgraph* describes interactions between nodes within a community instance. Across a time window $[1, \ldots, \Delta]$, graph interactions can be denoted by a 3-dimensional tensor: $\mathbb{A} = [\mathbf{A}^1, \ldots, \mathbf{A}^\Delta] \in \mathcal{R}^{n \times n \times \Delta}$. For the total $N_c = \sum_{i=1}^{\Delta} l_i$ *basis subgraphs*, another 3-dimensional tensor is defined: $\mathbb{B} = [\mathbf{B}^1_1, \ldots, \mathbf{B}^\Delta_{l_\Delta}] \in \mathcal{R}^{n \times n \times N_c}$.

Then, the final communities are obtained by minimizing the objective function: $D(\mathbb{A} \parallel \mathbb{B}\mathbf{U}\mathbf{V}^T)$. The matrices $\mathbf{U} = [u_{kj}]_{N_c \times n_c}$ and $\mathbf{V} = [v_{ij}]_{\Delta \times n_c}$ are the solution of the optimization problem. For each dynamic community j, u_{kj} is a vector of weight on k-th basis subgraph. At each time step i, v_{ij} is a community intensity for j-th final community.

In this method, the size of time windows and basis subgraphs are issues. A good size value of time windows allows us to group small community instances into a final community, if these small community instances have high frequency grouped together. The size of basis subgraphs is related to insignificant subgraphs (for example, a subgraph with only a couple of nodes), as insignificant subgraphs are removed for the computation. The larger size threshold of basis subgraphs is, the less iterations are used for computing $\mathbf{U}$ as less number of N_c. Therefore, the computation time can be optimized by increasing the size threshold of basis subgraphs.

For the number of communities n_c, they try different values to compare the reconstruction error and then choose one that is reasonably small and at the same time explains data reasonably well.

In [121], authors use a similar approach which tracks community evolution by connecting community instances but they use another notion of final community. A quality function called *node cost* is defined to determine the community membership for each node over time. This function is the sum of two costs: the cost of one node to keep its community membership and the cost of one node to change its community membership. Therefore, final community detection is transformed into the problem of optimizing this function. Optimizing this function is shown to be a NP-complet problem. Another solution withan approximate factor is proposed in [120]. In their proposed node cost function, the importance of different costs is predefined. Giving a high importance to cost of a node to keep its community membership, makes node membership stable for a long time duration. Giving a high importance to the cost of a node to change its community member, makes node membership to fit to current snapshot structure.

Survival-graph-based method gives results about how dynamic communities evolve over time directly. It simplifies the problem of tracking community evolution. Compared to other two-stage approaches, which track community evolution by identifying observations at each time step, this technique is more practical. However, some issues arise: How to choose the *time window size* ? How to choose *the number of clustering*

iterations in [41]? How to choose *the size threshold of basis subgraphs* and *the number of final communities* in [23]? And how to choose *the importance value* in [121].

Conclusion

Methods presented above are two-stage like approaches:

1. Clusters are detected at each time step independently of the results at any other time step;

2. Relationships between clusters at different time steps are inferred successively.

Such natural process often produces significant variations between partitions that are close in time, especially when the datasets are noisy. Since the first phase is independent of the past history, smooth transitions are impossible. Such an approach may produce artifacts if the data are noisy and variations between partitions may also be generated by the community detection algorithm it-self. Such artifacts yield to artificial community dynamics rather than the real graph evolution. For each graph, let $\mathcal{O}(P)$ denote the partition detection time and $\mathcal{O}(M)$ represent the computation time for the matching problem. The total time complexity of a two-stage approach on a time window of length $\mathcal{T}$ is in $\mathcal{O}((P+M)\,\mathcal{T})$.

1.3.3 Evolutionary clustering

An evolutionary clustering approach follows a principle of detecting community structure based on the current graph topology information at a given time t and on the community structure at previous time steps. The quality function used for dynamic community structure is: α-cost (See Eq. 1.3). By assuming that a good community structure has a high α-cost value, many optimization methods are proposed and are applied to real dynamic networks. For instance, Lin *et al.* [123, 124] used a probabilistic model to capture community evolution by maximizing α-cost. On one hand, proposed frameworks called *community model* usually search the optimal community structure for modeling the sequence of graphs by encompassing interactions of the whole graphs. On the other hand, *incremental/online algorithms* only consider interaction changes such as link insertion or link deletion which also make sense in detecting structural changes. In the following, we will review these evolutionary clustering methods.

Community model

Community evolution can be modelled by a sequence of graphs based on a probabilistic model, which assumes that:

1. The interactions of the graph at each time step follow a certain distribution;

2. The community structure follows a certain distribution that is determined by the community structure at the previous time step.

The first attempt has been done by Lin *et al.* [123, 124] through α-cost function optimization. Let $\mathbf{W}^t$ denote a graph structure at time t and $\mathbf{X}^t\mathbf{\Lambda}^t$ represent its community structure. By defining $\mathbf{Z}^t = \mathbf{X}^t\mathbf{\Lambda}^t(\mathbf{X}^t)^T$, the authors have devised an α-cost (Eq. 1.3):

$$\text{cost} = \alpha\, D(W^t \parallel \mathbf{Z}^t) + (1-\alpha)\, D(\mathbf{Z}^{t-1} \parallel \mathbf{Z}^t)\ .$$

Consequently, they estimate $\mathbf{X}^t$ and $\mathbf{\Lambda}^t$ for optimizing the cost. The problem of community detection at each time step becomes a problem in terms of maximum a posteriori (MAP) estimation. An EM algorithm for solving the MAP problem is given in [123, 124] with a low complexity where the graph structure is sparse.

This technique enables to detect overlapping community structure and track community evolution directly. So it is a good resolution for the problem of community detection in dynamic graphs. However, a priori the value of α is a drawback.

Yang *et al.* [132] also used a dynamic stochastic block model (DSBM) for finding communities and their evolutions in a dynamic social network. In their study, they have applied a Bayesian treatment for parameter estimation that computes the posterior distributions for all the unknown parameters.

Let $\mathbf{W}^t \in \mathbb{R}^{n\times n}$ denote a graph structure at time t and $\mathbf{Z}^t \in \mathbb{R}^{n\times n_c}$ is its community structure. For each node i, it is assigned into community k with a probability π_k, such as $\Pi = [\pi_1, \ldots, \pi_{n_c}] \in \mathbb{R}^{n_c}$. For a pair of nodes i and j whose community memberships are k and l respectively, the link connecting them is assumed to follow a Bernoulli distribution with parameter P_{kl}, such as $w_{ij}^t \sim \text{Beronulli}(\bullet \mid \text{P}_{\text{kl}})$, *i.e.*, $\mathbf{W}^t \sim \Pr(\mathbf{W}^t \mid \mathbf{P}, \mathbf{Z}^t)$, where $\mathbf{P} = [P_{kl}]_{n_c\times n_c}$. For a community matrix $\mathbf{Z}^{t-1}$, a transition matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$ is assumed to model $\mathbf{Z}^t$, such as $\mathbf{Z}^t \sim \Pr(\mathbf{Z}^t \mid \mathbf{Z}^{t-1}, \mathbf{B})$. So we write the likelihood for the DSBM model as follows:

$$\Pr(\mathbf{W}^t, \mathbf{Z}^t \mid \Pi, \mathbf{P}, \mathbf{B})\ .$$

With the Bayesian Model, a posterior probability $\Pr(\mathbf{Z}^t \mid \mathbf{W}^t)$ is computed with an inference algorithm.

There is no parameter in this technique. However, the authors only provide performances of the applications to networks with nearly ten time steps and a few hundred nodes. For large networks such as millions nodes and hundreds of time steps, the performance of this technique is not clear.

The community model captures community evolution by modeling the sequence of graphs. It performs well when applied to stable evolving graphs. However, it suffers from scalability problems due to an expensive matrix computation and storage cost.

Incremental/Online algorithms

The incremental spectral clustering [96] is one of the early incremental algorithms that update matrices like the degree matrix or the Laplacian matrix according to changes of graph interactions [81]. In traditional spectral clustering, community detection is transformed into the eigenvalue problem of $L\mathbf{q} = \lambda D\mathbf{q}$, where L is the Laplacian matrix, $\mathbf{q}$ is

the cluster indicator, λ is the eigenvalue and D is the degree matrix. Using incremental computation yields to a lower computational cost than the standard spectral clustering. Incremental computation only takes into account changes, thus the computation matrix is sparse. In addition, a tunable threshold τ is used to balance the computational cost and the accuracy. One drawback is that errors are accumulated after several steps and when the dataset grows or changes frequently the associated cost becomes expensive.

Modularity optimization is the most popular method for community detection. It is extended to detect community evolution, *e.g.*, the modularity-driven clustering proposed by Gorke *et al.* [50]. Their basic idea is to detect community structure by starting from a pre-clustering obtained from a standard modularity optimization heuristic. Then, they proposed and discussed heuristics based on global greedy algorithms or on local greedy algorithms. They pass a pre-clustering to the global version to adapt it to the dynamic case (dGlobal). Similarly, the local version remembers its old results: roughly speaking, the dynamic local version (dLocal) starts by letting all free (elementary) nodes reconsider their cluster. Then it lets all those (super-)nodes on higher levels reconsider their cluster, whose content has changed due to lower level revisions. Similarly, Dinh *et al.* [29] proposed another method extended from community optimization.

The community detection based on node similarity such as DBSCAN [37] is also extended for detecting dynamic community evolution [36]. DBSCAN considers a community as a core node and a *neighbourhood*. For each core node, its community must consist of at least η nodes within a radius distance ε. In IncrementalDBSCAN [36], each community updates its neighbourhood if its community members have changed their neighbours. Similarly, DENGRAPH [39] detected community evolution according to the core nodes and their neighbourhoods. Instead of a distance radius ε, a different distance function is proposed to compute core nodes and their neighbourhoods.

Incremental or online method can detect dynamic communities and save time by avoiding computations on sub-graphs where there is no change. However, all above approaches need predefined parameters.

Conclusion

There exist many other evolutionary clustering approaches. As mentioned in Section 1.2.2, information theory has also been used to detect community evolution in dynamic graphs. Sun *et al.* applied the MDL to find the minimum encoding cost to describe a time sequence of graphs and their partitions into communities. The basic principle of this method is to encode the graph topology into a compression information with the minimum cost of the description. This method enables to provide meaningful information on community evolutions. However, one drawback is the problem called *relevant variable*, which is the variance between real data and data compression. To what extent is information theory able to capture community structures? To our knowledge, we are still far from a precise definition of community while modularity (defined by Eq.1.2) is the widest accept quality function.

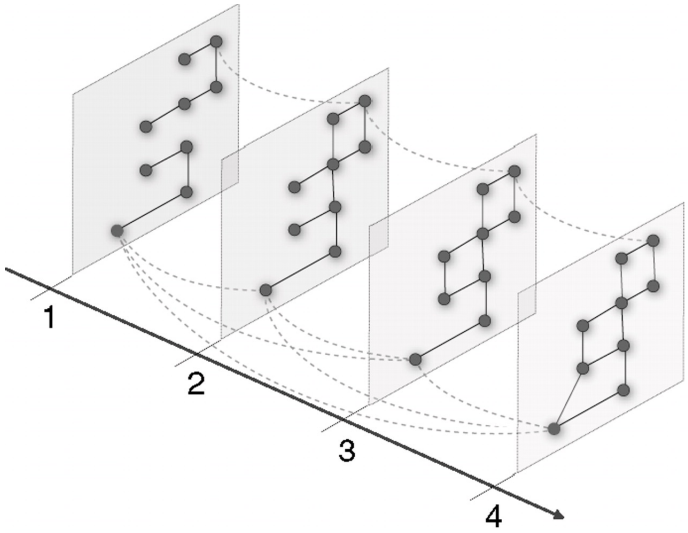


Figure 1.8: Schematic of a multislice (couplings) network. Four slices $s = \{1, 2, 3, 4\}$ represented by adjacencies A_{ijs} encode intra-slice connections (solid). Inter-slice connections (dashed) are encoded by C_{jrs} , specifying coupling of node j to itself between slices r and s. For clarity, inter-slice couplings are shown for only two nodes and depict two different types of couplings: (1) coupling between neighboring slices, appropriate for ordered slices; and (2) all-to-all inter-slice coupling, appropriate for categorical slices. The figure is gained from Ref. [88].

As opposed to two-stage approaches, evolutionary clustering does not encounter the matching problem. However, most methods are using parameters. Furthermore, we stress that evolutionary clustering results are generally too strongly correlated with community history which may occult structural changes.

1.3.4 Coupling graph clustering

Coupling graph clustering approach is based on a coupling graph as shown in Fig. 1.4. The underlying idea is once the coupling graph built (encompassing the time dimension as edges) to use an efficient standard static community detection heuristic. The first attempt is [62] where authors built a temporal graph and then used the classical community detection algorithm Walktrap [102]. The community evolution can be traced through group memberships over time.

Another method is proposed by Mucha *et al.* [88]. They detected dynamic communities by optimizing a modified modularity, which is motivated by α-cost (Eq. 1.3). The modified modularity balances the contribution of community memberships to each slice and the cost for changing community memberships. The major advantage of this algorithm is to smooth community evolution. However, its results rely on the parameter α and the relative weight of coupling.

This idea of coupling graph clustering simplifies the problem of detecting community evolution. However, it introduces the problem about how to construct coupling graphs: how to add *the weight on coupling edges*? what is *the length of coupling windows* (*i.e.*, the longest time interval between nodes connected by coupling edges)? For the length of coupling windows, we illustrate examples in Fig. 1.8. This figure is taken from [88], where each snapshot graph is called a *slice*. Two different lengths of coupling windows are given: *a*) couplings between neighboring slices such that the length is two time steps; and *b*) all-to-all inter-slice couplings such that the length is the total time steps.

1.4 Benchmarks

When designing a new algorithm, it is necessary to stress it through series of simple benchmark graphs, artificial or from the real world, for which the community structure is known. If the algorithm provides results agreeing with the ground truth, we may consider that the algorithm is reliable and can be used in applications. In this section, we firstly describe current benchmarks for testing dynamic community detection algorithms, and secondly review measures for comparing the similarity between computed modular structure and a ground truth.

1.4.1 Benchmark graphs

Computer-generated graphs

Computer-generated graphs try to build random graphs that have natural partitions. The simplest model of this form is for the graph bisection problem. This is the problem of partitioning the vertices of a graph into two equal-sized sets while minimizing the number of edges bridging the sets. To create an instance of the planted bisection problem, we first choose a partition of the vertices into equal-sized sets V_1 and V_2. When then choose probabilities $p_{\text{in}} > p_{\text{out}}$, and place edges between vertices with the following probabilities: The expected number of edges crossing between V_1 and V_2 will be $p_{\text{out}}|V_1|\,|V_2|$. If p_{in} is sufficiently larger than p_{out}, then every other bisection will have more crossing edges. There have been many analyses of the generalization of planted partition models to more than 2 partitions [26, 85]. The number of sub-graphs is equal to the number of predefined communities, and nodes within the same community are connected with a probability of p_{in} and connect to the rest with a probability of p_{out}. In addition, each subgraph is modeled by an Erdös-Rényi's model, which assigns equal probability to all graph edges. The model is motivated by the idea that vertices (or general items) belong to certain categories, and that vertices in the same categories are more likely to be connected. Such models also arise in the analysis of clustering algorithms. However, it is not clear that these models represent practice very well.

Lin *et al.* [123] have proposed a computer-generated benchmarks for testing their evolutionary clustering framework called FacetNet (See Section 1.3.3). They use the model of Newman [95] similar to the previous model as a basis (4 clusters of 32 nodes). They generate different graphs for each time steps. In each time step, dynamic is

introduced as the following: from each community, they randomly select 3 members to leave their original community and to join randomly the other three communities. Edges are added randomly with a higher probability p_{in} for within-community edges and a lower probability p_{out} for between-community edges. The average degree for nodes is set to 16.

Another similar benchmark is proposed in [31]. To introduce change points (See Section 1.2.2), sequence of graphs are separated into *segments*. Each *segment* is a sequence of graphs sharing the same community structure. The average degree of nodes and the internal and external connection probability are fixed. The edge weights are integers randomly chosen from 1 to 10 for intra-community edges and from 1 to 6 for inter-community edges.

All benchmarks for dynamic community detection extended from the planted partition model, used by Newman *et al.* have two main drawbacks: *a*) all nodes have the same expected degree; *b*) all communities have equal size. These features are unrealistic, as complex networks are known to be characterized by heterogeneous distributions of degree and community sizes.

Greene and Doyle [51] proposed a set of benchmarks based on Lancichinetti and Fortunato's technique [72]. Lancichinetti and Fortunato assumed that the distributions of degree and community size are power laws, with exponents τ_1 and τ_2, respectively. Each node shares a fraction $1 - \mu$ of its edges with the other nodes of its community and a fraction μ with the rest of the graph; μ is a mixing parameter in range of $[0, 1]$. Greene and Doyle contracted four different synthetic networks for four different event types, covering $15,000$ nodes over 5 time steps. In each of the four synthetic datasets, 20% of node memberships were randomly permuted at each step to simulate the natural movement of users between communities over time. Subsequently, community dynamic events were added as follows:

Intermittent communities at each time step, 10% of communities are unobserved from time $t = 2$ onwards.

Expansion and Contraction at each time step, 40 randomly selected communities expand or contract by 25% of their previous size.

Birth and death at each time step, 40 additional communities are created by removing nodes from other existing communities, and randomly remove 40 existing communities.

Merging and splitting at each time step, 40 temporal clusters of communities split, together with 40 cases where two existing communities were merged.

Chen *et al.* [22] constructed benchmark graphs using GTgraph [7] based on a recursive matrix graph model (R-MAT) [20]. The R-MAT model follows the preferential attachment idea (growing model where new nodes prefer to connect to existing nodes with higher degrees). In order to build a graph, the R-MAT recursively subdivides

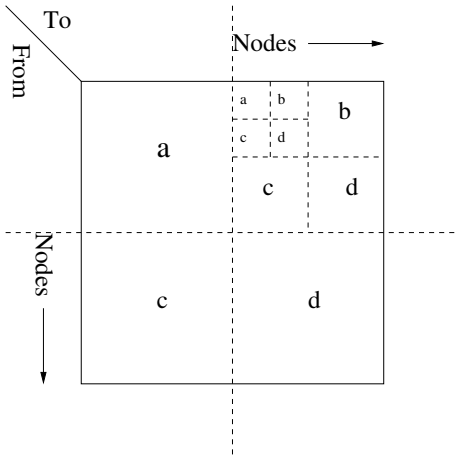


Figure 1.9: The R-MAT model. The figure is gained from Ref. [20].

the adjacency matrix into four equal-sized partitions, and assigns edges within these partitions with a unequal probabilities:

1. Starting with an empty adjacency matrix, which represents a subgraph for edge assignment;

2. Assign edges into the matrix with probabilities a, b, c, d respectively (See Fig. 1.9).

The chosen partition is again subdivided into four smaller partitions, and the above procedure is repeated until the chosen partition is composed of a simple cell such as a single node. In Chen *et al.*'s method, they define some nodes as graph-dependent nodes. These graph-dependent nodes play the role of core nodes, and are used to identify communities. The community dynamics can be revealed by the community member changes, where these communities are mapped through graph-dependent nodes.

The main drawback of above computation-generated benchmarks is that the evolution of a dynamic network corresponds to a fixed probability. We may expect that in real networks communities may experience heterogeneous changes such as bursty node insertion probability, node deletion probability, link insertion probability or link deletion probability.

1.4.2 Real networks

Real networks are also used to show performances of algorithms, such as Karate, Football, Dolphins and Neural. When dealing with real data, the main issue is generally the ground truth or a fine and precise expertise on the data sets. Real networks are released by Newman and can be downloaded from `http://www-personal.umich.edu/~mejn/netdata/`.

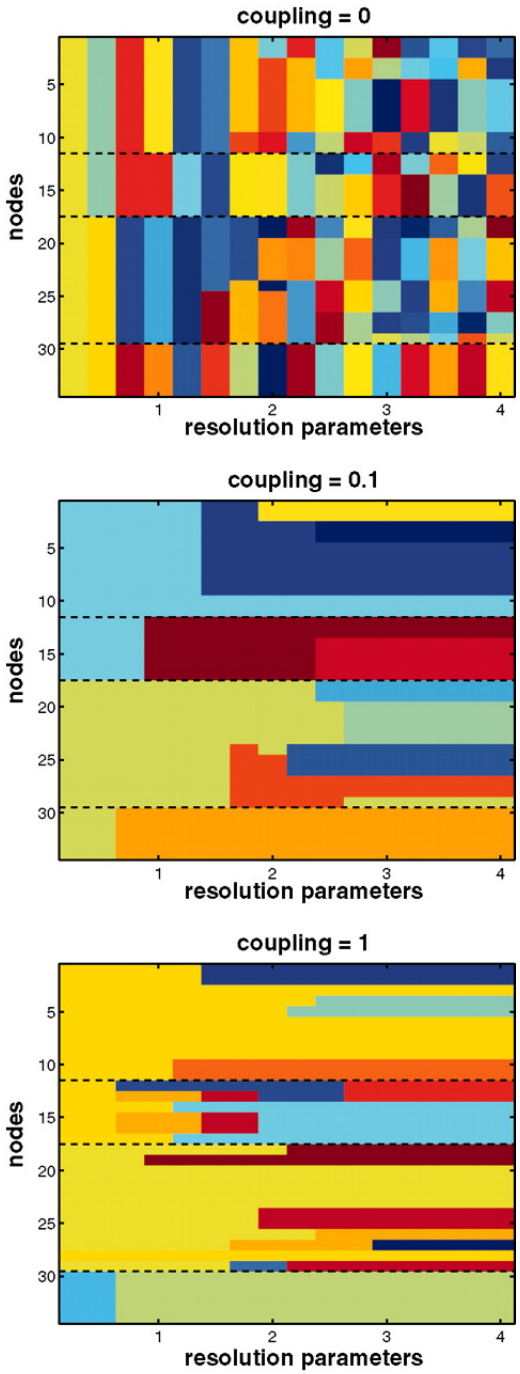


Figure 1.10: Multislice community detection of the Zachary Karate Club network [133] across multiple resolutions. Colors depict community assignments of the 34 nodes in each of the 16 slices (with resolution parameters $\gamma_\Delta = \{0.25, 0.5, \ldots, 4\}$), for $\omega = 0$ (top), $\omega = 0.1$ (middle), and $\omega = 1$ (bottom). Dashed lines bound the communities obtained using Newman-Girvan modularity [95]. The figure is gained from Ref. [88].

Mucha *et al.* [88] performed simultaneous community detection across multiple resolutions (scales) in the well-known Zachary Karate Club network, which encoded the friendships between 34 members of a 1970s university karate club [133]. Keeping the same unweighted adjacency matrix across *slices* (each *slice* represents a graph at a time step), the resolution associated to each slice is dictated by a specified sequence of γ_Δ parameters, such as $\gamma_\Delta = \{0.25, 0.5, 0.75, \ldots, 4\}$. In other words, given a serie of slices $\mathcal{A}_{ij\Delta} = \{A_{ij}(1), \ldots, A_{ij}(\Delta)\}$, these slices share the same unweighted adjacency matrix such as $\forall\ t_r, t_s\ , A_{ij}(t_r) = A_{ij}(t_s)$. Figure 1.10 depicts the community assignments obtained for coupling strengths $\omega = \{0, 0.1, 1\}$ between each neighboring pair of the 16 ordered slices. These results simultaneously probe all scales, including the partition of the Karate Club into four communities at the default resolution of modularity. Additionally, nodes that have an especially strong tendency to break off from larger communities are identified.

The previous definition for building benchmark graphs does not change interactions between nodes. Community structure changes observed are caused by tuning the resolution (scale) of the networks. Therefore, we can not use it to test the reliability of community dynamic detecting algorithms. Its other drawback is that the algorithm should use the same resolution parameter, otherwise it fails to test the performance of the algorithm in smoothing community evolution.

1.4.3 Comparing partitions

To measure the similarity between the built-in modular structure of a benchmark and the one delivered by an algorithm, several similarity measurements are possible. The most used similarity measurement is the *normalized mutual information*, which is based on information theory [28]. The idea is that, if two community structures are similar to each other, only little information is used to infer one community structure by given the other one.

The normalized mutual information is based on the *mutual information*. The mutual information for two random variables X, Y is denoted by $\mathrm{I}(X, Y)$, and is defined as:

$$\mathrm{I}(X, Y) = \sum_x \sum_y P(x, y) \log \frac{P(x, y)}{P(x)P(y)}$$

where $P(x)$ indicates the probability that $X = x$ (similarly for $P(y)$) and $P(x, y)$ is the joint probability of X and Y, *i.e.*, $P(x, y) = P(X = x, Y = y)$. Actually, $\mathrm{I}(X, Y) = \mathrm{H}(X) - \mathrm{H}(X|Y)$, where $\mathrm{H}(X)$ is the Shannon entropy of X and $\mathrm{H}(X|Y)$ is the the entropy of X conditional on Y.

Danon *et al.* [28] defined the normalized mutual information (NMI) for comparing the similarity between two partitions: $\mathcal{P}_x$ and $\mathcal{P}_y$. Let n_x and n_y denote the number of communities in the partition $\mathcal{P}_x$ and $\mathcal{P}_y$ respectively. The normalized mutual information is defined as:

$$\mathrm{NMI} = \frac{2\mathrm{I}(\mathcal{P}_x, \mathcal{P}_y)}{\mathrm{H}(\mathcal{P}_x) + \mathrm{H}(\mathcal{P}_y)}\ . \tag{1.7}$$

Let

$$\begin{aligned} \mathrm{I}(\mathcal{P}_x, \mathcal{P}_y) &= \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} P(\mathcal{C}_i, \mathcal{C}'_j) \log \frac{P(\mathcal{C}_i, \mathcal{C}'_j)}{P(\mathcal{C}_i)P(\mathcal{C}'_j)} \\ \mathrm{H}(\mathcal{P}_x) &= -\sum_{i=1}^{n_x} P(\mathcal{C}_i) \log P(\mathcal{C}_i) \end{aligned}$$

where n_x and n_y denote the number of communities in two partitions $\mathcal{P}_x$ and $\mathcal{P}_y$ respectively, $P(\mathcal{C}_i) = \frac{|\mathcal{C}_i|}{n}$ and $P(\mathcal{C}_i, \mathcal{C}'_j) = \frac{|\mathcal{C}_i \cap \mathcal{C}'_j|}{n}$.

Danon *et al.*'s normalized mutual information can be directly written in:

$$\mathrm{NMI} = \frac{-2 \sum\sum N_{ij} \log \frac{N_{ij} N}{N_{i\bullet} N_{\bullet j}}}{\sum_{i=1}^{n_x} N_{i\bullet} \log \frac{N_{i\bullet}}{N} + \sum_{j=1}^{n_y} N_{\bullet j} \log \frac{N_{\bullet j}}{N}} , \tag{1.8}$$

where N_{ij} represents the size of overlaps in communities i and community j, $N_{i\bullet}$ is the sum of i-th row in matrix N_{ij}, and $N_{\bullet j}$ is the sum of j-th column. The normalized mutual information is equal to 1 if the partitions are identical, whereas it has an expected value of 0 if the partitions are independent.

This normalized mutual information is extended for comparing covers in [70]. The normalized mutual information for covers $\mathcal{S}_x$ and $\mathcal{S}_y$ is denoted by $\mathrm{N}(\mathcal{S}_x|\mathcal{S}_y)$, and is defined as:

$$\mathrm{N}(\mathcal{S}_x|\mathcal{S}_y) = 1 - \frac{1}{2}\left[\mathrm{H}(\mathcal{S}_x|\mathcal{S}_y)_{\mathrm{norm}} + \mathrm{H}(\mathcal{S}_y|\mathcal{S}_x)_{\mathrm{norm}}\right] \tag{1.9}$$

where the normalized conditional entropy of $\mathrm{H}(\mathcal{S}_x|\mathcal{S}_y)_{\mathrm{norm}}$ (similarly to $\mathrm{H}(\mathcal{S}_y|\mathcal{S}_x)_{\mathrm{norm}}$) of the cover $\mathcal{S}_x$ with respect to $\mathcal{S}_y$ is defined as:

$$\mathrm{H}(\mathcal{S}_x|\mathcal{S}_y)_{\mathrm{norm}} = \frac{1}{n_x}\sum_{i=1}^{n_x} \frac{\mathrm{H}(S_i|\mathcal{S}_y)}{\mathrm{H}(S_i)} \text{ , where } S_i \in \mathcal{S}_x \text{ , } n_x = |\mathcal{S}_x|$$

The conditional entropy of S_i with respect to all the components of $\mathcal{S}_y$ is defined by:

$$\mathrm{H}(S_i|\mathcal{S}_y) = \min_{S'_j \in \mathcal{S}_y} \mathrm{H}(S_i|S'_j) \tag{1.10}$$

where $\mathrm{H}(S_i|S'_j)$ denotes the conditional entropy of a community S_i by given a community S'_j.

As Eq. 1.10 only counts the minimum $\mathrm{H}(S_i|S'_j)$, this extended normalized mutual information suffers from the following problem: some communities sharing few common nodes may be not be taken into account. Moreover, this normalized mutual information is not ideal: given two covers $\mathcal{S}_x, \mathcal{S}_y$, if only one community of $\mathcal{S}_x$ is divided into several small ones in $\mathcal{S}_y$ while all the others communities stay identical, the normalized mutual information is low because some communities have very low conditional entropy.

The main drawback of the above similarity measurements is that they are proposed for static graphs, and they do not consider the community dynamics. Therefore, we propose to measure the similarity between the found community structure and the ground

truth of dynamic graphs by counting the similarity between every pair of communities' evolution paths. We can write NMI (Eq. 1.7) by setting

$$\begin{aligned} P(\mathcal{C}_i) &= \frac{\sum_{t=1}^{\Delta} |C_i(t)|}{\sum_{t=1}^{\Delta} n(t)} \\ P(\mathcal{C}_i, \mathcal{C}_j) &= \frac{\sum_{t=1}^{\Delta} |C_i(t) \cap C_j(t)|}{\sum_{t=1}^{\Delta} n(t)} \end{aligned}$$

where $n(t)$ represents the nodes assigned to the partition in time t and $C_i(t)$ represents the observation of community $\mathcal{C}_i$ at time t (similarly for $C_j(t)$).

1.5 Conclusion

In this chapter, we have reviewed current research about community detection in dynamic networks. From our review, we observe that this issue has attracted a lot of work in recent years. Diverse approaches have been proposed and applied for detecting communities in dynamic networks and mining community dynamic models. A number of important issues stay open, such as benchmark graphs, overlapping community evolution. Finally, the main motivation encouraging us is to mine the relationship between the algorithmic communities compare to the reality. Why communities split, or merge, or disappear? What is the effect of overlapping nodes? To answer these questions, we study features behind graph topology and hope to learn more information.

Chapter 2

Overlapping communities and modularity

In real networks, it is common for nodes to belong to several communities. Communities may thus overlap with each other. For example, people may share the same hobbies in social networks [122], some predator species have the same prey species in food webs [66] and different sciences are connected by their interdisciplinary domain in co-citation networks [86]. However, most of heuristic algorithms are proposed for partition detection, whose results are disjoint communities. We devote this chapter to the detection of overlapping community structure.

Diverse methods have been proposed to detect overlapping community structure. However, the problem remains. For example, Palla et al. [98] have proposed the clique percolation method (CPM) to detect overlapping communities. This method is based on clique percolation: a k-clique (a complete subgraph of k nodes) is rolled over the network through other cliques with $k-1$ common nodes. In this way a set of nodes can be reached, which is identified as a community. One node can participate in more than one community, therefore overlaps naturally occur. The method, however, is not suitable for non-trivial networks, such as WikiTalk which is a sparse network consisting of star-like communities.

In order to provide the exhaustive information about overlapping community structure of a graph, we introduce a novel quality function to measure the quality of the overlapping community structure. This quality function is derived from the Hamiltonian and explains the quality of community structure through the energy of spin system.

In this chapter, we propose two different methods to detect overlapping nodes based on partitions. We can obtain overlapping community structure by adding these overlapping nodes to their related communities. Our first method is called clique optimization. Clique optimization aims at detecting *granular overlaps*. The clique optimization method is a fine grain scale approach. Each granular overlap is a node connected to distinct communities and it is highly connected to each community. Roughly speaking, a granular overlap is shared by several distinct communities while being intrinsically a

member of each of them. The second method is named fuzzy detection. Fuzzy detection is at a coarser grain scale and aims at identifying *modular overlaps*. *Modular overlaps* represent groups of nodes that have high community membership degrees with several communities. A modular overlap is itself a possible cluster/sub-community. As opposed to granular overlaps, modular overlaps imply the hierarchical organization of the graph: modular overlaps are sub-communities shared by several communities. The obtained results of the two methods are different. Since the two methods offer a different granularity scale (fine and coarse), they are complementary and meaningful in characterizing overlapping nodes.

The outline of this chapter is as follows. Section 2.1 introduces current work in cover detection. In Section 2.2, we describe our novel extension of modularity. In Section 2.3 and Section 2.4, we present clique optimization and fuzzy detection in details. We also show their performances when analyzing a real network in Section 2.5. In Section 2.6, we discuss our methods and give a brief conclusion in Section 2.7.

2.1 Related work on cover detection

In the following, we present a class of network clustering algorithms which allow nodes to belong to more than one community.

Baumes *et al.* [11] proposed a density metric for clustering nodes. In their method, nodes are added into clusters if and only if their fusion improves the cluster density. Under this condition, the results really depend on the initial seeds. Seeds can be a random node or disjoint communities. As shown in their results, there is a huge variation in the number of communities regarding the type of seed used.

Lancichinetti *et al.* has made efforts in cover detection including fitness-based function [71] and OSLOM (Order Statistics Local Optimization Method) [73]. The former is based on the local optimization of a k-fitness function, whose drawback is to introduce the tunable parameter k. The later uses the statistical significance [74] of clusters wich induces an expansive computational cost as it sweeps all nodes for each "worst" node. For the optimization, Lancichinetti *et al.* [73] propose to detect significant communities based on a partition. They detect a community by adding nodes, between which the togetherness is high. This is one of the popular techniques for overlapping community detection. There have similar endeavors like greedy clique expansion technique [76] and community strength-based overlapping community detection [127]. However, as all approaches applied Lancichinetti *et al.*'s k-fitness function, the results are limited by the tunable parameter k.

Some cover detection approaches are based on different basis. For example, Reichardt *et al.* [106] introduced the energy landscape survey method, and Sales Pardo *et al.* [110] proposed the modularity-landscape survey method to construct a hierarchical tree. They aim at detecting fuzzy community structure, whose communities consist of nodes having high probability to belong to the same group. As noticed in [110], they are mainly limited by a scalability factor in terms of network size.

Evans *et al.* [38] proposed to construct the *line graph* of the original network which

transforms the problem of node clustering into the problem of link clustering. It allows nodes to be shared by several communities. The main drawback is that, in their results, whatever the network, overlapping communities always exist.

2.2 Modified modularity for covers

2.2.1 A novel modularity

Modularity has been employed by a large number of community detection methods. However, it only evaluates the quality of partitions. Here, we introduce a novel extension for covers, which is combined with the Hamiltonian.

Many scientists deal with the problems in the area of computer science based on principles from statistical mechanics or analogies with physical models. When using spin models for clustering of multivariate data, the similarity measures are translated into coupling strengths and either dynamical properties such as spin-spin correlations are measured or energies are interpreted as quality functions. A ferromagnetic Potts model has been applied successfully by Blatt *et al.* [103]. Bengtsson and Roivainen [14] have used an antiferromagnetic Potts model with the number of clusters as input parameter and the assignment of spins in the ground state of the system defines the clustering solution. These works have motivated Reichardt and Bornholdt [106] to interpret the modularity of the community structure by an energy function of the spin glass with the spin states. The energy of the spin system is equivalent to the quality function of the clustering with the spins states being the community indices.

Let a community structure be represented by a spin configuration $\{\sigma\}$ associated to each node u of a graph G. Each spin state represents a community, and the number of spin states represents the number of communities of the graph. The quality of a community structure can thus be represented through the energy of spin glass. In [106], a function of community structure is proposed to

1. reward within-community links (internal links),
2. penalize within-community missing links (internal non-links),
3. reward non-links between different communities (external non-links), and
4. penalize existing links between different communities (external links).

Its expression is written as:

$$\begin{aligned}\mathcal{H}(\{\sigma\}) = & - \sum_{i\neq j} a_{ij} \underbrace{A_{ij}\delta(\sigma_i,\sigma_j)}_{\text{internal links}} + \sum_{i\neq j} b_{ij} \underbrace{(1-A_{ij})\delta(\sigma_i,\sigma_j)}_{\text{internal non-links}} \\ & + \sum_{i\neq j} c_{ij} \underbrace{A_{ij}(1-\delta(\sigma_i,\sigma_j))}_{\text{external links}} - \sum_{i\neq j} d_{ij} \underbrace{(1-A_{ij})(1-\delta(\sigma_i,\sigma_j))}_{\text{external non-links}}\end{aligned}$$

where σ_i denotes the spin state (or community index) of node i, and $a_{ij}, b_{ij}, c_{ij}, d_{ij}$ denote the weights of different contributions, respectively. The Kronecker delta symbol $\delta(\sigma_i, \sigma_j)$ yields 1 if and only if $\sigma_i = \sigma_j$ and 0 otherwise.

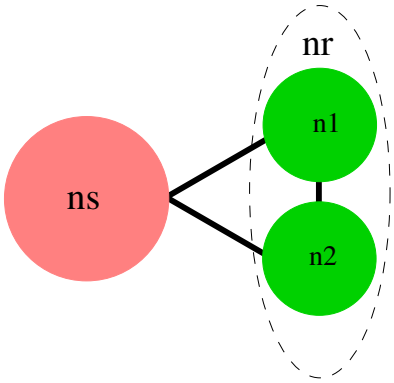


Figure 2.1: Example of $[\cdot]_{p_{ij}}$, where the union of clusters n_1 and n_2 is n_r such that $n_1 \cup n_2 = n_r$ and the cluster n_s belongs to the rest of the graph.

Let the weights on existing links be equal, *i.e.*, $a_{ij} = c_{ij}$. (Similarly for non-links, we have $b_{ij} = d_{ij}$). Then, only internal links and non-internal links are considered. A convenient choice to balance the importance of internal links and non-internal links is $a_{ij} = 1 - \gamma p_{ij}$ and $b_{ij} = \gamma p_{ij}$, where γ is a parameter and p_{ij} denotes the probability of a link existing between nodes i and j, normalized such that $\sum_{i \neq j} p_{ij} = 2m$.

A further simplified *Hamiltonian* for measuring the quality of a community structure, is written as:

$$\mathcal{H}(\{\sigma\}) = -\sum_{i \neq j} (A_{ij} - \gamma p_{ij})\, \delta(\sigma_i, \sigma_j) \tag{2.1}$$

We also can write the function (Eq. 2.1) in the following two ways:

$$\mathcal{H}(\{\sigma\}) = -\sum_{s} (m_{ss} - \gamma [m_{ss}]_{p_{ij}}) = -\sum_{s} c_s \tag{2.2}$$

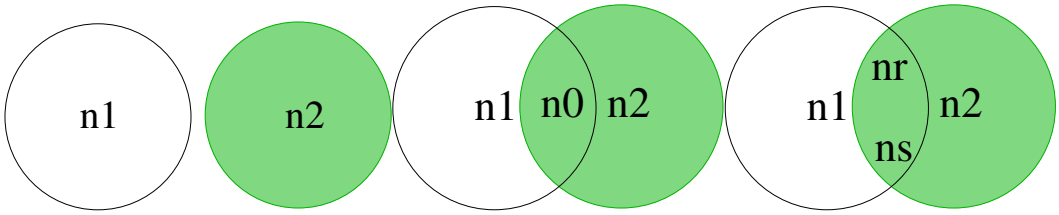
and

$$\mathcal{H}(\{\sigma\}) = \sum_{s<r} (m_{sr} - \gamma [m_{sr}]_{p_{ij}}) = \sum_{s} a_{sr} \tag{2.3}$$

where for each community $\mathcal{C}_s$, we note m_{ss} the number of links within $\mathcal{C}_s$, m_{sr} represents the number of links between a community $\mathcal{C}_s$ and another community $\mathcal{C}_r$, $[m_{ss}]_{p_{ij}}$ and $[m_{sr}]_{p_{ij}}$ are the expected number of links given a link distribution p_{ij}. The cohesion of $\mathcal{C}_s$ is noted c_s and a_{sr} represents the adhesion between a community $\mathcal{C}_s$ and another community $\mathcal{C}_r$.

We can assume diverse expressions of $[\cdot]_{p_{ij}}$, which is an expectation under the link distribution p_{ij}. In case of Fig. 2.1 for disjoint clusters n_1 and n_2, the choice should satisfy the following:

1. when n_s is a cluster belonging to the rest of the graph, $[m_{1s}]_{p_{ij}} + [m_{2s}]_{p_{ij}} = [m_{1+2,s}]_{p_{ij}}$;



(a) $n_1 \cap n_2 = \emptyset$ (b) $n_{01} \cap n_{02} = n_0$ (c) $n_{rs1} \cap n_{rs2} = n_r \cup n_s$

Figure 2.2: Let us denote the union of the clusters n_0 and n_1 by n_{01}. Similarly, we denote the union of the clusters n_0 and n_2 by n_{02}, the union of the clusters n_r and n_s by n_{rs}, the union of the clusters n_1, n_r and n_s by n_{rs1} and the union of the clusters n_2, n_r and n_s by n_{rs2}. Three different subdivisions of the community n_3: (a) two disjoint sub-communities n_1, n_2 ; (b) two overlapping sub-communities n_{01}, n_{02} sharing a cluster n_0; and (c) two overlapping sub-communities n_{rs1}, n_{rs2} sharing two clusters n_r , n_s, where n_r, n_s are disjoint sub-communities of n_0 such as $n_r \cap n_s = \emptyset$ and $n_r \cup n_s = n_0$.

2. when n_r is an union cluster composed of n_1 and n_2, $[m_{rr}]_{p_{ij}} = [m_{11}]_{p_{ij}} + [m_{22}]_{p_{ij}} + [m_{12}]_{p_{ij}}$.

Similarly, we give a relation for the cohesion of a community n_3 (the whole graph) and two sub-communities n_1 and n_2 with an empty intersection such as $n_1 \cup n_2 = n_3$ and $n_1 \cap n_2 = \emptyset$ (See Fig. 2.2 (a)). From Eq. 2.2 and Eq. 2.3, we can easily prove:

$$c_3 = c_1 + c_2 + a_{12} \tag{2.4}$$

where c_3 denotes the cohesion of n_3 that is the union of n_1 and n_2 with an empty intersection, a_{12} denotes the adhesion between n_1 and n_2, c_1 and c_2 are the cohesions of sub-communities n_1 and n_2 respectively.

Furthermore, we can give the relations for the cohesion of n_3 and two sub-communities n_1 and n_2 in other cases (See Fig. 2.2).

In the subdivision (See Fig. 2.2 (b)), there is an overlapping cluster n_0 between n_{01} and n_{02}. We write the cohesions for sub-communities n_{01} and n_{02} as:

$$\begin{cases} c^0_{01} = c^0_0 + c_1 + a^0_{01} \\ c^0_{02} = c^0_0 + c_2 + a^0_{02} \ , \end{cases}$$

where c^0_{01} and c^0_{02} denote the cohesion of the sub-communities n_{01} and n_{02} respectively, a^0_{01} and a^0_{02} denote the adhesion between n_0 and n_1, n_2. Here, n_0 is shared by n_{01} and n_{02}.

For the adhesion, we have:

$$a^0_{01,02} = a^0_{01} + a^0_{02} + a_{12}$$

between n_{01} and n_{02}.

For the union of $n_3 = n_{01} \cup n_{02}$, we obtain

$$\begin{aligned} c_3 &= c_0 + c_1 + c_2 + a_{01} + a_{02} + a_{12} \\ &= 2c_0^0 + c_1 + c_2 + 2a_{01}^0 + 2a_{02}^0 + a_{12} \, . \end{aligned}$$

So we derive

$$c_0^0 = \frac{1}{2}c_0 \, , a_{01}^0 = \frac{1}{2}a_{01} \text{ and } a_{02}^0 = \frac{1}{2}a_{02} \, . \tag{2.5}$$

In the subdivision (See Fig. 2.2 (c)) such as $n_r \cup n_s = n_0$, we replace c_0 and c_0^0 by

$$\begin{cases} c_0 = c_r + c_s + a_{rs} \\ c_0^0 = c_r^r + c_s^s + a_{rs}^{rs} \, , \end{cases} \tag{2.6}$$

where c_r^r and c_s^s denote the cohesion of overlapping sub-communities n_r and n_s respectively. a_{rs}^{rs} denotes the adhesion between overlapping sub-communities n_r and n_s, which satisfies $a_{rs}^{rs} = \frac{1}{2}a_{rs}$ due to Eq. 2.5.

Therefore, we propose the contribution of a_{rs} for all communities $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$:

$$\sum_1^k \frac{1}{|d_r \cup d_s|} a_{rs} = \frac{|d_r \cap d_s|}{|d_r \cup d_s|} a_{rs} \, , \tag{2.7}$$

where d_r and d_s denote the community memberships of n_r and n_s, respectively.

With the Hamiltonian (Eq. 2.1), we rewrite the modularity Q 1.2 as:

$$Q = -\frac{1}{m}\mathcal{H}(\{\sigma\}) \, . \tag{2.8}$$

Consequently, we can write the quality of an overlapping community structure in the form of the modularity function:

$$Q_{ov} = \frac{1}{2m} \sum_{i \neq j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \frac{|d_i \cap d_j|}{|d_i \cup d_j|} \, , \tag{2.9}$$

where d_i and d_j are memberships of nodes i and j, respectively. For a pair of nodes i and j always belonging to the same community such as $d_i \cap d_j = d_i \cup d_j$, their contribution to the modularity is $\left(A_{ij} - \frac{k_i k_j}{2m} \right)$. For a pair of nodes i and j never belonging to the same community such as $d_i \cap d_j = \emptyset$, their contribution is 0. Otherwise, their contribution is within the range of $\left[0, \left(A_{ij} - \frac{k_i k_j}{2m} \right) \right]$. Furthermore, if the found community structure is a strict partition, its quality Q_{ov} is equal to the initial modularity Q defined by the Equation 1.2.

2.2.2 Existing modularity for covers

There are other extensions of modularity designed to evaluate the quality of overlapping community structure. However, we are going to prove that they fail to satisfy above necessary constraints.

In the case Fig. 2.2 (c), we assume that n_r is an overlapping node v_i. Similarly for n_s, n_s is an another overlapping node v_j which connects to v_i. The union of v_i and v_j is n_0 such that $n_0 = v_i \cup v_j$. The overlapping communities n_{01} and n_{02} are denoted by $\mathcal{C}_x$ and $\mathcal{C}_y$ of a graph G_{example}, respectively.

Let O_v be the number of communities to which node v belongs. Shen *et al.* [112] have introduced an extended modularity:

$$Q_{\text{shen}} = \frac{1}{2m}\sum_{i=1}^{n_c}\sum_{v\in\mathcal{C}_i, w\in\mathcal{C}_j, v\neq w}\frac{1}{O_v O_w}\left(A_{vw} - \frac{k_v k_w}{2m}\right)\delta(\sigma_v, \sigma_w) \tag{2.10}$$

From Eq. 2.8, it is easy to obtain $a^0_{01_{\text{shen}}}$ derived from Q_{shen} (Eq. 2.10):

$$a^0_{01_{\text{shen}}} = \frac{1}{2}\sum_{v\in n_0, w\in\mathcal{C}_x\setminus n_0}\left(A_{vw} - \frac{k_v k_w}{2m}\right) + \frac{1}{2}\left(A_{v_i v_j} - \frac{k_{v_i}k_{v_j}}{2m}\right)$$

It fails to satisfy $a^0_{01} = \frac{1}{2}a_0$ (Eq. 2.5), where

$$a_{01_{\text{shen}}} = \sum_{v\in n_0, w\in\mathcal{C}_x\setminus n_0}\left(A_{vw} - \frac{k_v k_w}{2m}\right) + 2\left(A_{v_i v_j} - \frac{k_{v_i}k_{v_j}}{2m}\right)$$

In other words, through the definition of Q_{shen}, we obtain different values of the quality in views of Fig. 2.2 (b) and Fig. 2.2 (c) although they represent the same cover.

In [89], Tamas Nepusz *et al.* haved proposed a variant of modularity measure, which is defined by:

$$Q_{\text{fuzzy}} = \frac{1}{2m}\sum_{i,j}\left(A_{ij} - \frac{k_i k_j}{2m}\right)s_{ij}$$

where $s_{ij} = \sum_{k=1}^{n_c} u_{ki}u_{kj}$. The membership degree between node i and community k, u_{ki} satisfies $\sum_{k=1}^{n_c} u_{ik} = 1$.

As we did previously, for node $v_k \in n_0$ in G_{example}, under the assumption: $u_{v_i\mathcal{C}_x} = u_{v_i\mathcal{C}_y} = u_{v_j\mathcal{C}_x} = u_{v_j\mathcal{C}_y} = \frac{1}{2}$, it is easy to obtain

$$s_{v_k v_w} = \begin{cases} 0 & v_w \notin \mathcal{C}_x \cup \mathcal{C}_y\ , \\ 0.5 & v_w \in \mathcal{C}_x \cup \mathcal{C}_y, v_w \notin n_0\ , \\ 0.25 & v_k \neq v_w \end{cases} \tag{2.11}$$

We obtain that

$$a^0_{01_{\text{fuzzy}}} = \frac{1}{2}\sum_{v\in n_0, w\in\mathcal{C}_x\setminus n_0}\left(A_{vw} - \frac{k_v k_w}{2m}\right) + \frac{1}{2}\left(A_{v_i v_j} - \frac{k_{v_i}k_{v_j}}{2m}\right)$$

It also does not satisfy $a^0_{01} = \frac{1}{2}a_0$ (Eq. 2.5) with $a_{01_{\text{fuzzy}}} = a_{01_{\text{shen}}}$.

By using the novel proposed modified modularity (Eq. 2.9), we obtain

$$a^0_{01_{\text{ov}}} = \frac{1}{2} \sum_{v \in n_0, w \in \mathcal{C}_x \setminus n_0} \left(A_{vw} - \frac{k_v k_w}{2m} \right) + \left(A_{v_i v_j} - \frac{k_{v_i} k_{v_j}}{2m} \right) .$$

It satisfies $a^0_{01} = \frac{1}{2}a_0$ (Eq. 2.5), therefore we consider that our novel modified modularity is more reasonable to evaluate the quality of overlapping community structure. However, we can not detect covers by optimizing it since overlapping nodes may degenerate the modularity value. For example, in the case Fig. 2.2 (b), the quality can be represented by

$$Q^{\text{cover}}_{ov} = -\frac{1}{m}\mathcal{H}(\{\sigma\}) = -\frac{1}{m}\left(c_0 + c_1 + c_2 + a^0_{01} + a^0_{02}\right)$$

where $a^0_{01} = \frac{1}{2}a_{01}$ and $a^0_{02} = \frac{1}{2}a_{02}$. And the quality of the partition is

$$Q^{\text{partition}}_{ov} = \begin{cases} -\frac{1}{m}\left(c_0 + c_1 + c_2 + a_{01}\right) \ , when\ \mathcal{P} = \{n_{01}, n_2\} \\ -\frac{1}{m}\left(c_0 + c_1 + c_2 + a_{02}\right) \ , when\ \mathcal{P} = \{n_1, n_{02}\} \end{cases}$$

We find $Q^{\text{cover}}_{ov} = Q^{\text{partition}}_{ov}$ when $a_{01} = a_{02}$; otherwise, $Q^{\text{cover}}_{ov} < Q^{\text{partition}}_{ov}$ due to $\min(a_{01}, a_{02}) < a^0_{01} + a^0_{02} = \frac{1}{2}a_{01} + \frac{1}{2}a_{02} < \max(a_{01}, a_{02})$. Thus, even in a toy example where clearly there is a clear overlap (See Fig 2.2 (b)), if the number of links between n_0 et n_1 differs from the number of links between n_0 et n_2 the quality of the cover will be less than the quality of the partition once the difference between the number of links is greater than 0.

To overcome this optimization issue, we propose two methods not based on modularity like function. One is called clique optimization for detecting *granular overlaps*, and the other is named fuzzy detection aiming at identifying *modular overlaps*. Although granular overlaps and modular overlaps are used to denote overlapping nodes shared by several communities, they are intrinsically different. Granular overlaps represent nodes that have high togetherness with distinct communities while modular overlaps denote sub-communities shared by several communities. Therefore, given a pair of communities, we may observe several modular overlaps shared by them, while there are only one group of granular overlaps.

Communities are groups of nodes which probably share common properties. For instance, communities are groups of proteins participating a specific function in protein-protein interaction networks; communities are groups of pages dealing with the same or related topics in the World Wide Web; communities are groups of customers with similar interests in the network of purchase relationships between customers and products of online retailers (*e.g.,* `www.amazon.com`). Communities may overlap, *i.e.*, distinct communities share many nodes. These overlapping nodes reveal the relationships between

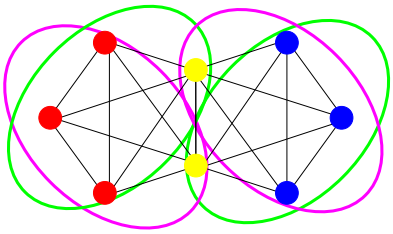


Figure 2.3: Two different partitions given by a partition detection method. One partition is shown in green and the other is in red. Both partitions have the same modularity.

communities. Detecting these overlapping nodes and characterizing them can help us to more understand communities.

One method to detect overlapping nodes is through cover detection, *i.e.*, communities in covers share overlapping nodes. Of course, we can detect covers by overlapping node detection, *i.e.*, only overlapping nodes are shared by several communities. Therefore, we propose the following methods to detect covers. Both them are composed of two phases: in the first phase, we detect overlapping nodes based on a partition, *i.e.*, the found overlapping nodes have strong connection or strong *membership degree* (how strong the nodes belong to the communities.); and in the second phase, we add these overlapping nodes to their related communities. Both phases are based on the same partition. Consequently, we obtain covers and become able to characterize overlapping nodes.

2.3 Clique optimization

The definition of community is not standard. There are different definitions, which depend on the context such as global definition, local definition and the community based on node similarity (See Section 1.1.1). Although communities are detected based on any of above definitions, the most commonly used one for overlapping community detection is that communities are clique-like objects. Given a clique, each member has connections with all other members. They are supposed to share common interests. The applications which detect clique-like communities like CPM [98], SCP [69] on social networks have good performance. Based on these observations, we propose to detect covers based on cliques.

2.3.1 Our proposed definition

On the graph example shown on Figure 2.3, its community structure is a cover composed of two k-cliques. By applying a partition detection method (a modularity optimization algorithm such as the Louvain algorithm), we obtain two different partitions with the same high modularity (See Fig. 2.3). We observe that overlapping nodes are separated by disjoint community boundaries. This observation motivates us to detect overlapping nodes through cliques, which are separated by disjoint community boundaries.

CPM [98] is one popular method for cover detection. It is designed to uncover the

community structure composed of *k-clique-communities*. A *k-clique-community* is the union of all k-cliques that can be reached from each other through a series of *adjacent k-cliques*. Two k-cliques are said to be *adjacent* if they share $k-1$ nodes. A *k-clique template* is a clique-like object. It is placed onto any k-clique of the network, and rolled to an adjacent k-clique by relocating one of its nodes and keeping its other $k-1$ nodes fixed. In CPM, each k-clique-community of a graph is a subgraph that can be fully explored by rolling a k-clique template on them. Each k-clique template is maximal for the 'rolling' process: there does not exist any other k-clique $k-1$ adjacent to the k-clique template. Through the definition of k-clique-community, each k-clique can be assigned to the community that contains its one adjacent k-clique.

Similarly, for each disjoint community of a partition, we propose to apply the k-clique adjacency rolling process on them. A clique is *adjacent* to a community if and only if both share $k-1$ common nodes. If a disjoint community can be rolled to an adjacent k-clique, all members of this k-clique can be assigned to this community. If a node can be assigned into more than one community, it is a *granular overlapping node*.

In the following, we give the definition of granular overlapping nodes in two senses:

Definition 3. *A node v is a k-granular overlapping node shared by ℓ communities $\mathcal{E} = \{\mathcal{C}_1, \ldots, \mathcal{C}_\ell\}$ in a strong sense if it belongs to a clique K adjacent to these communities, that is: $\forall \mathcal{C}_i \in \mathcal{E},\ |K \cap \mathcal{C}_i| \geq k-1$.*

Definition 4. *A node v is a k-granular overlapping node shared by ℓ communities $\mathcal{E} = \{\mathcal{C}_1, \ldots, \mathcal{C}_\ell\}$ in a weak sense if it is involved in ℓ' cliques $\mathcal{K} = \{K_1, \ldots, K_{\ell'}\}$ which are adjacent to them, that is: $\forall \mathcal{C}_i \in \mathcal{E},\ \exists K_j \in \mathcal{K}$ such that $|K_j \cap \mathcal{C}_i| \geq k-1$.*

Remark: Clearly an overlapping node in the strong sense is also an overlapping node in the weak sense, whereas the converse is not true.

Algorithm 2 A k-clique detection

Input: $e = (i^{\text{ini}}, j^{\text{ini}})$, N_{ij}^{ini}, k
Output: $\mathcal{K}$ a set of nodes describing a k-clique

$\mathcal{N} \leftarrow N_{ij}^{\text{ini}}$, $\kappa \leftarrow k-2$, $\mathcal{K} \leftarrow \{i, j\}$
while $\kappa > 0$ **do**
  if $\kappa = 1$ **then**
    Add a node $v \in \mathcal{N}$ to $\mathcal{K}$: $\mathcal{K} \leftarrow \mathcal{K} \cup v$
    $\kappa \leftarrow \kappa - 1$
  else
    Add a pair of connected nodes $\{i^{\text{pic}}, j^{\text{pic}}\} \subseteq \mathcal{N}$ to $\mathcal{K}$: $\mathcal{K} \leftarrow \mathcal{K} \cup \{i^{\text{pic}}, j^{\text{pic}}\}$
    $\mathcal{N} \leftarrow \mathcal{N} \cap N_{ij}^{\text{pic}}$
    $\kappa \leftarrow \kappa - 2$
  end if
end while
Return $\mathcal{K}$

Algorithm 3 Clique optimization

Input: $\mathcal{G} = (V, E)$, k
Output: $\mathcal{S} = \{S_1, \ldots, S_{n_c}\}$ an overlapping community covering of V
1: Obtain a partition $\mathcal{P} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{n_c}\}$ by running an efficient partition detection algorithm on the graph $\mathcal{G}$.
2: $\mathcal{S} \leftarrow \mathcal{P}$
// STEP 1: Find cliques which are $k-$adjacent to communities
3: **for all** Edges connecting one granular overlapping node candidate **do**
4: Find a clique K_j, which is $k-$adjacent to at least one community
5: Find all communities $\mathcal{E}_j = \{\mathcal{C}_1, \ldots, \mathcal{C}_\ell\}$ k-adjacent to K_j: $\forall \mathcal{C}_i \in \mathcal{E}_j$, $|K_j \cap \mathcal{C}_i| \geq k-1$
// STEP 2: Update overlapping communities
6: **for all** k-adjacent communities $\mathcal{C}_i \in \mathcal{E}_j$ **do**
7: Merge K_j to $\mathcal{C}_i$:$S_i \leftarrow \mathcal{S}_i \cup K_j$
8: **end for**
9: **end for**
10: Return $\mathcal{S}$

2.3.2 The clique optimization algorithm

Our clique optimization is proposed to detect k-granular overlapping nodes for cover detection. This algorithm consists of two phases: based on a partition, the first phase is to detect cliques which are k-adjacent to communities; the second phase is merging the above detected cliques into communities. The algorithm is sketched in Algo. 3. We describe it in details below.

After obtaining a partition by running an efficient partition detection algorithm (such as the Louvain algorithm) on the graph (line 1), we start our first phase.

In order to detect cliques, we use a k-clique detection algorithm (Algo. 2). It starts by one edge $e = (i^{\text{ini}}, j^{\text{ini}})$. Then this algorithm proceeds by collecting all nodes that are neighbors of both nodes $N_{ij}^{\text{ini}} = N_{i^{\text{ini}}} \cup N_{j^{\text{ini}}}$, where N denotes neighborhood. Now, when the edge $e = (i^{\text{ini}}, j^{\text{ini}})$ is added, each $k-2-$clique contained in the set $\mathcal{N}$ ($\mathcal{N}$ is initialized by N_{ij}^{ini}) will give rise to a new k-clique (lines 2 – 11 in Algo. 2). Therefore, all newly formed k-cliques are found by detecting all the $k-2-$ cliques in the $\mathcal{N}$, where $\mathcal{N}$ is iteratively updated through the selected edges $(i^{\text{pic}}, j^{\text{pic}})$. For commonly used small clique sizes, this is very fast: for 3-cliques, $k-2$-clique is a single node, while for $k = 4$, all connected pairs of nodes in $\mathcal{N}$ give rise to a new 4-clique.

We define a node to be a *granular overlapping node candidate* if its external degree is at least $k-1$. In the first phase (line 3 – 9), we detect all cliques which are k-adjacent to communities. A simple resolution is based on edges connecting one granular overlapping node candidate to detect a clique which is k-adjacent to at least one community. Chosen a granular overlapping node candidate, when a $k-1-$clique whose $k-1$ nodes belong to the same community is found from $\mathcal{N}$ ($\mathcal{N}$ is initialized by the neighbourhood of the chosen granular overlapping node candidate), we find another $k-1$ clique whose members belong to another community from the current $\mathcal{N}$. The final clique is k-adjacent to at

least one community.

Next, we merge this clique to communities in the second phase (line 6 – 8). For each clique which shares sets of $k-1$ nodes with one community, we merge them. If this clique shares sets of $k-1$ nodes with several communities, we merge this clique into several communities. Finally, we obtain a cover where granular overlapping nodes are shared by overlapping communities.

In general, we detect granular overlapping nodes in a weak sense by setting $k = 4$, where 4-clique is the smallest cluster larger than a triangle. However, if more than half of nodes are identified as granular overlapping nodes in a weak sense by using $k = 4$, we restrict the definition such that the number of granular overlapping nodes should be less than half of the nodes in the graph.

The granular overlapping nodes in the strong sense can be used to characterize community overlaps when we observe many communities sharing a large number of granular overlapping nodes in a network. In this case, we use granular overlapping nodes in the strong sense to characterize community overlaps for the following reasons: *a*) granular overlapping nodes in the strong sense are granular overlapping nodes in the weak sense and *b*) the number of nodes sharing the same common interest with a granular overlapping node in the strong sense is larger than in the weak sense.

Since granular overlapping nodes in the strong sense are granular overlapping nodes in the weak sense, the obtained characteristics through granular overlapping nodes in the strong sense should be shared by granular overlapping nodes in the weak sense. When the number of overlapping nodes between communities is large enough (*e.g.*, more than 100), these overlapping nodes can be considered as a large community for the characterization[1]. Therefore, we expect to identify the common interest shared by these overlapping nodes. Although the weak overlapping nodes have dense connections between several communities, the number of nodes sharing the same common interest with a granular overlapping node in the strong sense is more than in the weak sense. For instance, the maximal clique containing one weak granular overlapping nodes may have k members. In other words, only $k-1$ nodes share the common interest with this weak granular overlapping node. However, for a strong granular overlapping node, at least $2k-2$ nodes share the common interest.

The worst-case complexity of clique optimization is in $\mathcal{O}(n^k k^2)$: there are $\mathcal{O}(n^k)$ subgraphs to check, each of which has $\mathcal{O}(k^2)$ edges, where n represents the number of nodes whose external degree is at least 1. Note than n is the size of the community given by the partition algorithm and one may expect that n is smaller than the total number of nodes in the graph. Our method is faster than CPM [98] or SCP [69], since it only detects cliques separated by community boundaries.

Remark on directed graph: From the definitions given above, our clique optimization is defined for undirected and unweighted graphs. When analyzing an arbitrary

[1]In [77], best communities are defined to have a size scale between 10 to 100 nodes. Therefore, when the number of overlapping nodes is above 100, it is better to treat it as a community for the characterization.

system, one could decide that the directionality of the links could be ignored if it makes sense. If $u \to v$ means that the entity u is in interaction with the entity v, we may want to infer that $v \to u$ remains valid, yielding $u \leftrightarrow v$.

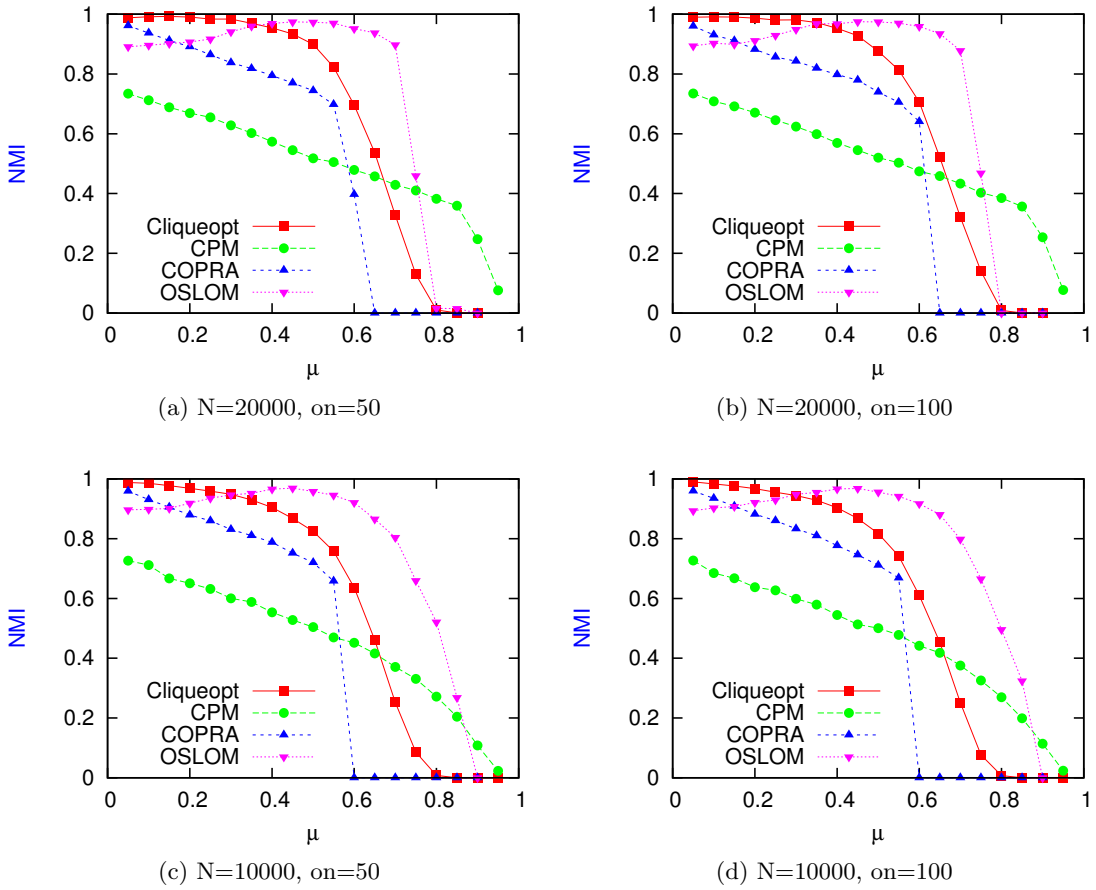
Remark on weighted graphs: If connections are weighted, a threshold weight ω^* is used to prune weak links and keep those that are stronger than ω^*. Depending on the weight distribution, the threshold could be $\omega^* = \frac{1}{2m}\sum_{v=1}^{n} k_v$, where k_v is the weighted degree of node v. If we want to keep all links, ω^* is simply set to zero. If the threshold weight is increased, the number of edges is decreased and so is the number of overlapping nodes. Note that, if ω^* is increased, the granular overlapping nodes should have stronger links to their related communities.

2.3.3 Benchmark graphs

We are now going to test the performances of clique optimization. We have considered a set of synthetic networks and a real network for which the community structure is known. We show the accuracy of our method through the *normalized mutual information* (NMI) [71] by comparing the computed covers to a ground truth. The higher the variation of information is, the more similar two covers are. If two covers are identical, NMI is 1. The results obtained by our clique optimization on the following benchmark graphs are good and presented bellow.

Synthetic networks

In Fig. 2.4, we present the comparison between our clique optimization heuristic and other cover detection algorithms including CPM [98], COPRA [52] and OSLOM [73]. Figure 2.4 presents the NMI of the results of all selected algorithms applied to LFR benchmarks [74]. LFR benchmarks are constructed by using a series of parameters: N the number of nodes, k the average degree, $\max_k$ the maximum degree, number of overlapping nodes on, the number of overlapping community memberships om and a mixing parameter μ. The *mixing parameter* μ is the ratio of intra-community to inter-community connections. For each overlapping node u shared by ν_u communities, if it belongs to community ξ, its adjacent links to ξ satisfies: $k_u^{\xi} = k_u^{in}/\nu_u$. As we can see, clique optimization performs near perfectly for small μ and small portion of overlapping nodes on/N: the NMI obtained if roughly greater than 0.9 when $\mu < 0.5$. It outperforms all other heuristic when $\mu \leq 0.3$ and has only a lower NMI than OSLOM when $\mu > 0.3$. Such case could be early explain since OSLOM only detects *significant communities*. A *significant community* is a group of nodes having a larger density of internal connections than of external links. If a node can not improve any community's significance (the difference between the internal connection density and external connection density), it is defined as an individual node and it is not considered in the community structure which changes the rules of the comparison.



(a) N=20000, on=50

(b) N=20000, on=100

(c) N=10000, on=50

(d) N=10000, on=100

Figure 2.4: Tests of our clique optimization on computer generated networks with known community structure and comparison with CPM [98], CORPA [52] and OSLOM [52]. Here, x-axis denotes the varying mixing parameter μ and y-axis represents the average NMI of 50 samples by comparing the found community structure and the ground truth. Besides the number of nodes N, the number of overlapping nodes *on* and the tunable parameter μ, the other parameters are identical: average degree $k = 20$, maximum degree $\max_k = 300$, minus exponent for the degree sequence $t_1 = 2$, minus exponent for the community size distribution $t_2 = 1$, minimum community sizes $\min_c = 10$, maximum for community $\max_c = 300$, and number of memberships of overlapping nodes $\mathrm{om} = 2$.

Yeast protein complexes

To perform further tests, we consider yeast protein complexes data base (See Fig 2.5). The combined-AP/MS network[2] describes 9070 interactions among 1622 proteins. In order to compare the results to a ground truth, we use a catalogue of protein complexes provided by CYC2008 [103]. All results are shown in Tab. 2.1.

[2]Available at `http://interactome.dfci.harvard.edu/S_cerevisiae/`

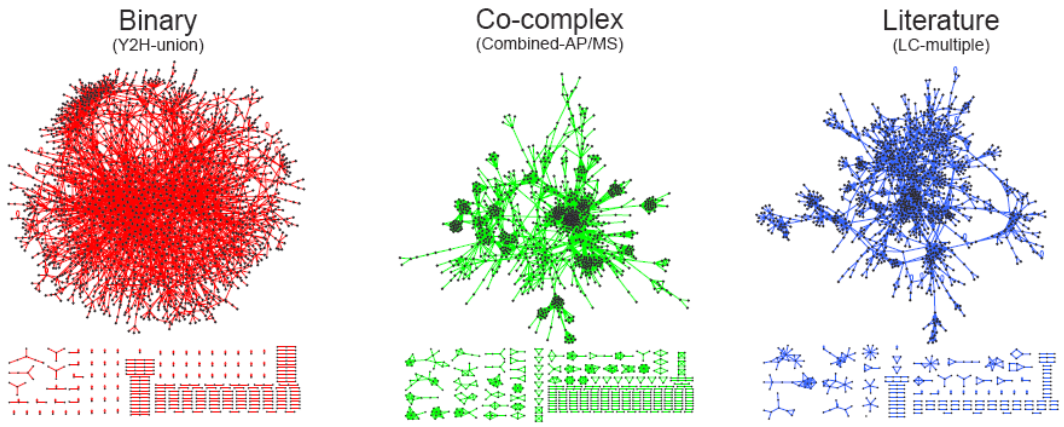


Figure 2.5: Graphical representation of three different types of yeast interactome datasets. (taken from *High-Quality Binary Protein Interaction Map of the Yeast Interactome Network*, Yu *et al.*, Science 2008.)

Method	NMI	Sensitivity	Specificity	Accuracy	Modularity Eq. 2.9
Clique Optimization	0.824323	0.514852	0.874587	0.6947195	0.772569
Fuzzy detection	0.702184	0.970297	0.290757	0.630527	0.866759
CPM	0.699512	0.287129	0.801471	0.5442995	0.816893
OSLOM [73]	0.52039	0.257426	0.965677	0.6115515	0.662716
Copra [52]	0.517806	0.118812	0.967657	0.5432345	0.888672

Table 2.1: Results of different overlapping community detections on Yeast protein complexes, in views of NMI, sensitivity, specificity, accuracy and modularity. The results of fuzzy detection are used to show its performance in identifying granular overlaps. As we can see, its advantage is not obvious.

We see that clique optimization identifies protein complexes with a high degree of success. By comparing to other overlapping detection techniques, it provides the highest NMI [71]. Remind that NMI measures the similarity between the results and the ground truth. We also provide additional measures: sensitivity, specificity, accuracy and modularity. *Sensitivity* is related to the ability to identify the real overlapping nodes, which is the proportion of real overlapping nodes among the found overlapping nodes. The low sensitivity of clique optimization is caused by our definition of k-granular overlapping nodes, *i.e.*, not all real overlapping nodes participate in k-cliques. *Specificity* is related to the ability to identify non-overlapping nodes, which is the proportion of non-overlapping nodes among all found non-overlapping nodes. The *accuracy* is a "balanced accuracy", which is the sum of sensitivity and specificity. The accuracy focuses on the capacity of detecting overlapping nodes. One can observe that our clique optimization heuristic offers the highest accuracy score.

2.4 Fuzzy detection

In this section, we will introduce another method for cover detection named *fuzzy detection*. This novel cover detection heuristic aims at identifying modular overlaps. Modular overlaps are groups of nodes shared by communities. As mentioned above, there is a difference between *granular overlaps* introduced in the previous section and *modular overlaps*. Modular overlaps are related to the hierarchy organization. That is, modular overlaps are sub-communities shared by several communities.

2.4.1 Motivation

Our fuzzy detection algorithm is based on the Louvain algorithm [16]. The Louvain algorithm is an efficient partition detection algorithm that provides good partitions with high modularity. It consists of two phases that are iteratively repeated until no more positive gain of modularity is obtained. Initially, all nodes are assigned into a single community. Then, for each node whose move improves the modularity, it will be removed from its current community to the neighbor community which offers the largest gain of modularity. The first phase repeatedly and sequentially sweeps all nodes until no further improvement of modularity can be gained. The second phase builds a new meta graph based on communities found in the first phase. It aggregates nodes of the same community and builds a new network whose nodes are the communities. Once the second phase is completed, the first phase is reapplied to the new network. The two phases are iteratively applied until no more change in community structure or maximum modularity is achieved. In the following, we use iteration to denote the combination of these two phases. The partition found by this algorithm is hierarchical organized, the hierarchy height is determined by the number of iterations. The Louvain algorithm is extremely fast and provides highly optimized partitions with high modularity.

When running several times the Louvain algorithm on the same given network, we observe from a run to another that nodes may be grouped together with different community members in distinct partitions. Since the Louvain algorithm sweeps nodes in a non deterministic fashion (a random permutation of V), it naturally introduces instability which may be a weakness. It turns out that we can take benefit of this instability. By detecting nodes that jump from one community to another between distinct runs, we are in fact able to uncover nodes that have high community memberships with distinct communities. Such "*oscillating*" nodes can be considered as overlapping nodes. Therefore, we propose a fuzzy detection algorithm which detects groups of nodes having strong connection probability with several communities.

2.4.2 Fuzzy detection algorithm

To have the benefit of the potential Louvain algorithm instability [4], we force the algorithm to use a random seed at each run. The random seed makes the nodes be swept in a random permutation during the modularity optimization. Thus, different runs may produces different partitions. By repeating Louvain algorithm, we are able to compute, a

Algorithm 4 Louvain algorithm.

Input: $G = (V, E)$, l^* a level threshold
Output: $\mathcal{P}$ a partition

$l \leftarrow 0; G_0 \leftarrow G$
repeat
  $l \leftarrow l + 1$
  Initialize a partition $\mathcal{P}_l$ of $G_l(V_l, E_l)$
  // First phase: partition update
  repeat
    Nodes in a random permutation
    for all Nodes: $v \in V_l$ **do**
      Move from σ_v to one selected $\sigma_{v'}$ (v' is a neighbour of v)
    end for
  until no more change increases modularity
  // Second phase: Construct a new meta graph
  Replace each community by a node
  Replace connections between a pair of communities by one weighted edge
until $\mathcal{P}_l$ is not updated or $l = l^*$.
Return $\mathcal{P}$ corresponding to the roots of the hierarchical tree.

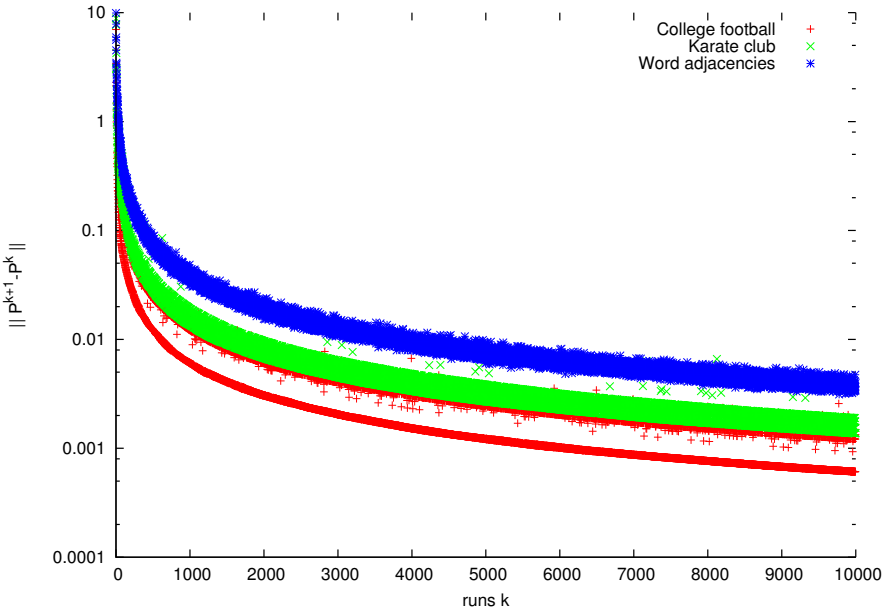


Figure 2.6: As the number of runs increases, the shape of the function value Eq. 2.12 gets closer and closer to 0. The figure shows results on College football [48], Karate club [133] and Word adjacencies [92].

Algorithm 5 Fuzzy detection.

Input: $G = (V, E)$, α^*, β^*
Output: $\mathcal{S}$ an overlapping community covering of V
// *STEP 1: Detect robust clusters*
$\mathbf{P}^0 \leftarrow 0$; $k \leftarrow 0$;modularity$_{\max} \leftarrow -\infty$
repeat
  $k \leftarrow k + 1$
  $\mathcal{P} \leftarrow$ Run the Louvain algorithm on G
  Update $\mathbf{P}^k$
  if modularity of $\mathcal{P}$ greater than modularity$_{\max}$ **then**
    Save the partition $\mathcal{P}$ in $\mathcal{P}_{\text{opt}}$ and update modularity$_{\max}$
  end if
until $\|\mathbf{P}^k - \mathbf{P}^{k-1}\| \leq \epsilon$
$\mathcal{P}_{\text{sc}} = \mathcal{P}_{\text{opt}}$
for all edge $e = (i, j)$ such that $p_{ij} < \alpha^*$ **do**
  Remove the external edge e from $\mathcal{P}_{\text{sc}}$
end for
// *STEP 2: Adjust the membership of robust clusters*
Input: $G = (V, E)$, $\mathcal{P}_{\text{sc}}$, $\mathcal{S} \leftarrow \mathcal{P}_{opt}$
for all $\mathcal{C}_i \in \mathcal{P}_{opt}$ **do**
  Identify community core: $\widehat{c}_i = \arg\max_{c_j \subseteq \mathcal{C}_i} |c_j|$
end for
Compute $\mathbf{P}_{c_i, c_j}$
for all $c_j \in \mathcal{P}_{\text{sc}}$ and $c_j \notin \{\widehat{c}_1, \ldots, \}$ **do**
  if $p_{c_j, \widehat{c}_i} \geq \beta^*$ **then**
    $S_i \leftarrow S_i \cup c_j$
  end if
end for
Return $\mathcal{S}$

co-appearance matrix $\mathbf{P} = [p_{ij}]_{n \times n}$. For each pair of nodes (i, j), p_{ij} of $\mathbf{P}$ represents the probability for the pair nodes i and j to appear in the same community. Having $p_{ij} = 1$ implies that nodes i and j are always in the same community while edges $e = (i, j)$ having a p_{ij} close to 0 implies that edge e connects two different communities. The underlying idea of fuzzy detection approach is thus to detect overlapping communities from a classical partition approach.

Detecting overlapping nodes also allows to detect more stable nodes that always belong together in the same community. In this algorithm, we use the notion of *community cores* to denote communities. Given a community, its *core* is a group of nodes offering high stability against random perturbation. To detect community cores, we're going to remove edges in order to keep only core nodes. First we remove all *external edges*, *i.e.*, all edges $e = (i, j)$, having a connection probability p_{ij} less than a threshold α^*. After this pruning phase, a set of disjoint robust cluster is obtained. A *robust cluster* is a

group of nodes connected by edges having in-cluster probability larger than or equal to α^*. Note that a given community may have several robust clusters. We choose the community core corresponding to the robust cluster having the maximum size. The notion of external edges was used in [46] where authors add a random noise over the weight of the edges of the network (equally distributed between $[-\sigma, \sigma]$). Once community cores are identified, we continue iteratively, following the Louvain approach. Similarly, in our method, we replace the robust clusters by supernodes and connect them through the connection between robust clusters. In this case, the weight of the edge between the supernodes is the sum of the weights of the edges between the identified robust clusters. We run again the Louvain algorithm to compute the probability of robust clusters and community cores to appear in the same community. Finally, we add each robust cluster to the community if they have a high community membership degree such as their probability of appearing in the same community is high.

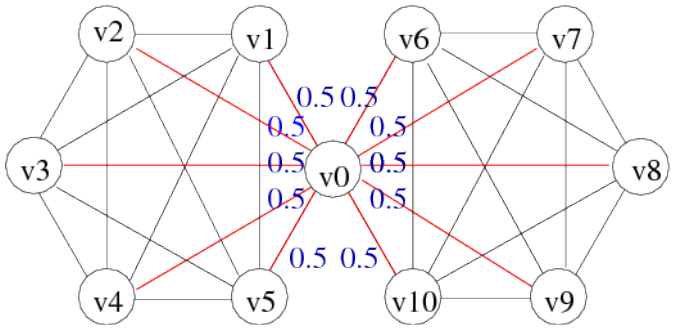


Figure 2.7: Illustration of our fuzzy detection on a toy graph which consists of two overlapping cliques. After removing all edges in low probability $p_{ij} = 50\%$ shown in red, robust clusters are obtained, concluding $\{v_1, v_2, v_3, v_4, v_5\}$, $\{v_6, v_7, v_8, v_9, v_{10}\}$, and a single v_0.

The global algorithm is shown in Algo. 5. First, (lines 2 – 9) we compute the co-appearance matrix $\mathbf{P} = [p_{ij}]_{n \times n}$ by running the Louvain algorithm of Algo. 4 several times with a random seed[3]. The number of runs is determined by the convergence criteria (line 9):

$$\|\mathbf{P}^{k+1} - \mathbf{P}^k\| = \sqrt{\frac{1}{m} \sum_{(i,j) \in E} \left(p_{ij}^{k+1} - p_{ij}^k\right)^2} < \varepsilon, \tag{2.12}$$

where $\mathbf{P}^k$ represents the result after k-th run and p_{ij}^k denotes the statistical probability of nodes i and j to belong to the same community after k-th runs (line 5) and ε is a

[3]Louvain algorithm is a hierarchical clustering algorithm. It iteratively merges small clusters to maximize modularity. Therefore, it provides a hierarchical tree (or dendrogram) to illustrate the hierarchical form of organization. If the level parameter is not set, Louvain algorithm gives the partition corresponding to the largest value of the modularity; otherwise, this algorithm returns the partition corresponding to the roots, that is the partition obtained in the last iteration.

small threshold. Figure 2.6 illustrates the convergence of the norm when running fuzzy detection algorithm. We observe that $\|\mathbf{P}^{k+1} - \mathbf{P}^{k}\|$ decreases as the number k of runs increases.

Then, we detect robust clusters $\{c_1, c_2, \ldots, c_s\} = \mathcal{P}_{\text{sc}}$ (lines 10 – 13). Given a partition $\mathcal{P}_{opt}$ which has the maximum modularity among all computed partitions obtained during the first phase, the robust clusters are detected by removing all edges having a probability p_{ij} lower that a given threshold α^* (typically $\alpha^* = 0.9$). A simple illustration is given in Fig. 2.7.

Finally in the second phase, we identify modular overlaps which have high community memberships with several communities. Given a community $\mathcal{C}_i \in \mathcal{P}_{\text{opt}}$, its core $\widehat{c}_i$ is the robust cluster $c_j \subseteq \mathcal{C}_i$ having the maximum size, such as:

$$\widehat{c}_i = \arg\max_{c_j \subseteq \mathcal{C}_i} |c_j| \tag{2.13}$$

We assign each robust cluster c_j to the community $\mathcal{C}_i$ if and only if their community membership $p_{c_j,\widehat{c}_i}$ is larger than a threshold β^* such as $p_{c_j,\widehat{c}_i} > \beta^*$ (typically $\beta^* = 0.1$). If one robust cluster is assigned to at least two communities, we call it a *modular overlaps*. Given a modular overlaps, its members are possible granular overlapping nodes. Only the granular overlapping node are required to have dense connection with related communities. The nodes shared by the same modular overlaps are not only required to have dense connection with related communities and also are required to have high internal modular degree (the number of links connected to other members within the robust cluster).

Fine tunning: In cases where a community consists of several robuster clusters of comparable size, one may tune and increase the value of α^* in order to refine the core identification.

Since fuzzy detection is used to identify modular overlaps, which are sub-communities shared by several communities, we restrict the modular overlaps to have a size greater than 3. We can now introduce the notion of *unstable nodes*, which are nodes connecting communities with few links but are observed to have high co-appearance probability with several communities. The Fig. 2.8 illustrates such case. Due to unstable nodes, we do not use fuzzy detection to identify granular overlaps. As shown in Tab. 2.1, the results of fuzzy detection may be degenerated by unstable nodes. Moreover, the method suffers from the classical resolution limit of modularity optimization [44]. Indeed, due to this resolution limit, two weakly connected communities may possibly be grouped together if their merge improves the modularity during modularity optimization phase. Therefore, we may observe some modular overlaps that are not real overlapping nodes but are the results of modularity optimization. We call them *unstable clusters*.

The running time of fuzzy detection mainly depends on the co-appearance matrix calculation. The complexity to find a partition by the Louvain algorithm is estimated by authors in [16] to be in $\mathcal{O}(m)$, where m is the number of edges in the network (the worst complexity is much higher, but in practice, on real network, Louvain algorithm performs very well). Thus the computational complexity of fuzzy detection is in $\mathcal{O}(Km)$, where

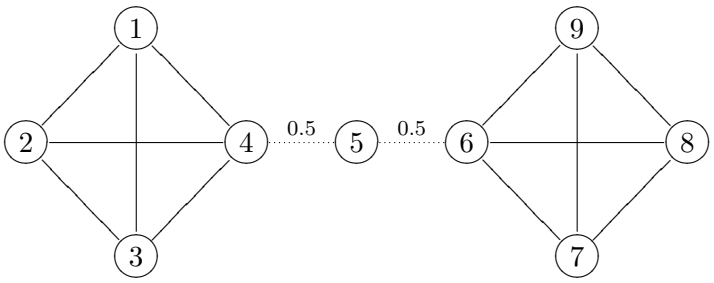


Figure 2.8: An example graph that contains a unstable node 5. Node 5 has a relatively high membership degrees with two communities ($p = 0.5$). However, it is connected to each community with only 1 link.

K is the number of runs of Louvain algorithm needed before reaching an acceptable convergence of $\mathbf{P}$. Once more, in practice, we take benefit of the efficient Louvain algorithm running time and our fuzzy detection is fast. We experiment storage limitation due to the matrices $\mathbf{P}^k$ and $\mathbf{P}^{k+1}$ more that time computing one.

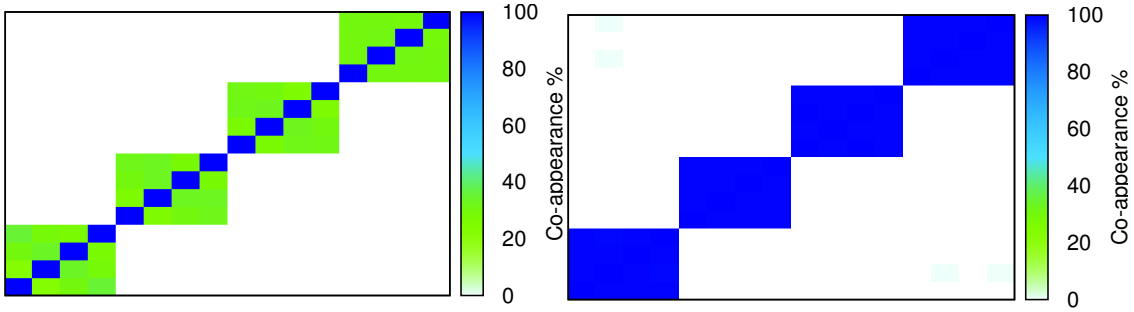
2.4.3 Benchmark graphs

In the following, we test the performances of fuzzy detection. We have considered a set of synthetic networks and a real network for which the community structure is known. The results show that our fuzzy detection algorithm extracts communities while preserving the *hierarchical organization* and also providing overlaps.

A community structure can be hierarchically ordered when the graph offers several levels of organization/structure at different scales. In this case, the community structure is *hierarchically constructed* by small communities at each level, all nested within large communities at higher levels. As an example, one may consider in a social network the granularity of the living place (town), the working place (school) and refine it toward the graduate or class level.

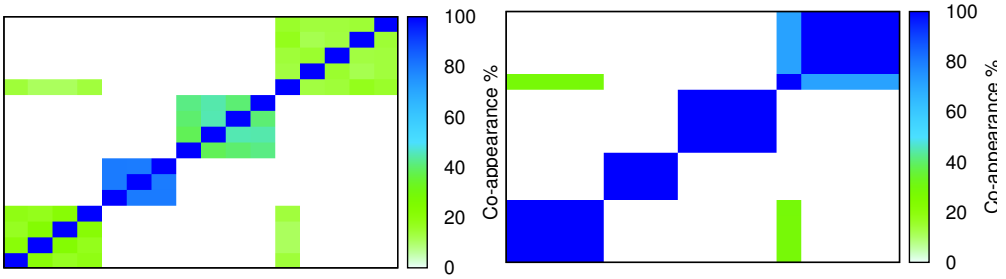
Synthetic graphs containing hierarchical structure

First, we apply the fuzzy detection algorithm to an artificial graph containing hierarchical structure [71]. The result is shown in Fig. 2.9. We observe that fuzzy detection extracts communities in hierarchical organization. The benchmark graph consists in 512 nodes, assigned into 16 groups of 32 nodes each. These 16 groups are ordered into 4 supergroups. The benchmark is constructed by assigning edges between nodes within the same micro-community. Each node has a micro-internal degree $k_1 = 41$. Then we assign edges between nodes belonging into different micro-communities but in the same macro-community. Each node has a macro-internal degree $k_2 = 17$. Finally we add edges between nodes to connect them to the rest of the network. All nodes have the same total degree $k = 64$ and an external degree $k_3 = 6$. This process constructs two



(a) $l^* = 1$ (b) Without l^*

Figure 2.9: The co-appearance matrix of synthetic networks containing a hierarchical structure. The color corresponds to the probability of nodes to be in the same community: the darker the color, the higher the probability; color is white if the probability is 0.0.



(a) $l^* = 1$ (b) Without l^*

Figure 2.10: The co-appearance matrix of artificial networks containing hierarchical structure. The color corresponds to the probability of nodes in the same community: the deep color represents the high probability; the color is white if the probability is 0%.

hierarchical levels: one consisting of 16 small groups, and the other one composed of 4 supergroups with 128 nodes each. Figure 2.9 (b) illustrates the co-appearance matrix by running the Louvain algorithm without fixing the level threshold l^* (See Algo. 4), while Figure 2.9 (a) provides the result by running the Louvain algorithm with $l^* = 1$. In both figures, the nodes are sorted in the same order corresponding to the robust clusters and the selected partition $\mathcal{P}_{\text{opt}}$. As the distinction among robust clusters is not clear in Fig. 2.9 (b), we use Fig. 2.9 (a) for the visualization. We observe 4 communities and 32 robust clusters, which agrees with the ground truth.

Remark that, when running our fuzzy detection to identify modular overlaps, we may need to increase the value of α^* to obtain a reasonable community core whose size is larger than the others within the same community. It occurs when one community contains several large robust clusters having comparable size.

Next, we apply the fuzzy detection algorithm to a random graph containing modular

overlaps. The graph is composed of 512 nodes, which belong to 12 groups, arranged into 4 supergroups and one group is shared by two supergroups. Every node has an average of $k_1 = 30$ links with nodes in the same micro-community, $k_2 = 13$ links with nodes in the same macro-community but different micro-community. In addition, each node has $k_3 = 5$ links with the rest of the networks. As the modular overlaps has macro-links with two communities, its nodes have a total degree $k = 61$ while the other nodes only have a total degree $k = 48$. Figure 2.10 illustrates the result. We observe two communities that share one modular overlap. Results show the good performance of fuzzy detection algorithm in uncovering modular overlaps.

College football network

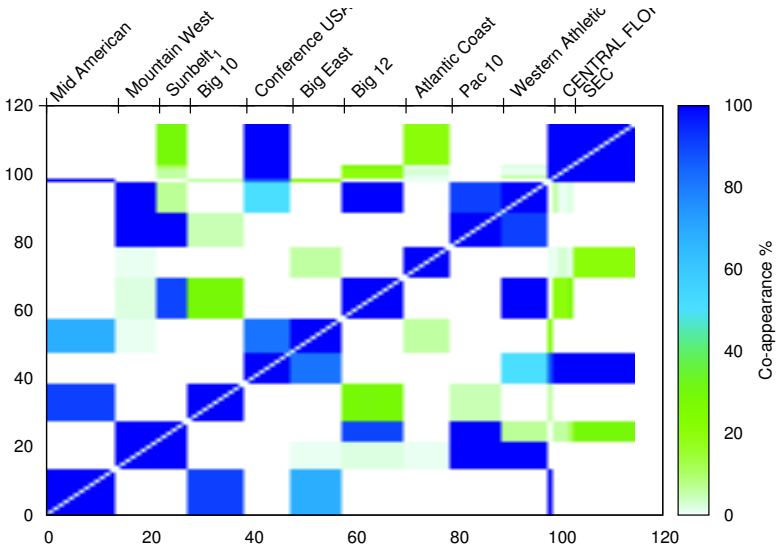


Figure 2.11: The co-appearance matrix of college football network by running our fuzzy detection. We order the nodes corresponding to their conferences and mark the conference indices. The color corresponds to the probability of nodes in the same community: the deep color represents the high probability; the color is white if the probability is 0%.

We also run the fuzzy detection algorithm to real networks. A famous real but small and tractable network is the *US college football* [48] . This network records the schedule of Division I games for the 2000 season: 115 nodes represent teams (identified by their college names) and 613 edges represent regular season games between the two teams they connect. What makes this network interesting [48] is that it incorporates a known community structure. The teams are divided into "conferences" containing around 8 to 12 teams each. Games are more frequent between members of the same conference than between members of different conferences, with teams playing an average of about

7 intra-conference games and 4 inter-conference games fraction of vertices classified correctly in the 2000 season. Inter-conference play is not uniformly distributed; teams that are geographically close to one another but belong to different conferences are more likely to play one another than teams separated by large geographic distances.

In Fig. 2.11, we illustrate the results: the community "Mountain West Sunbelt" is split into "Mountain West" and "Sunbelt_1", the community "Sunbelt SEC" has a possible subdivision into "Sunbelt_2"[4] and "SEC", and a node "CentralFlorida" is split from the community "Pac 10". Among them, only "Sunbelt_1" is identified as a modular overlaps. "CentralFlorida" has high membership degree with different communities, too. But it is a granular overlapping node rather than a modular overlaps. In reality, the team "CentralFlorida" did not belong to any conference, and the teams in the "Sunbelt" conference played nearly as many games against Western Athletic teams as they did within their own conference. Therefore, we consider fuzzy detection has a good performance in detecting modular overlaps for this real network.

2.5 Application to real networks: Complex System Science

In this section we consider the applications of clique optimization and fuzzy detection to a real network called Complex System Science. It is a co-citation network, whose dataset is composed of articles extracted from the ISI Web of knowledge. Article were published between 2000 and 2009. The network is composed of 141 163 nodes and 19 603 888 links. The nodes correspond to articles containing a set of keywords relevant to the field of complex systems. The weight of the links between articles is calculated through their common references (bibliographic coupling [65]). A link exists between two articles if they share references, meaning that they cite common work which may implies that they are dealing with a same scientific object/domain. More precisely, given two articles (nodes) i and j, each one having a set of references R_i (respectively R_j), there exists a link $e = (i, j)$ between i and j if i and j share at least one reference and the weight is measured by: $w_{ij} = \dfrac{|R_i \cap R_j|}{\sqrt{|R_i|\,|R_j|}}$.

For the visualization, we only show clusters which contain at least 100 nodes[5]. The partition of the graph is shown in Fig. 2.12. Each community corresponds to a unique color. Our obtained robust clusters are shown in Fig. 2.13. The color of each robust cluster corresponds to the relevant community in the partition shown in Fig. 2.12. Only robust clusters belonging to the same community in the partition share the same color.

In Fig. 2.12, we observe 12 communities. These communities can be identified by research topics or theoretical fields through studies in topic keywords, see Tab. B.1. We compute the frequency of topic keywords by aggregating the number of units (articles). For instance, if only one unite contains the topic keywords "Neurons", the corresponding

[4] We do not mark "Sunbelt_2" due to the visualization, since its position is too close to "CentralFlorida" in the figure.

[5] In [77], the community which has size roughly 100 nodes is good.

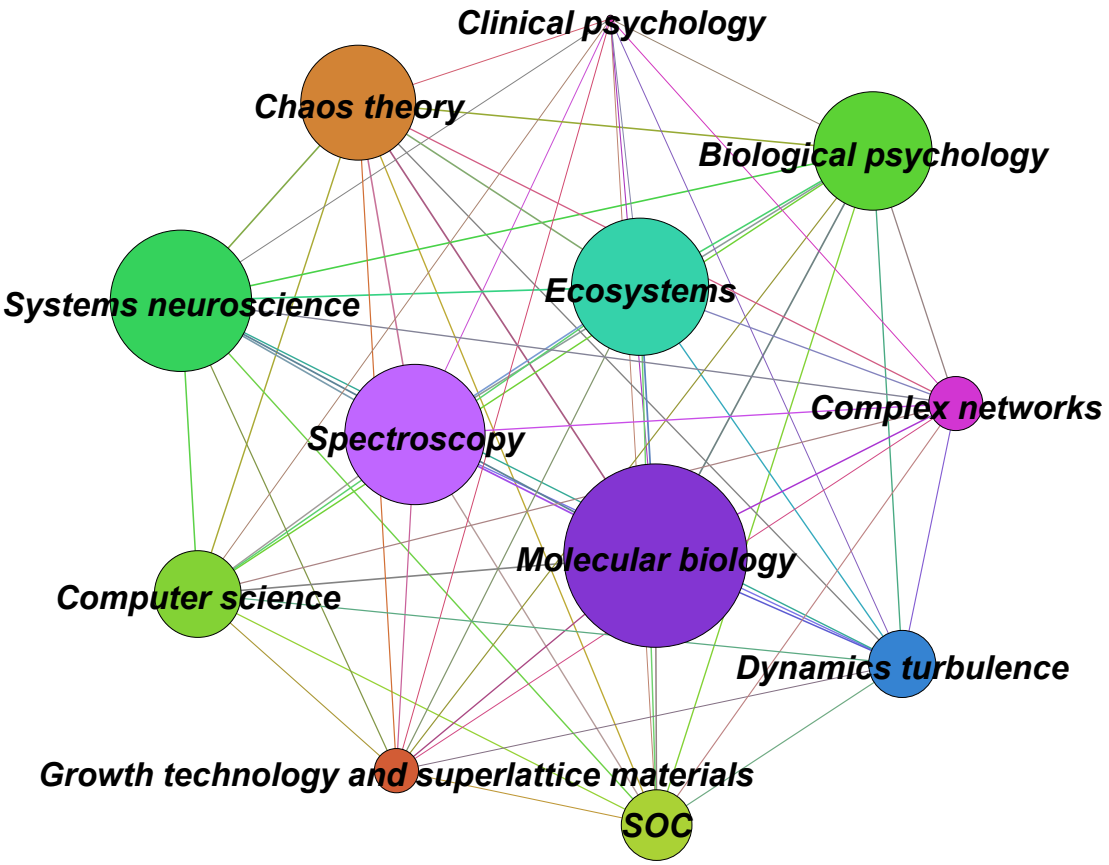


Figure 2.12: The community structure of Complex System Science, in which communities are identified by research topics or theoretical fields.

frequency is 1. In the figure, the light green community is identified by neuroscience: biology psychology. This community contains high frequent keywords (NEURONS, PERFORMANCE, CENTRAL-NERVOUS-SYSTEM) very general in neuroscience while some high frequent keywords (BRAIN, LONG-TERM POTENTIATION, DISEASE) seem to emphasize the study in the field of biological psychology. To our knowledge, biological psychology or behavioral neuroscience is the study of the biological substrates of behavior and mental processes. Physiological psychologists use animal models, typically rats, to study the neural, genetic, and cellular mechanisms that underlie specific behaviors such as learning and memory and fear responses. Cognitive neuroscientists investigate the neural correlates of psychological processes in humans using neural imaging tools, and neuropsychologists conduct psychological assessments to determine, for instance, specific aspects and extent of cognitive deficit caused by brain damage or disease.

Table B.2 shows results of clique optimization in identifying granular overlaps in a strong sense[6] with a choice of $k = 5$. We see the applications of chaos theory in

[6]Between several pairs of communities, their granular overlaps contain more than 100 nodes.

different disciplines including complex networks, nervous systems and ecosystems. We also observe the intermediation: visual cortex between neural networks and neuroscience: biological psychology. Visual cortex is one part of the visual systems, which receives visual information for processing images. These results are interesting in understanding the combination of different disciplines and applications.

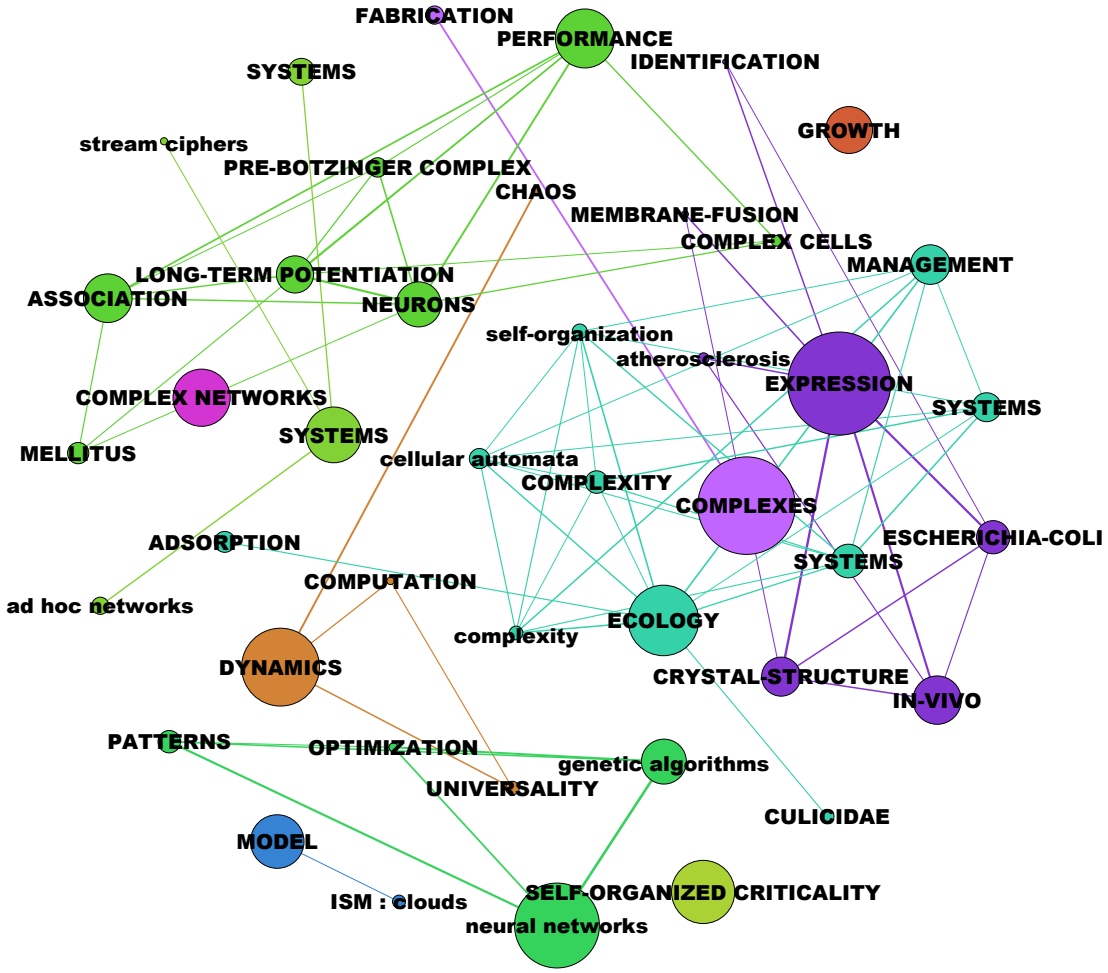


Figure 2.13: Results of fuzzy detection on Complex System Science. Robust clusters are marked by the highest frequent topic keywords. Their colors correspond to the relevant communities as shown in Fig. 2.12.

Robust clusters are depicted on Fig. 2.13. These robust clusters can be considered as sub-specialities of the identified disciplines listed in Tab. B.5. For example, the community identified by neuroscience: biology psychology is composed of several clusters, which are also characterized by research topics or theoretical areas. Note that, the study in neuroplasticity supports the treatments of brain damage, long-term potentiation concerns learning and memory, pre-botzinger complex is essential for respiratory rhythm,

and the activities in prefrontal cortex are considered to be orchestration of thoughts and actions in accordance with internal goals. All these topics and fields refer to the study in neuroscience and biological psychology. It reveals that fuzzy detection can extract communities in hierarchical organization.

In terms of modular overlaps, our results are shown in Tab. B.4. Except astronomy-ISM(Interstellar medium) which acts like a unstable cluster, the rest has a good agreement compared to the reality: discrete-event systems and multi-agents are very common for modeling and analyzing general systems, computational complexity is a common property of complex systems, and genetic expression [59, 79] studies are often used to determine whether a genetic variant is associated with a disease or trait.

Granular and/or Modular Overlaps. Comparing the results of granular overlaps and modular overlaps is interesting since it reveals their intrinsic differences. For instance, fuzzy detection considers three modular overlaps related to computer science: communication systems and ecosystems simultaneously, while clique optimization does not provide any result. We can also observe their similarity. For example, both results use visual cortex to characterize the overlapping nodes shared by neural networks and neuroscience: biological psychology. It mainly indicates that, for some cases, the two types of overlapping nodes can reach an agreement in characterizing overlaps.

Obviously, we can not compare and rank the two methods in a definitive and quantitative way. Granular overlaps and modular overlaps represent results based on different definitions. To the best of our knowledge, both definitions seem really reasonable to use since they are more complementary by their intrinsic uncovering structure. Finally, we conclude that both methods: clique optimization and fuzzy detection, are useful to identify overlaps in complex networks and to give insights on the complex structure of real networks.

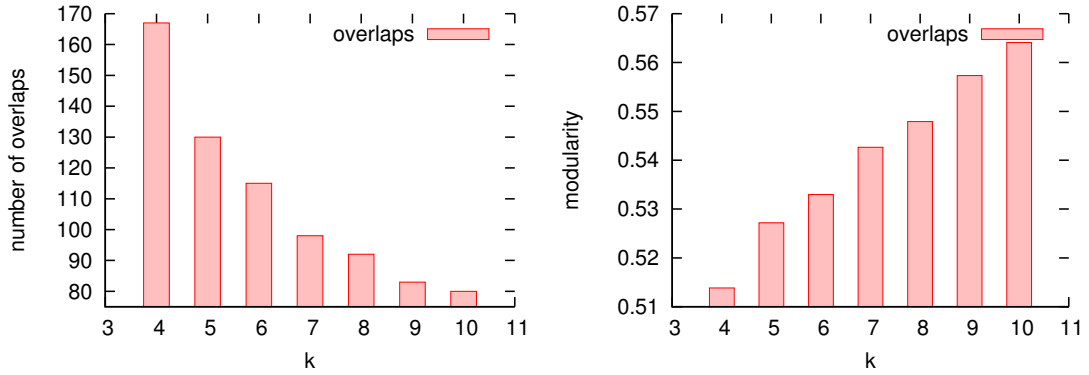
2.6 Discussion

In this section, we discuss the value of parameter used in our methods. We first present two networks used in our discussion.

Geography collaboration Geography collaboration is a co-author network combined with NUTS (The Nomenclature of Territorial Units for Statistics or Nomenclature of Units for Territorial Statistics)[7]. Nodes represent geo-codes, which are subdivisions of countries. Nodes are connected if there exists the collaboration between regions in scientific publications.

Wikipedia vote network Wikipedia is a free encyclopedia written collaboratively by volunteers around the world. A small part of Wikipedia contributors are administrators,

[7]NUTS is a geocode standard for referencing the subdivisions of countries, which is based on the existing national administrative subdivisions.



(a) number of overlaps

(b) modularity Eq. 2.9

Figure 2.14: Relative performances of clique optimization for the Geography Collaboration network when $k = [4, 10]$. The number of overlaps decreases 2.14 (a) as k is increasing. The modularity value increases 2.14 (b) as k is increasing. We notice that the modularity of the community structure containing overlapping nodes is less than the partition whose modularity is 0.620506.

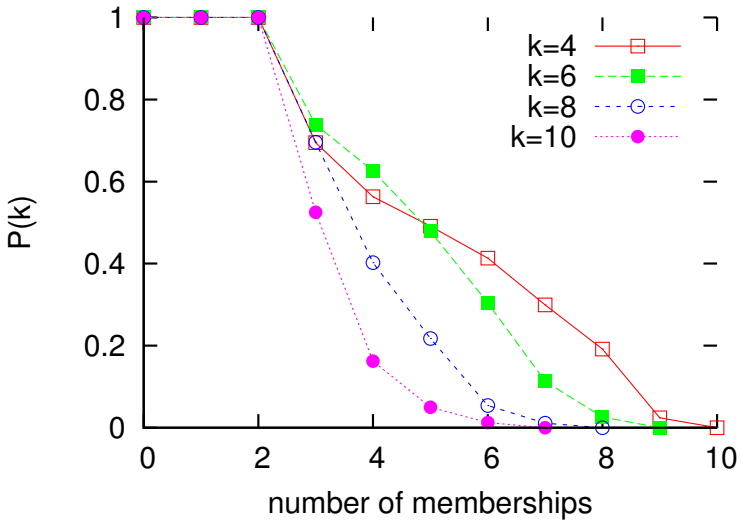
who are users with access to additional technical features that aid in maintenance. In order for a user to become an administrator a Request for adminship (RfA) is issued and the Wikipedia community via a public discussion or a vote decides who to promote to adminship. Using the dump of Wikipedia page edit history, $2,794$ elections with $103,663$ total votes and $7,066$ users participating in the elections (either casting a vote or being voted on) are extracted. About half of the votes in the dataset are by existing admins, while the other half comes from ordinary Wikipedia users [8].

2.6.1 Granular overlaps and the parameter k

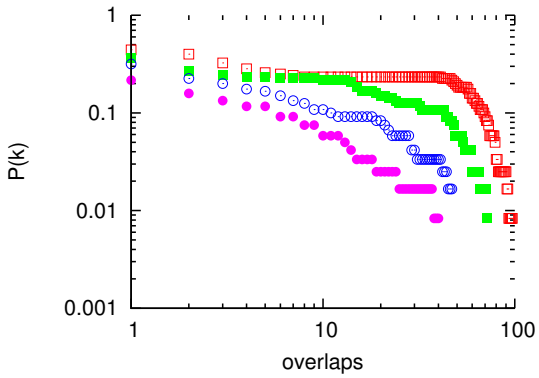
In clique optimization, the parameter k is used to prune nodes that are not overlapping nodes. If k increases, the number of granular overlapping nodes becomes smaller, but they are also more cohesive to the relevant communities. By applying clique optimization to real networks, we discuss the impacts of the parameter k.

We apply our clique optimization to the geography collaboration network. The average weight of the network edges $\omega^* = 0.00402245$ is used to prune the weak links. We compare the relative performances in detecting weak granular overlapping nodes for different values of k. In Fig. 2.14 (a) the number of overlapping nodes decreases as the value of k increases. The modularity of overlapping community structure is less than the modularity of the partition $\mathcal{P}_{\text{opt}}$ ($Q_{\mathcal{P}_{\text{opt}}} = 0.620506$), see Fig. 2.14 (b). Next we compare the relative distributions for overlapping nodes in weak sense obtained for different values of k (See Fig. 2.15). In Fig. 2.15 (a) there are differences in the membership number of overlapping nodes for different values of k. For $k = 4$, the maximum membership number

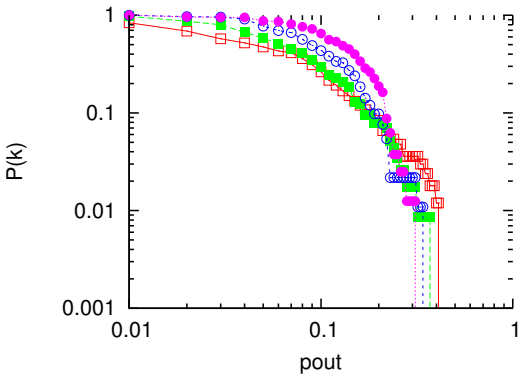
[8] http://snap.stanford.edu/data/wiki-Vote.html



(a) CDF of the membership number of overlapping nodes



(b) CDF of the overlap size



(c) CDF of the p_{out}

Figure 2.15: Statistics of clique optimization for Geography Collaboration at $k = 4, 6, 8, 10$. (a) the cumulative distribution function of the membership number of overlapping nodes (b) the cumulative distribution function of the overlap size, (c) the cumulative distribution function of p_{out} which is the portion of the sum weights on external degrees to the sum of the weights on the total degrees for a overlapping node.

of overlapping nodes is 9, but it is 6 for $k = 10$. We also note the small difference for the membership number distribution for overlapping nodes between $k = 4$ and $k = 6$: only 26% overlapping nodes have the membership number om ≤ 3 for $k = 4$ while it is 30% for $k = 4$ however due to nearly 51% overlapping nodes have the membership number om ≤ 5 for $k = 4$ while it is 50% for $k = 6$. It reveals that the overlapping nodes obtained at $k = 4$ are easier to have the membership number om $= 4$. Figure 2.15 (b) shows the overlap size distributions for pairs of communities found in the network. We observe that most pairs of communities share 1 or 0 overlapping nodes: 56% for $k = 4$ and 79%

for $k = 10$. It is a very good agreement between the relevant statistic distributions that overlapping nodes are not very common. Finally the distributions of the portion of the sum weights on external degrees are shown in Fig. 2.15 (c). we note all overlapping nodes having $p_{out} < 50\%$.

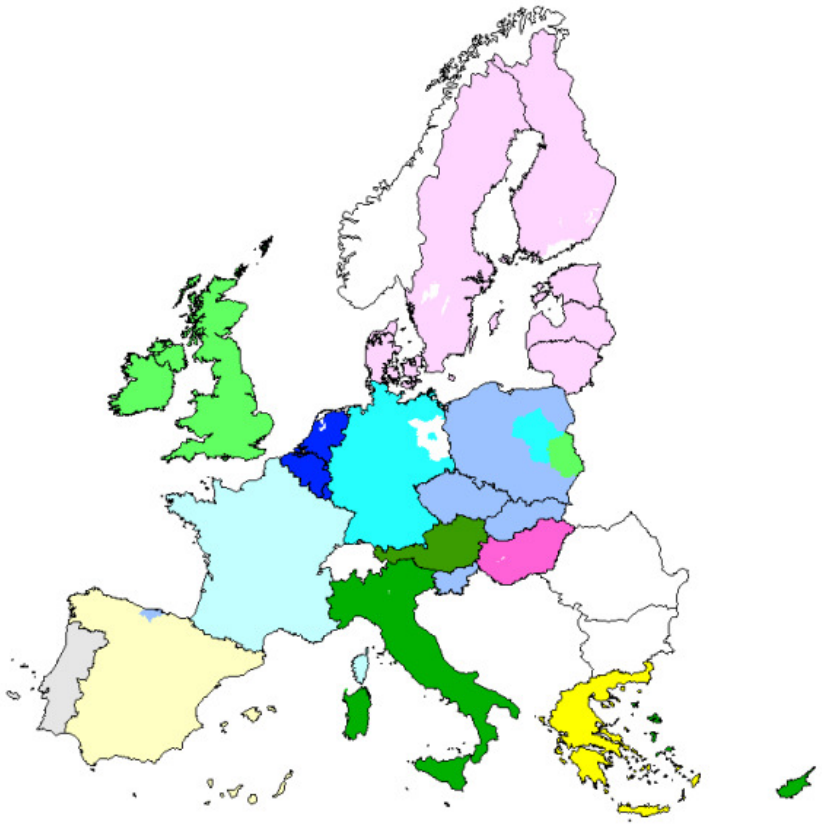


Figure 2.16: Community structure of Geography Collaboration. The figure illustrates communities by different colors. We find the found communities corresponding to countries, where each country can be identified by its geographic boundaries.

We also study the results in characterizing overlapping nodes. The community structure is displayed on the map of EU Countries (See Fig. 2.16): each color represents a computed community. We observe that the community structure corresponds to countries. For example, the regions belonging to United Kingdom form a light green community. The organization of the geography collaboration network into different countries indicates that the collaborations within the same countries are much more important (from a quantitative point of view) than international collaborations.

Our clique optimization detects a large number of overlapping nodes at $k = 4$. Some "*popular*" overlapping nodes can even be assigned up to 8 communities. The results are shown in Tab. 2.2. We observe that they refer to large cities or regions have well

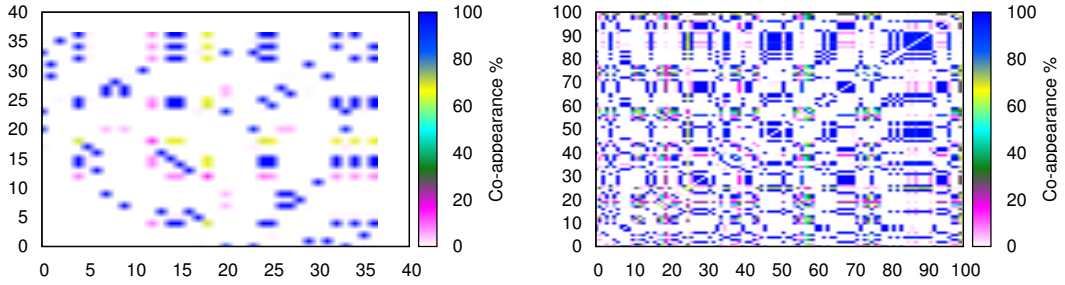
established and famous universities. For example in UKJ14($Oxfordshire$), there is the University of Oxford. It reveals that our results are well matched to the reality since these large cities or regions having famous universities play important roles in international collaborations.

By comparing to the results obtained at $k = 5$ or $k = 6$, we observe their high similarity. We define *popular overlapping nodes* as nodes having the maximum number of memberships at different values of k. We found 18 popular overlapping nodes for $k = 5$ and 12 for $k = 6$ that are part of popular overlapping nodes for $k = 4$. The total number of popular overlapping nodes (om = 8) are 27 for $k = 4$, the total number (om = 7) are 18 for $k = 5$ and the number (om = 6) are 12 for $k = 6$. Of course, the popular overlapping nodes are also large cities or regions that have famous universities and international collaborations (See Tab. 2.2). It tends to show that our method can characterize the fundamental properties of overlapping nodes among communities, which are independent of the value of k.

Next we study the impact of k on another dataset: Complex System Science (See details in Section 2.5). The results are listed in Tab. B.2 and Tab. B.3. It can be seen that the obtained granular overlaps have the same highest frequent topic keywords and very similar high frequent topic keywords at $k = 5$ or $k = 6$. For example, high frequent topic keywords owned by overlaps between ecosystems and chaos theory obtained for $k = 6$ totally match to the results obtained for $k = 5$: Dynamics, Self-Organization, Model, Complexity, Chaos, Systems, Stability, Patterns where "Evolution" and "Celluar Automata" are not shown in Tab. B.3 as their frequency are not high enough for $k = 6$. Totally, we obtained the same characteristics by mining these granular overlaps at different values of k. In other words, the fundamental properties of the found granular overlaps are largely independent of k and represent the characteristics owned by the system itself.

Node	Location	Node	Location
DE122	Karlsruhe, Stadtkreis	ES300	Madrid
ES523	Valencia	FR101	Paris
FR716	Rhône	ITC11	Torino
ITC45	Milano	ITD36	Padova
ITD55	Bologna	NL326	Amsterdam
SE010	Stockholm County	SE044	Skane County
UKD52	Liverpool	UKJ14	Oxfordshire

Table 2.2: Part results of popular overlapping nodes which are shared by 8 communities for $k = 4$ (which are still popular for $k = 5$). These popular overlapping nodes are related to big cities or regions having famous universities.



(a) Geographical Collaboration

(b) Wikipedia vote network

Figure 2.17: Performances of fuzzy detecting in networks for p_{c_j,c_i} between pairs of robust clusters, where robust clusters are sorted by their size in ascending order.

2.6.2 Community memberships and membership degree

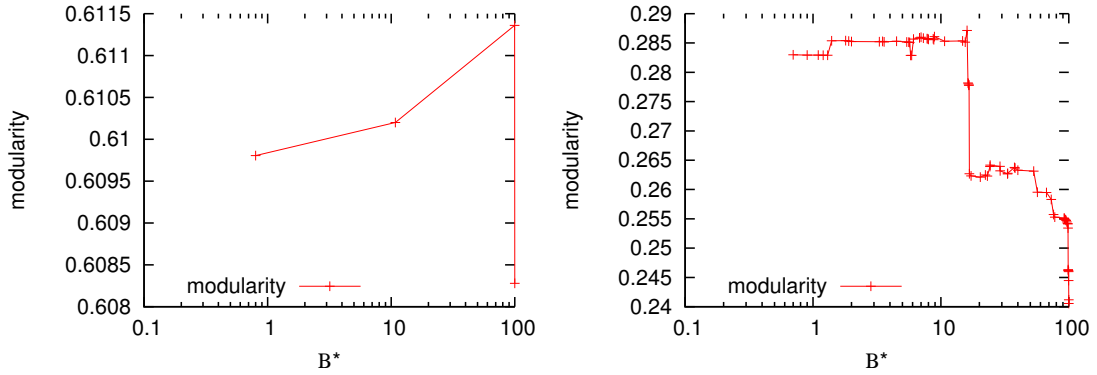
Our fuzzy detection typically sets $\beta^* = 0.1$ to determine community memberships. If the threshold β^* increased, the number of modular overlaps decreased; otherwise, more robust clusters are identified as modular overlaps. The criterion we used to fix the optimal β^* values should be based on finding a community structure having the good quality.

We studied the membership degree, which is used to determine community memberships. In Fig. 2.17, we show results by applying fuzzy detection to Geographical Collaboration and Wikipedia vote network. The figures show the obtained p_{c_i,c_j} for pairs of robust clusters, where the nodes are listed through their size in ascending order. From the results, we observe that most p_{c_j,c_i} are in values of approximate 99.9% (dark blue) and a few of p_{c_j,c_i} are in values of nearly 10% (light pink). It seems that robust clusters which perform unstable (belonging to different communities in the partition examples) have the low membership degree with the relevant community. If we set β^* in a high value, we would find no modular overlap.

In Fig. 2.18, we show the modularity by increasing the value of β^*. These results are obtained by applying fuzzy detection in Geographical Collaboration and Wikipedia vote network. We observe some critical points, which are important for the modularity like $\beta^* = 9\%$ in Fig. 2.18 (a) and $\beta^* = 18\%$ in Fig. 2.18 (b). In practice, we use the value corresponding to the critical point to set β^*, which is approximate 10%.

2.7 Conclusion

In this chapter, we have presented our studies in overlapping community structure. We have discussed the limits of existing modularity for qualifying the covers, and proposed a new extension, which is based on the Hamiltonian. We have also introduced two novel



(a) Geographical Collaboration (b) Wikipedia vote network

Figure 2.18: Performances of fuzzy detection in Geographical Collaboration and Wikipedia vote network, where the value of the modularity corresponds to the community structure obtained by the relevant β^*. Several critical points for modularity are observed.

methods to identify overlapping nodes. One is called clique optimization for identifying granular overlaps, and the other is named fuzzy detection for modular overlaps. Both methods have been tested successfully in synthetic graphs. Moreover, studies and the analysis on large networks like the Complex System Science one give good results and useful insights on the structure of the network.

We believe that the elements presented in this chapter can be of great help in the analysis of networks. On the one hand, the definition of granular overlaps and modular overlaps provide different insights in characterizing overlapping nodes for network analysis. On the other hand, the introduction of clique optimization and fuzzy detection could open the way for applications to large-scale systems. Remind the results in studying Yeast protein complexes in Section 2.3.3. The low sensitivity of clique optimization is caused by our definition of k-granular overlapping nodes, *i.e.*, not all real overlapping nodes participate in k-cliques. However, fuzzy detection provides results with a high sensitivity. Since fuzzy detection assigns nodes into communities without computing their connections. Simultaneously, clique optimization will not misclassify unstable nodes. Therefore, it has a higher specificity value than fuzzy detection. It suggests us to combine both methods to study overlapping community structure. We may obtain the complementary results.

Chapter 3

Overlapping communities and community evolution

In dynamic networks, change is a fundamental ingredient of interaction patterns in biology, technology, economy, and science: interactions within and between organisms change; transportation patterns by air, land, and sea all change; the global financial flow changes; and the frontiers of scientific research change.

Network clustering methods have become important tools to detect community evolution. Most methods make endeavours to distinguish between real trends and noisy data. However, detecting community dynamics is also important to study community evolution. For example: How has the network of global air traffic changed over the past half century? How does the organization of social contacts change when diseases develop and spread? How does the network structure of the federal funds market change when credit markets freeze up? How do gene regulatory networks differ between cancer and non-cancer states? And how does science evolve as research tools, strategies, and agendas shift through time?

Asur *et al.* [3] have detected clusters at each snapshot graph independently and used event definitions (Def. 2) to compute and identify community dynamics. Their studies in the DBLP co-authorship network showed how semantic content and category hierarchy information were related to community fusion or split. However, their event definitions need the parameter value. Furthermore, we do not have a good visualization tool to illustrate these diverse community dynamics.

In this chapter, we contribute to community detection in dynamic networks (Section 3.1) including a matching technique and a visualization tool. Our matching technique is able to resolve the problem of characterizing community dynamics. Our visualization tool makes community dynamics observable. We validate our method by applying it to a synthetic dataset and a blog network in Section 3.2. We also analyze a dynamic co-citation network called the past history of complex system science in Section 3.3 with the discussion of modular overlaps.

3.1 Tracking community evolution in dynamic networks

In the context of dynamic graphs, the interactions between nodes change over time. As the community is defined as a set of nodes having dense internal connections and sparse external connections, the changes of interactions can cause the evolution of communities in networks.

In the early analysis of community structure in dynamic networks, Palla *et al.* [97] have already introduced six basic scenarios in the evolution of communities: birth, growth, contraction, merging, splitting and death as we mentioned in Fig. 1.2. Recently, some studies [82, 101] have discussed the reason why a community structure may change. Their reasons can be divided into two categories: *a*) internal and *b*) external influences.

- Internal influences

  Common ground. It is the "mutual knowledge, mutual beliefs, and mutual suppositions" shared by individuals [24]. It attracts interactions between individuals. However, the common ground may change. For instance, increasing community size may increase fringe nodes (or unstable nodes). It increases the dissimilarity among members. Therefore, it can be observed that a rapidly growing community looses its common ground, and then changes.

  Community membership. A community may have relatively permanent members (or core nodes), but also has many *fluid members* (who join in communities occasionally and change their community memberships after several time steps). In addition, there are new members joining in communities, which also influences community dynamics.

- External influences: The external influences are various: a specie community changes by following food seasons [22]; a political community might be more active in the run-up to elections; and a science field community can disappear caused by a replaced criteria such as AIDS-related complex[1].

The influences described above result in the change of a community: creation, growth, shrunken, fusion, split or disappearance.

If one community is not involved in fusion events or split events but only increases its size or decreases its size, we say that the community *survives*. In [97], it turns out that the age of a community is positively correlated with its size, *i.e.*, the older communities are also larger (on average). In an organizational context, a community is created or survives if it maintains a coherence, *i.e.*, common ground. The investigation of a survival community allows us to learn how common ground evolves over time.

For the fusion of communities, Gongla and Rizzuto [49] have given two definitions: *a*) fusion between equal communities and *b*) fusion between unequal communities.

[1] AIDS-related complex was widely discontinued by the year 2000 in the United States after having been replaced by modern laboratory criteria

Fusion between equal communities. If communities share a lot of common members or properties, *i.e.*, interests, they may merge and are replaced by a new larger community.

Fusion between unequal communities. If a community constitutes a specialized sub-domain of a larger community it may willingly join in the broader community or be absorbed.

In the event of community split or community disappearance, Gongla and Rizzuto [49] have identified three factors:

Organizational change. A community is usually sponsored by a particular interest such as a topic, a function or a group of leaders. When these interests change, the community is at risk of changing such as dying. For example, a change of the group leaders may result in new priorities and redeployment of resources [49].

Knowledge domain change. A knowledge domain of a community is not necessarily static. An interest may change, and the community evolves. For example, in a co-citation network, the cooperation between communities can innovate a new research topic.

Community leadership change. The leaders (or the core members) of a community have a high influence on the evolution of the community. They can be more active and attract more new members. They also can make the community less active and finally disappear.

A disappearing community usually becomes gradually smaller or attracts less and less new members: community members have less and less interactions until the community vanishes.

As shown above, communities changes may caused by diverse factors. Their dynamic behaviors make the problem of tracking community evolution more complex. However, in analyzing networks, the properties of community persistence and community development should not be ignored. These properties can be important as they reveal the evolving tendencies of networks. The evolution of communities may be significant (Section 1.2.2). For example, a unique new created community is enough to change the whole community organization. Capturing when community dynamics occur and charactering these dynamics is an important aspect when investigating communities over time.

3.1.1 Group persistence two-stage method

We describe a novel heuristic to track community evolution in dynamic network. Given a dynamic graph $\mathcal{G}$, its community structure is a set of communities $\{\mathcal{C}_1, \ldots, \mathcal{C}_{n_c}\}$ which evolve over time. A given community $\mathcal{C}_i$ can be observed at several time steps. Of course, It also may change.

To resolve the problem of community detection in dynamic networks, we apply a two-stage approach which is briefly described as: in the first step, we use the fuzzy detection

algorithm described in the previous chapter to detect a partition with the maximum modularity, robust clusters and modular overlaps at each time step (Section 2.4); in the second step, we use a mapping method (Section 3.1.4) to connect partitions at different time steps. Simultaneously, we track community evolution and identify community dynamics. For tracking community evolution, we use an evolution path (Def. 1) to describe how one community evolves over time. The length of the evolution path denotes the duration time of the community. The variation of community members shows how one community attracts new members or loses old members. The connection between communities at different time steps is used to identify communities changes: creation, continuation, fusion, split or disappearance.

Although as described in Section 1.3.2, diverse mapping methods [42, 58, 117] are proposed and are used to identify communities at different time steps, the limit remains. For example, in terms of matching metric, a problem is whether the similarity value is based on $\min(|X|, |Y|)$ or $|X \cup Y|$, where X, Y represent the temporal clusters at different time steps. Moreover, definitions used for characterizing communities dynamics need parameter value such as the parameter κ in Def. 2.

To overcome the matching problem, we apply *group persistence* to match communities. Our method is motivated by [114] and connects communities depending on *overlap size*, *i.e.*, $|C(t) \cap C(t+1)|$, for two temporal clusters at time steps $t, t+1$ respectively. Next, we describe our method in details.

3.1.2 Motivation

Let take an example illustrating the notion of *group persistence* [114]. A group of five members $\{a, b, c, d, e\}$, is strongly related to a subsequent group with $\{m, b, c, d, e\}$ members. It is also clear that a group with $\{a, b, c, d, e\}$ is not related to a subsequent group of $\{f, g, h, i, j\}$. Through the knowledge of dynamic network analysis, one can infer that the properties of a group are not a summary of the properties of individual members. Instead, they emerge from the structure of interactions among members. Therefore, given a group $\{a, b, c, d, e\}$, one supposes that the development takes the following course: $a, b, c, d, e \longrightarrow m, b, c, d, e \longrightarrow m, n, c, d, e \longrightarrow m, n, o, d, e \longrightarrow m, n, o, p, e \longrightarrow m, n, o, p, q$. In this case, each stage is differentiated from the previous stage by only one member, and at each moment it shares the same majority elements with its neighboring moments. Consequently, the group with $\{a, b, c, d, e\}$ is defined to be linked with its subsequent group of $\{m, n, o, p, q\}$.

Motivated by this simple toy example, we use group persistence to track community evolution. First, we establish a relationship called *community predecessor and successor* between temporal clusters for every pair of contiguous stages. For two temporal clusters in predecessor/successor relationship, their overlap size must exceed a threshold γ^*. Then, we use this community predecessor and successor relationship to map temporal clusters and identify community dynamics.

Definition 5 (Community predecessor and successor)**.** *Given a temporal cluster $C_i(t)$ at time t, if the temporal cluster $C_j(t-1)$ has the maximum overlap size among all*

temporal clusters at time $t-1$, we define that $C_j(t-1)$ is the predecessor of $C_i(t)$. If the temporal cluster $C_k(t+1)$ has the maximum overlap size among all temporal clusters at time $t+1$, we define that $C_k(t+1)$ is the successor of $C_i(t)$.

In the following, given a pair of temporal clusters (X, Y), we use $X \to Y$ to denote that Y is X's successor and $X \leftarrow Y$ to represent that X is Y's predecessor.

When a community changes, the predecessor/successor relationship may be obtained by nodes which participate temporally in this community. It can be observed that some nodes may easily change their memberships. The threshold γ^* is used to handle this problem by filtering relationships caused by member fluctuation. This ensures that linked temporal clusters have high correlation when communities evolve over time, *i.e.*, for one community, there are few changes in its community members over time.

Remark. The relationship between one community and its successor (or its predecessor) may be asymmetrical. That is, for one community and its successor, this community may be not the predecessor of its successor. Similarly, for one community and its predecessor, it is possible that the community is not the successor of its predecessor. This asymmetrical property allows us to characterize community dynamics.

3.1.3 Community dynamics

As listed above, there are six basic community dynamics: a community emerges (creation), a community may grow (growth), a community can shrink (shrunken), several communities can merge together (fusion), a community can be split into several communities (split) or a community may disappear (disappearance).

The above definition of community predecessor/successor relationship allows us to characterize community dynamics.

Definition 6. *Let $G(t)$ and $G(t+1)$ be snapshots of $\mathcal{G}$ at two consecutive time steps with the temporal partition $\mathcal{P}(t)$ and $\mathcal{P}(t+1)$ denoting the community structure of $\mathcal{G}$ at time step t and $t+1$, respectively.*

Survive. *$C_j(t+1)$ is the continuation of $C_i(t)$, if and only if $C_i(t)$ is the predecessor of $C_j(t+1)$ and $C_j(t+1)$ is the successor of $C_i(t)$, such that:*

$$C_i(t) \leftarrow C_j(t+1) \wedge C_i(t) \to C_j(t+1)$$

This relationship is denoted by $C_i(t) \leftrightarrows C_j(t+1)$. We say that community $\mathcal{C}_i$ whose observation at time step t is $C_i(t)$, survives at time step $t+1$. The relationship between a temporal cluster and its continuation is symmetrical, such that, given a continuation, it must be the successor of its predecessor. If a community survives at the current time step, we identify whether it is a growing community or whether it is a shrinking community through the variance in size between its observation at previous time step and its current observation. We say that, a growing community has an increasing number of community members and a shrinking community has a decreasing number of community members.

Emerge. *$C_j(t+1)$ is a creation if and only if $C_j(t+1)$ has no predecessor such that:*

$$\nexists C_i(t) \in \mathcal{P}(t) | \, (C_i(t) \rightarrow C_j(t+1))$$

We say that a new community emerges if and only if its first observation has no predecessor.

Merge. *$C_j(t+1)$ is a fusion if and only if $C_j(t+1)$ is the successors of several clusters at time step t such that:*

$$\exists \{C_i(t), C_k(t)\} \subseteq \mathcal{P}(t) | \, (C_i(t) \rightarrow C_j(t+1) \wedge C_k(t) \rightarrow C_j(t+1))$$

where $i \neq k$. In case of $C_i(t) \rightarrow C_j(t+1)$ and $C_i(t) \nleftarrow C_j(t+1)$, we say that, community $\mathcal{C}_i$ is merged into $\mathcal{C}_j$ where $C_i(t)$ is the observation of $\mathcal{C}_i$ at time step t and $C_j(t+1)$ is the observation of $\mathcal{C}_j$ at time step $t+1$.

Split. *$C_j(t+1)$ is a split if and only if $C_j(t+1)$ is not the successor of its predecessor such that:*

$$C_i(t) \leftarrow C_j(t+1) \wedge C_i(t) \nrightarrow C_j(t+1)$$

We say that, a community is split from others if and only if its first observation is a split;

Disappear. *A community disappears at time $t+1$ if and only if its observation $C_i(t)$ at time step t has no successor such that:*

$$\nexists C_j(t+1) \in \mathcal{P}(t+1) | \, (C_i(t) \nrightarrow C_j(t+1))$$

Diagrams in Fig 3.1 show several cases illustrating community dynamics which can be featured by continuation, creation, disappearance, fusion and split. For better understanding community evolution, we show their evolution paths (Def. 1). For each community $\mathcal{C}$, its evolution path is $\text{Evol}(\mathcal{C}) := \{C(1), \ldots, C(\Delta)\}$, where each element $C(i)$ $(1 \leq i \leq \Delta)$ represents its observation at time step $t = i$. .

In the example illustrated by the Fig 3.1, we observe four communities, whose evolution paths are:

- $\text{Evol}(\mathcal{C}_1) := \{C_1(1), C_1(2), C_1(3), C_1(4)\}$,
- $\text{Evol}(\mathcal{C}_2) := \{C_2(2), C_2(3)\}$,
- $\text{Evol}(\mathcal{C}_3) := \{C_3(1), C_3(2), C_3(3)\}$,
- $\text{Evol}(\mathcal{C}_4) := \{C_4(3), C_4(4)\}$.

We can observe nearly all types of community changes:

- Community $\mathcal{C}_2$ is created at $t = 2$ as it has no predecessor at $t = 1$;

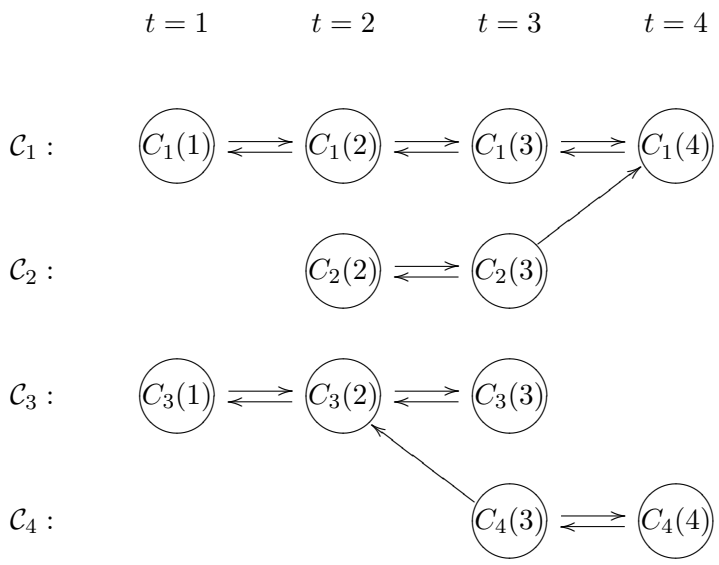


Figure 3.1: Diagrams of four communities observed during four time steps, featuring continuation, creation, disappearance, fusion and split.

- Community $\mathcal{C}_3$ disappears at $t = 4$ as it has no successor at $t = 4$;
- Community $\mathcal{C}_2$ is merged into $\mathcal{C}_1$ at $t = 4$ since its successor at $t = 4$ is $C_1(4)$ whose predecessor is not $C_2(3)$;
- Community $\mathcal{C}_4$ is split from $\mathcal{C}_2$ since $t = 2$ as its predecessor at $t = 2$ is $C_3(2)$ whose successor is not $C_4(3)$.

Community $\mathcal{C}_1$ is observable during all the observation window (only four time steps on this toy example). At time step $t = 4$, community $\mathcal{C}_2$ joins it. This community fusion event seems to be more an event related to $\mathcal{C}_2$ rather than to $\mathcal{C}_1$.

A more complex diagram is displayed in Fig. 3.2. We observe the changes of communities from time step $t = 2$ to $t = 3$. At time step $t = 3$, community $\mathcal{C}_2$ partially merges with $\mathcal{C}_3$ while its split $C_1(3)$ starts a new community $\mathcal{C}_1$.

The definition of community predecessor/successor relationship allows for linking communities at different time steps. It also makes the problem of characterizing community dynamics be captured easily.

3.1.4 Mapping method

Our framework uses fuzzy detection to detect community structure in each snapshot graph. The results include the optimal partitions in terms of modularity, a set of robust clusters, the community cores and the modular overlaps. Having a range of granularity and resolution in the results is an opportunity. We use a mapping method to track

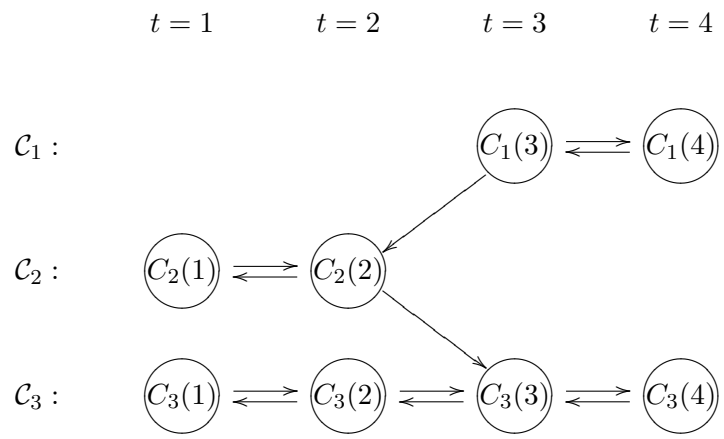


Figure 3.2: Diagram of four clusters observed during over 4 time steps, featuring fusion and split community events.

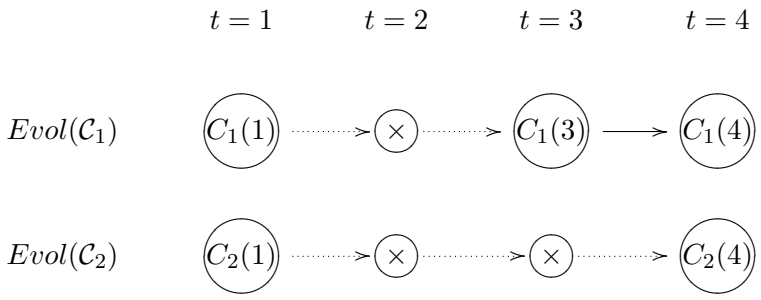


Figure 3.3: Community evolution paths of two evolutionary communities over four time points. During their evolution, they are unobserved at some time steps. For instance, community $\mathcal{C}_1$ is unobserved at $t = 2$ while it reappears at $t = 3$, and community $\mathcal{C}_2$ seems missing at $t = 2$ and $t = 3$.

community evolution and identify community dynamics. We are also able to use the same method to track robust cluster evolution. The results can help us to understand leadership change (community core members change) and influences of community members. Another advantage is to track the evolution of modular overlaps. In the context of science whose community structure is captured by different research fields, it provides insights in the evolution of interdisciplinary fields, which link several communities.

Before our description of our framework in details, we illustrate some special cases as depicted in Fig. 3.3 and introduce the definition of *reappearing community*. This Figure illustrates the case where some communities become unobservable at a time step but reappear after several time steps. For community $\mathcal{C}_1$ in Fig. 3.3, we say that its observation at time step $t = 2$ is an *invisibility*, *i.e.*, it occurs when one community is unobserved but reappears lately.

Our mapping method uses the definition of community predecessor and successor to track community evolution. Additionally, it uses a backward method to identify reappearing communities. The details of how we track community evolution is sketched in Algorithm 6.

We use the definition of community predecessor and successor to track community evolution: two temporal clusters are mapped if they share the relationship of community predecessor and successor. If a community becomes unobservable, we hold its last observation. Lately, we apply a backward method to identify it when it reappears.

For a community which is not the continuation, we use a *possible predecessor and successor relationship* to connect it with a disappeared community:

Definition 7 (Possible predecessor and successor)**.** *Given a temporal cluster $C_i(t)$ at time t which is not a continuation, it is a possible successor of a disappeared community $\mathcal{C}_j$, if the last observation of $\mathcal{C}_j$, i.e., $C_j(t-\Delta)$, shares the maximum overlap size among all temporal clusters at time t and the overlap size exceeds the size threshold γ^*. For a disappeared community $\mathcal{C}_j$ whose last observation is $C_j(t-\Delta)$, it is a possible predecessor of $C_i(t)$: if $C_i(t)$ has a predecessor $C_k(t-1)$, the overlap size between $C_i(t)$ and $C_j(t-\Delta)$ exceeds the overlap size between $C_i(t)$ and $C_k(t-1)$; otherwise, the overlap size between $C_i(t)$ and $C_j(t-\Delta)$ exceeds the size threshold γ^*.*

We connect a temporal cluster and a disappeared community if and only if the temporal cluster is a possible successor of the disappeared community and the disappeared community is the possible predecessor of this temporal cluster with the maximum overlap size among all disappeared communities. We use $C_j(t-\Delta) \leftrightsquigarrow C_i(t)$ to denote this relationship between a temporal cluster $C_i(t)$ and a disappeared community whose last observation is $C_j(t-\Delta)$. In this case, we say that a community reappears.

Our results are based on partitions. We do not consider overlapping communities for tracking community evolution since they make the problem more complex. For instance, when we establish the relationship between temporal clusters at different time steps, we count the total community members or only non-overlapping parts? When overlaps between a pair of communities become an independent community, this dynamic is classified into merge event or split event? When a community shares a lot of nodes with others, we may obtain a wrong successor for large overlapping nodes.

However, our fuzzy detection is able to provide modular overlaps. We can track modular overlaps to study how overlapping parts evolve. Therefore, we also use the same mapping method (Algo. 6) to track robust cluster evolution: two temporal robust clusters are mapped if they share a relationship of predecessor and successor. When tracking robust cluster evolution, we do not only study modular overlaps evolution but also investigate community cores evolution. The later is helpful to understand how leadership changes affect community evolution.

3.1.5 Visualizing community evolution

In this section we present our novel visualizing tool for revealing structural changes and illustrating "*stories*" in dynamic networks. We first review existing tools for visualizing

Algorithm 6 Method for tracking community evolution

Input: An evolving graph $\mathcal{G}(V, E)$, which consists of a sequence of snapshot graphs $\mathcal{G} = \{G(1), G(2), \ldots, G(\Delta)\}$ over Δ time steps.
Output: Dynamic communities $\{\mathcal{C}_x\}$.

```
D ← ∅, Evol(C) ← ∅
for all time steps t = 1 → Δ do
  Get a partition of communities P(t) = {C_1(t), ..., C_k(t)}
  // STEP 1: Mapping communities
  Map temporal clusters using C_k(t) ⇆ C_k(t + 1)
  Hold all disappeared communities: D ← D ∪ {C_i(t − 1), ...}
  // STEP 2: Feedback method for reappearing communities
  for all temple clusters which are not continuation C_k(t) do
    Find possible predecessors and successors
    if C_k(t − Δ) ⇜⇝ C_k(t) then
      Update D ← D \ C_k(t − Δ)
    end if
    Update all evolution paths Evol(C) ← Evol(C) ∪ C(t)
  end for
end for
```

community evolution in dynamic networks, and second describe our tool in details.

Visualizing dynamics in communities

In early work, several tools such as SoNIA [87] and TeCFlow visualize dynamic networks by creating graph movies, where nodes move as a function of changes in relations. However, these tools fail to indicate a changing behaviour of community memberships and community dynamics. In [88], matrix is used (Similar as Fig. 1.10), whose element represents the community membership of a node at a time step. Each node occupies a column. Colours are used to depict communities. We can observe how a node changes its community membership through the colour change in the corresponding column. The drawback is that we do not directly observe how one community emerges, merges, splits or disappears.

An example of a graph with dynamic communities is depicted in Fig. 3.4. The evolution path of a dynamic community is depicted by a diagram occupying a column. Each diagram represents a community as a block and show relationships between preceding and succeeding clusters through horizontally connected stream fields. This result is obtained by the algorithm of bootstrap [33] in [108]. It enables to show community dynamics. For example, we observe the orange module merges with the red module in Fig. 3.4. In addition, in this case, we are also able to observe the significance of clusters, which is shown by dark colour.

The tool of alluvial diagram seems good in displaying structural change in science, economics, and business. Next, we introduce our visualization tool which has the similar

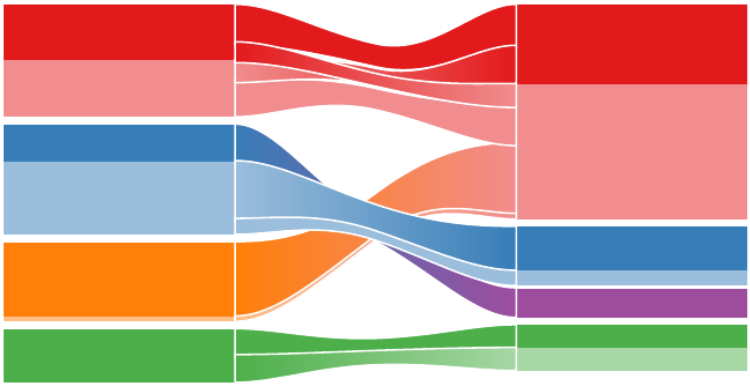


Figure 3.4: **Example of mapping between communities.** In the bottom networks, the darker colors represent nodes that are clustered together in at least 95% of the 1000 bootstrap networks. The alluvial diagram highlights and summarizes the structural changes between the time 1 and time 2 significance clusters. The height of each block represents the volume of flow through the cluster. The clusters are ordered from bottom to top by their size, with mutually nonsignificant clusters placed together and separated by a third of the standard spacing. The orange module merges with the red module, but the nodes are not clustered together in 95% of the bootstrap networks. The blue module splits, but the significant nodes in the blue and purple modules are clustered together in more than 5% of the bootstrap networks. The figure is obtained from [108].

good performance in depicting community evolution and showing community dynamics.

Visualizing community evolution through lineage diagrams

Our visualization tool illustrates "*stories*" in dynamic networks through *lineage diagrams* (See Fig. 3.5). Each lineage represents a separate evolutionary path, and occupies a column. The evolution of a community is shown from left to right. The temporal clusters representing the observations of the same community are shown in the same y-axis. Each cluster is shown by a circle whose size is proportional to its number of nodes. A lineage tie is added between two clusters if they share a successor or predecessor relationship. Therefore, if a circle has a link to another column, it indicates a community change. For example, in Fig. 3.5, we observe a link connecting a violet cluster and an orange cluster between $t = 2$ and $t = 3$. It represents a change event. We can characterize community dynamics through the orientation of links:

- If this link is oriented from left to right, it indicates that a community merges into another one;
- If this link is oriented from right to left, it indicates that a community is the result of a split from another one.

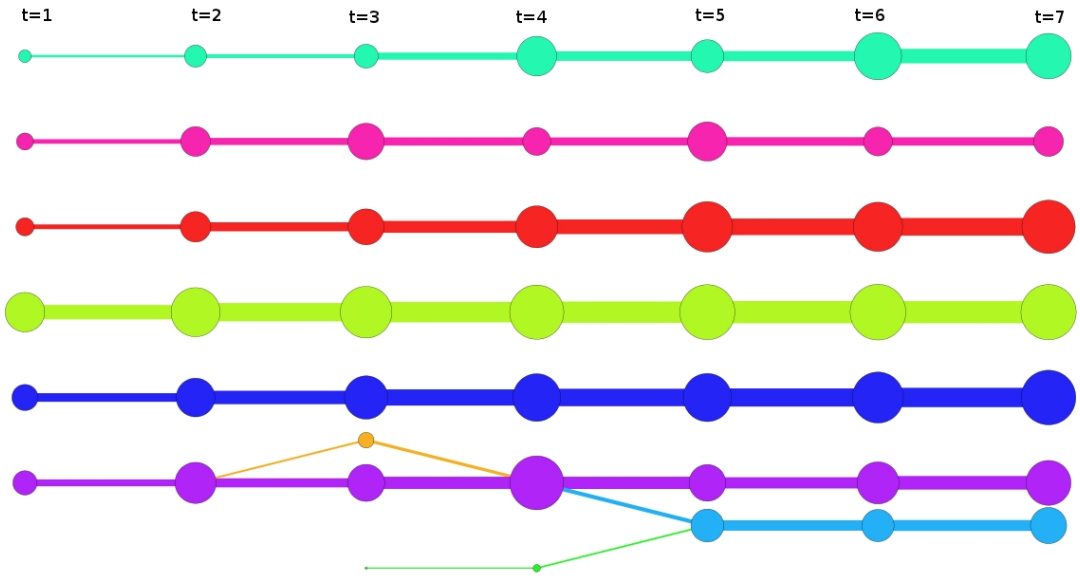


Figure 3.5: Applying our method to a sample dynamic network. Between $t = 2$ to $t = 3$, an orange cluster is split from the violet community. At $t = 3$, a new green cluster is emerged. Between $t = 3$ to $t = 4$, the orange cluster is merged into the violet community. Between $t = 4$ to $t = 5$, a blue cluster is split from the violet cluster and it is merged with the green community simultaneously.

The orientation of links is shown in colour such that the link colour is given by the link parent. For example, in Fig. 3.5, we observe a violet cluster having a link with blue colour which connects it and a blue cluster. In this case, we say that the blue cluster is a split of the violet community. Moreover, the blue cluster has a link with green colour which link it to a green cluster. We say that the green community merges into the blue community.

In addition, we use colours to indicate community memberships. The observations of a community at different time steps share the same colour.

In terms of robust clusters, we also use lineage diagrams to show structural changes. Each lineage represents a separate evolutionary path of a robust cluster, and occupies a column. The evolution of a cluster is shown from left to right. The temporal clusters representing the observations of the same clusters are shown in the same y-axis. Each cluster is shown by a circle whose size is proportional to its node number. Each lineage

tie is added between two clusters if they have the successor or processor relationship.

As colors correspond to community memberships, we can study community member shifts. For instance, some robust clusters may change their community memberships as the graph evolves.

3.2 Experimental results

Through our method, community dynamics like merge or split become easy to be identified. In case of reappearing communities, we validate our method through a set of synthetic networks. We also show performance of our method by applying it to a real dataset.

3.2.1 Synthetic datasets

Greene and Doyle [51] proposed a set of benchmarks based on Lancichinetti and Fortunato's technique [72]. Lancichinetti and Fortunato assumed that the distributions of degree and community size are power laws, with exponents τ_1 and τ_2, respectively. Each node shares a fraction $1 - \mu$ of its edges with the other nodes of its community and a fraction μ with the rest of the graph; μ is a mixing parameter in range of $[0, 1]$. After predefining community structure, edges are randomly assigned corresponding to node internal degrees and external degrees.

For the event of community reappearance, Green and Doyle has constructed a set of synthetic datasets, which covers $15,000$ nodes over 5 time steps. At each time step, 10% of communities are unobserved by randomly permuting node memberships (and edges).

By applying our method to this dataset with $l^* = 1$ over all time steps[2] and $\gamma^* = 5$ for matching communities[3] (Def. 5), we track community evolution and observe at least 40 reappearing communities at each time step.

To validate our method, we compare our results and the ground truth. We describe our results by *observed communities* and the ground truth by *expected communities*. The *true positive nodes* represent the nodes assigned in both observed reappeared communities and expected reappeared communities. Table 3.1 show our results in views of the number of observed reappeared communities (NOC) and number of expected reappeared communities (NEC), the number of true positive and the number of the reality (NPOCM/NECM), and the mean positive predictive value (mean PPV) with the standard error (SE).

From Tab. 3.1, we observe the similar number of reappeared communities obtained by our methods to the ground truth. For each observed reappeared community, it has a positive predictive value. It is the ratio of the number of true positive nodes in the observed community. Therefore, the high value of mean PPV represents that most nodes

[2]Studies have shown our method gives the partition with the highest NMI at $l^* = 1$.

[3]In [77], best communities are defined to have the size scale of between 10 to 100 nodes. This is also the size scale of communities in the synthetic networks. Setting $\gamma^* = 5$ is able to filter artificial clusters caused by degree variation and guarantee the matched small communities maintaining their most community members.

Time	NOC/NEC	Mean PPV	SE	NPOCM/NECM
t=3	48/50	0.95826	0.199826	1152/1198
t=4	45/48	1	0	1244/1283
t=5	41/47	1	0	1096/1155

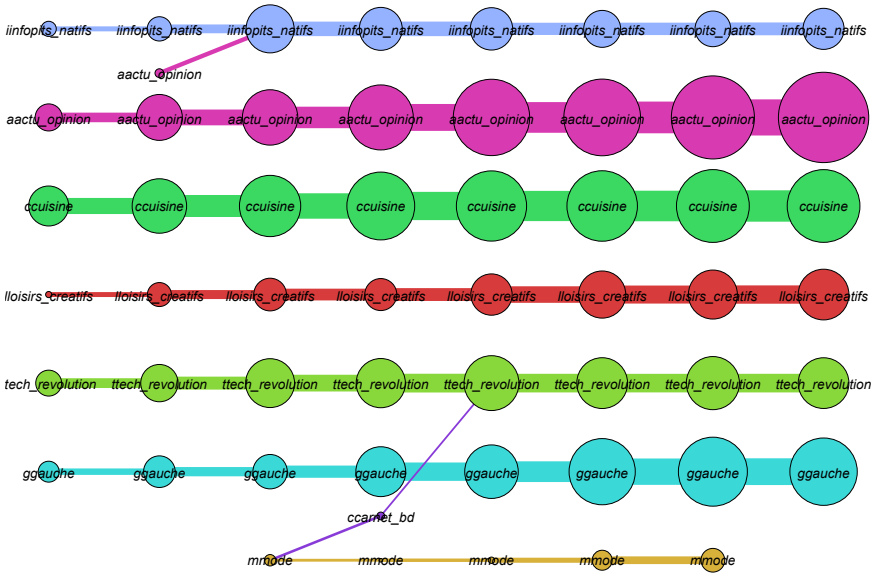
Table 3.1: Results of our method on hide dataset where 10% of communities are unobserved from time $t = 2$ onwards. In views of reappeared community number, *i.e.*, number of observed reappeared community (NOC) and number of expected reappeared community (NEC), our results are similar to the real ground truth. For instance at time $t = 3$, we observed 48 reappeared communities through our method, while the ground truth is that 50 communities reappear. In views of the mean positive predictive value (Mean PPV), our results gave a really high mean PPV value. Especially at $t = 4$ and $t = 5$, the mean PPV value is 1. It reveals that all nodes belonging to the observed reappeared communities totally match to the reality. The comparison between the number of true positive nodes and the number of the reality (NPOCM/NECM) shows that how many nodes belonging to the reappeared communities in the ground truth are found by our methods. As we can see, our framework has good performance in detecting reappeared communities.

in observed reappeared communities are positive truth nodes. Especially at $t = 4$ and $t = 5$, the mean PPV value is 1. It reveals that all nodes belonging to the observed reappeared communities totally match to the reality. The number of the reality (NECM) is the number of nodes belonging to the expected reappeared communities. Thus, the similar number of true positive nodes to NECM represents that most nodes belonging to the expected reappearing communities are found. As we can see, our framework has good performance in detecting reappeared communities.

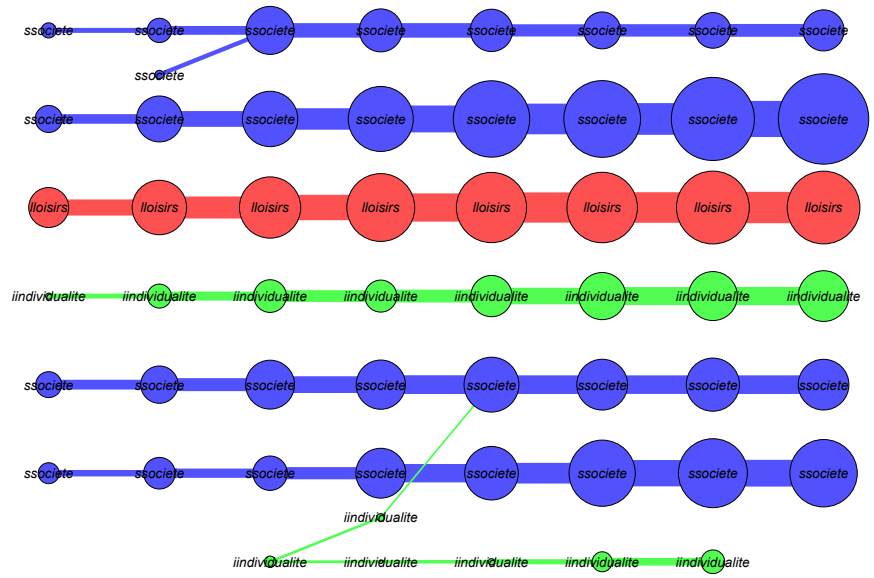
3.2.2 Blogs

Given a blogs network, approximately six thousand blogs were monitored to track the various articles and comments posted or the citation links between them for four months. We used the networks between blogs containing the aggregated data to the relevant day. we begin at the 1st day and then add each new day blogs and links between them. So we get a growing network consists 120 time steps.

By applying our method to blogs network, we show the results in Fig. 3.6. The node is labelled by the highest frequency class. When one post is added, its blog source and blog destination are classified into three levels. In the highest level, there are only three different classes (*iindividualite, ssociete, lloisirs*). We compute the frequency of classes by aggregating them. For example, once one blog is classified into the class of *iindividualite*, we aggregate the frequency of *iindividualite* in the community in which the blog is assigned. Finally, we select the class with the highest frequency to label the community.



(a) labelled by the class in the lowest level



(b) labelled by the class in the highest level

Figure 3.6: Applying our method to a blogs network.

For the visualization and comparison, the color of nodes corresponds to the label. For instance, the nodes at the bottom are coloured by green in Fig. 3.6 (b). It corresponds

to their labels: all nodes are labelled by *iindividualite*. Since there are 120 days, we set a time window $t = 2$ weeks (*i.e.*, 14 days), such that when $t = 1, G(1)$ is the aggregation of interactions between nodes during $[0, 14]$ days; when $t = 2, G(2)$ is the aggregation of interactions between nodes during $[0, 28]$ days; and so on. We set a size threshold $n^* = 100$ such that all shown temporal clusters contain at least 100 nodes. In this figure, the evolution is shown from left to right with x-axis denoting the time step and y-axis representing community index. Circles represent the found communities, with size proportional to community size. Links represent member continuity of at least 5 nodes, with size proportional to the continuity. The color of links corresponding to link parents denotes their orientation.

We compare the results labelled by classes in the lowest level and the highest level. We observe that nearly all communities hold their classes over all time steps. For example, we observe the community (3-th line) labelled by *ccuisine* in Fig. 3.6 (a) (or labelled by *lloisirs* in Fig. 3.6 (b)) survives from $t = 1$ until $t = 8$. In views of the highest level shown in Fig. 3.6 (b), we observe that most changes occur within the same classes. For example, the merge event at $t = 3$ occurs within the class *ssociete*. In views of the lowest level (See Fig. 3.6 (a)), more information is provided. For example, these observed communities (whose duration is at least two time steps) show the evolution of different classes. We observe a community labelled by *mmode* (in orange at the bottom) emerges at $t = 3$ (nearly 42 days). We also observe the community labelled by *mmode* has a split at $t = 3$. This split community is labelled by *ccarnet bd*, which merges into the community *ttech revolution* at $t = 5$. It may represent the close relation between *ccarnet bd* and *ttech revolution*.

These results are used to show that our method provides a good visualization tool: how communities change becomes easy to learn.

3.3 Application to a dynamic co-citation network

Finally, we apply our method to a dynamic co-citation network. It is called *past history of complex system science*. This data set (See also Section 2.5), collects extracted articles from the Institute for Scientific Information Web of knowledge[4]. All selected articles contain topic keywords relevant to the field of complex systems such as "complex*", "self organ*", "complex network*", "econophysics*", and so on.

Complex systems is a new approach to science that mathematically models behavior of systems, and builds relationship between system interacts and its environment. As early as 19th century, complex systems theory was used to capture economic computation problem. So far, it is used to model processes in computer science, biology, economics, physics, chemistry, and many other fields. The key problems of complex systems are modeling and simulating system behaviors. Various kinds of methods for identifying, exploring, designing and interacting with complex systems are used. In our early study of complex system science (Section 2.5), we obtain various claims to the

[4] `http://www.webofknowledge.com`

universality. The identified community structure provides a broader view of disciplines and methodologies using complex systems approach.

The past history of complex system science could be represented by a dynamic network. In the network, entities (articles) associated to their published time evolve over time. An intuitive way to capture the history of complex system science is to construct a sequence of snapshot graphs, whose community structure changes correspond to the science evolution. In the following, we first build a dynamic graph, and then detect, visualize and analyze community evolution and their dynamics.

Why use dataset about bibliographic coupling Citation analysis is the study of the frequency, patterns and graphs of citations in articles and books. It uses citations in scholar works to establish links to other works or other researchers. Today, there have various applications of citation analysis tools, which provide the understanding and analysis of information retrieval and science evolution. In the context of community organization of graphs, a citation network is associated with citation patterns, where each citation pattern corresponds to a scientific topic or research field. The evolution of community structure in a citation network reveals science history. For instance, HELLSTEN *et al.* [57] have used OPM (Optimal Percolation Method) to study the community structure of a citation network. The dataset is the ISI-indexed publication record of Werner Ebeling. The results showed that communities of this network corresponded to the author (Werner Ebeling)'s general contribution (*sequences, chaos, self-organization, systems*), a specific branch (*plasma research*) and collaboration contributions (with other authors).

Hopcroft *et al.* [58] used a network extracted from the NEC CiteSeer database [47] related to computer science, with a small collection covering other topics like physics, mathematics, and economics. In the result of their application, they observed the change of one community: the field of *quantum algorithms and communication* is emerged.

Applying a computational technique to a citation network becomes a popular method to analyse science history. In the later, we apply our method to a dynamic co-citation network. Before starting our dynamic studies, we review our investigation in a static co-citation network (See Section 2.5). Through its community structure, we observe:

Communities refer to research topics or theoretical fields. By characterizing community structure, we observe molecular biology, ecosystems, complex networks, dynamic turbulence these common research topics or theoretical fields in the science of complex systems but obviously refer to diverse disciplines.

Robust clusters can be considered as sub-specialities. Robust clusters have close relationship with their communities. That is, the relationship between robust clusters and their communities can be expressed by the relationship between subspecialities and specialities in views of cluster characterization. It reveals that robust clusters represent a possible hierarchical organization of communities.

Module overlaps link several topics and/or theoretical fields. Modular overlaps represent clusters of overlapping nodes, which link several topics and/or theoreti-

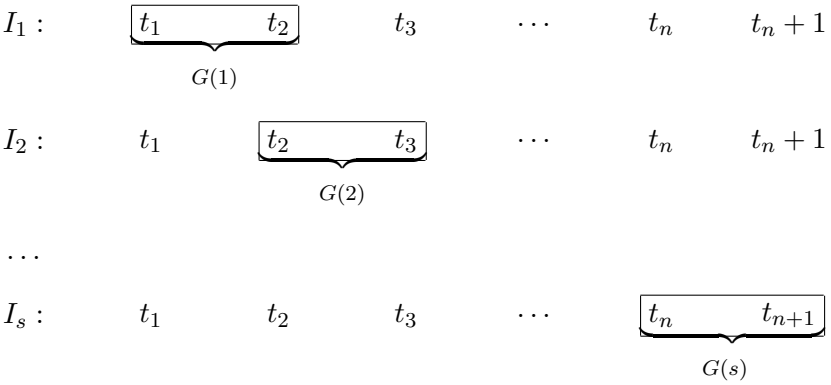


Figure 3.7: An example of sliding window with overlaps. The interval size is 2 with an overlap of 1.

cal fields in views of cluster characterization. For example, visual cortex is used to characterize the overlapping nodes shared by neural networks and neuroscience: biological psychology.

In case of dynamic co-citation networks, analysis in community evolution may provide us insights in understanding the past history of complex system science.

3.3.1 Building a dynamic graph

Articles in the past history of complex system science were published during 1985-2009. An edge $e = (i, j)$ connects two articles i and j if both articles share common references and the weight w_{ij} of the edge is given by the bibliographic coupling between i and j [115]. We note R_i the set of references cited by an article i. The bibliographic coupling between two articles i and j is $w_{ij} = \frac{|R_i \cap R_j|}{\sqrt{|R_i|\,|R_j|}}$.

We construct a dynamic graph according to each publishing year. Each snapshot graph captures interactions between nodes during a given time interval. Motivated by Palla *et al.* [97], we use a time *overlapping window*. It smooths out the gaps that sometimes occur between two discrete time intervals. We set the time interval size to 10 years with an overlapping time of 5 years. Then, we construct a dynamic graph on the obtained overlapping window (See Fig. 3.7). More details on the dynamic graph in the sequence of snapshot graphs are given in Tab. 3.2.

3.3.2 Detecting and visualizing community evolution

Let list the result factually. On the data set described above we obtained:

1985 − 1994: there are 14 communities[5];

[5]One community is not shown in the figures as its frequency of topic key words is really too small and does not allow us to characterize it.

Time period	Number of nodes	Number of edges	Total weight
1985-1994	20286	1004458	183594
1990-1999	62040	6179802	1.0569e+06
1995-2004	109458	12662556	2.1206e+06
2000-2009	141163	19603888	3.6701e+06

Table 3.2: Properties of the past history of Complex System Sciences.

1990 − 1999: the snapshot graph is still organized into 13 communities;

1995 − 2004: the snapshot graph is described by 16 communities;

2000 − 2009: we only observed 12 communities.

In terms of community dynamics, we observe:

between 1985 − 1994 and 1990 − 1999: 4 communities split and 3 merge ;

between 1990 − 1999 and 1995 − 2004: 3 communities split and 1 merges ;

between 1995 − 2004 and 2000 − 2009: 1 community splits and 3 merge.

The lineage diagrams are shown in Fig. 3.8. The Figure illustrates structural changes that occur in the past history of complex system science co-citation network over the years 1985 to 2009. We see the evolution of the number of communities and observe how important the split or merge events are to explain structural changes.

3.3.3 Evaluating the results

A key question remaining is how well our method is to track community evolution in our dataset. We choose to study the stability of communities. The stability measures the probability of community members to maintain their community memberships over time. Given a community $\mathcal{C}$ whose observation at time t is $C(t)$, its stability is the portion of active nodes at time $t+1$ that are assigned to its successor $C(t+1)$:

$$\text{stability}(\mathcal{C})(t) = \frac{|C(t) \cap C(t+1)|}{|C(t) \cap G(t+1)|} \tag{3.1}$$

where $C(t) \cap G(t+1)$ denotes the nodes belonging to $C(t)$ which are still active (or are recorded) at time $t+1$ and $C(t) \cap C(t+1)$ represents the nodes in common between $C(t)$ and $C(t+1)$.

Similarly, we also study the core stability, which is the portion of active core nodes $\widehat{c}(t)$ which are assigned to the successor $C(t+1)$:

$$\text{stability}(\widehat{c})(t) = \frac{|\widehat{c}(t) \cap C(t+1)|}{|\widehat{c}(t) \cap G(t+1)|} \tag{3.2}$$

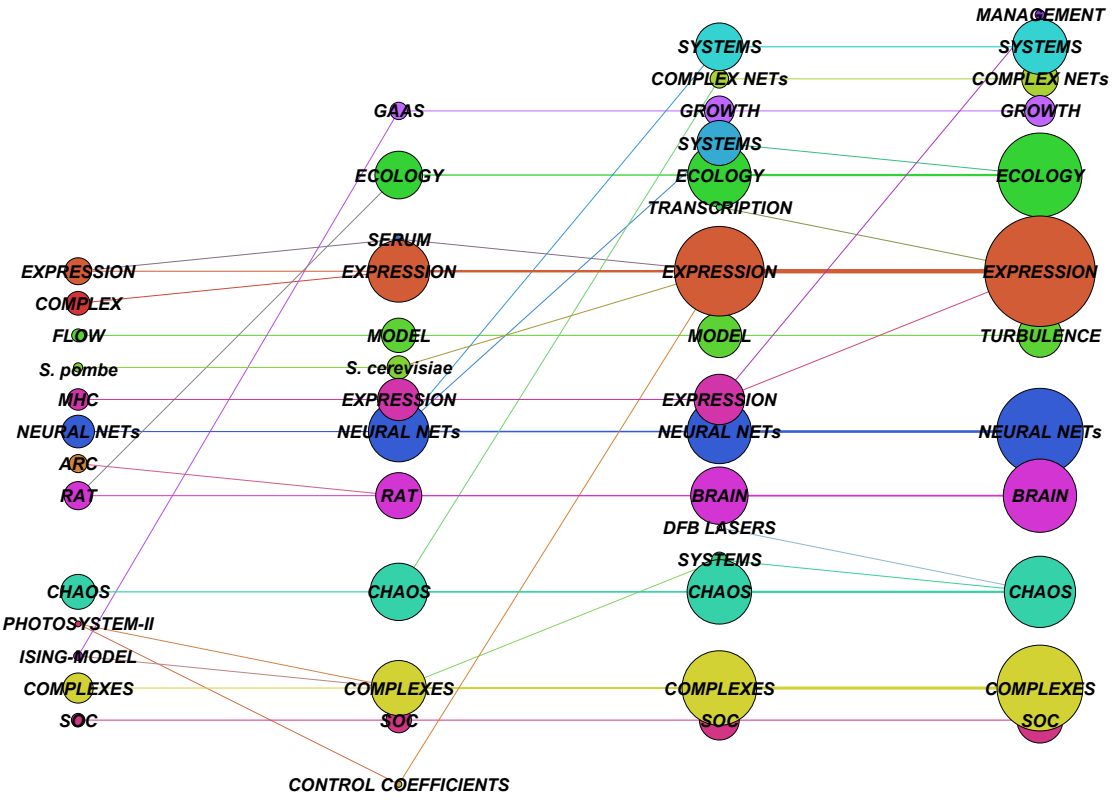


Figure 3.8: Results of our framework on the past history of complex system science network. Temporal clusters are marked by the most popular topic keywords. Their colors correspond to the relevant communities as shown in Fig. 2.12.

where $\widehat{c}(t) \cap G(t+1)$ denotes the core nodes of $C(t)$ which are still active (or are recorded) at time $t+1$.

As discussed in [58], the core nodes have high stability to hold their community memberships. Here, we compare the stability of core nodes and the disjoint community members to show how well our method is to track community evolution.

Table 3.3 gives the average stability value of clusters and core nodes between every pair of consecutive snapshot graphs. We observe that our results have a high agreement with the results of community identification through core nodes. Compared to general community members, the core nodes have higher probability to maintain their community memberships. For example, the first row in the table shows that for all communities with the size threshold[6] $n^* = 100$ during 1985-1994 (the total number of communities is 14), their average stability value is 0.740646 with a standard deviation of 0.152708. We also observe that most of core nodes during 1985-1994 appeared in their successor communities.

[6]For the visualization, we only show and analyse communities which have size above 100 nodes. In [77], the community which has size roughly 100 nodes is good.

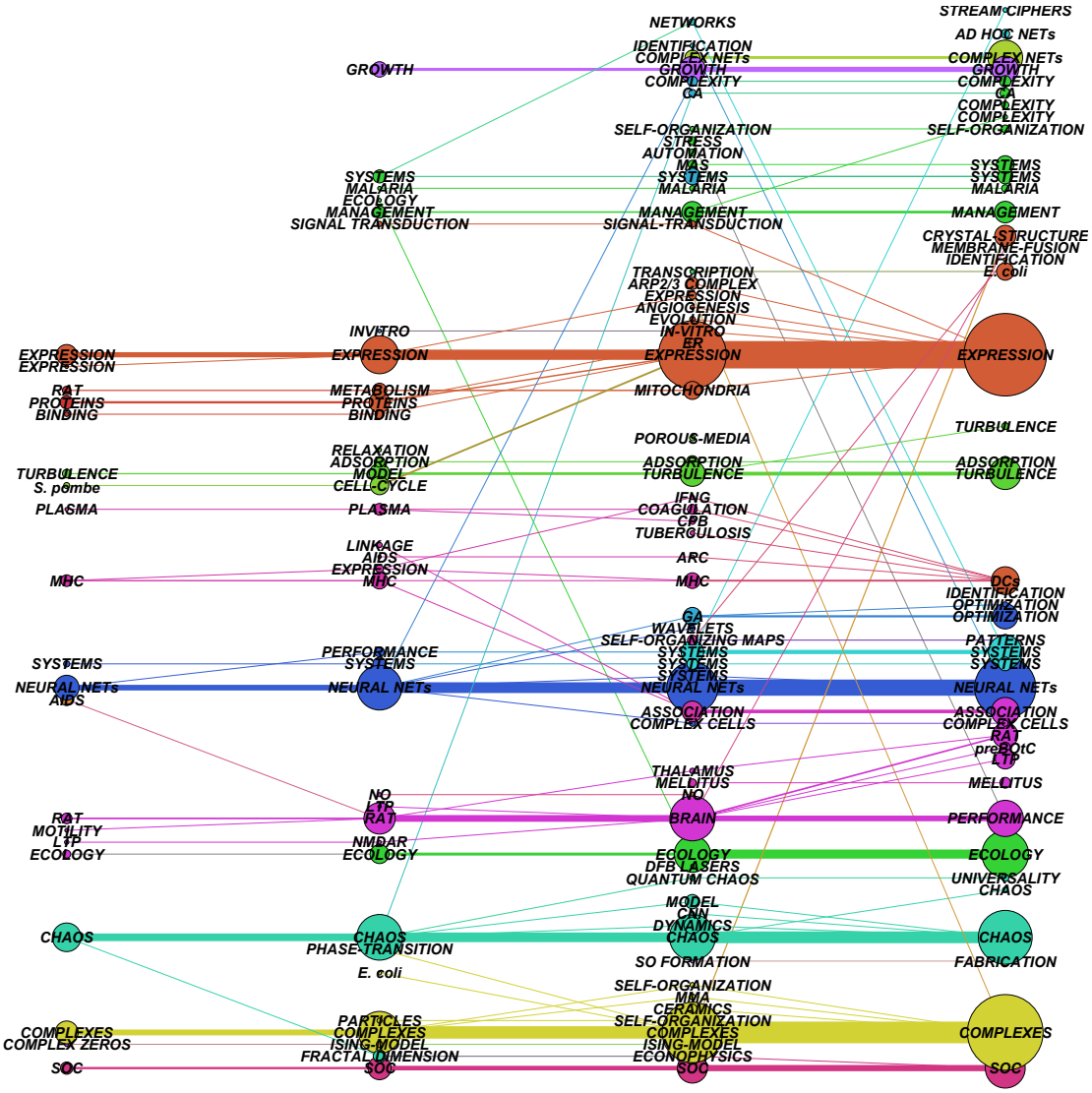


Figure 3.9: Results of our framework on the past history of complex system science network. Robust clusters are marked by the most popular topic keywords. Their colors correspond to the relevant communities as shown in Fig. 2.12.

In the following, we show our results through the citation analysis.

3.3.4 Study of the community evolution

By using citation analysis to detected communities, we examine the topics or fields (see Tab. B.6, Tab. B.7, Tab. B.8 and Tab. B.9), which are used to claim the universality of complex systems, and explain the complex systems history evolution.

Time transition	Mean stability($\widehat{c}$)	std($\widehat{c}$)	Mean stability($\mathcal{C}$)	std($\mathcal{C}$)
From 1985-1994 to 1990-1999	0.916185	0.0830868	0.740646	0.152708
From 1990-1999 to 1995-2004	0.944751	0.0556874	0.812754	0.126775
From 1995-2004 to 2000-2009	0.825698	0.302496	0.778244	0.151847

Table 3.3: Stability values of clusters and core nodes between every pair of consecutive snapshot graphs. Mean and standard deviation are given.

Stable communities. From our results, we can observe that some communities remain very stable and hold their community interests. For example, the community *SOC (Self-organized criticality)* has not been involved into any fusion or split event. By examining its high frequent topic keywords, we can see that the highest frequent topic keywords is "Self-organized criticality" over all time steps, whereas other high frequent topic keywords may change but are still very general in the studies referring to the topic "Self-organized criticality".

Fusion. We can examine fusion events, such as the evolution of the community in red with the highest frequent topic keywords "expression":

- the community *molecular biological:protein* and *molecular biological:gene* merge into the community *molecular biology* at t =1990 – 1999;
- the community *molecular biology* combines *molecular biology: saccharomyces cerevisiae and fission yeast* at t =1995-2004;
- the community *molecular biology* joins the community *immunology* at t =2000-2009.

By analyzing in details high frequent topic keywords, we observe the effects of merge events. At t =1990 – 1999, the merging cluster *molecular biology* contains many high frequent topic key workds like "complex", "binding", "messenger-RNA" and "escherichia-coli". Remark that "complex", "binding" have high frequency in the previous cluster *molecular biology: protein*, and "messenger-RNA" and "escherichia-coli" also have high frequency in the previous cluster *molecular biology:gene*. It means that the merging cluster is similar to the original clusters. Furthermore, the new formed cluster reflects the cooperation between the merged topics or fields. This kind of behavior is helpful to capture history of complex systems, and other new science topics or disciplines.

Split. We analyse split events. For example, the community *neural networks* is split into three distinct clusters *neural networks*, *genetic algorithm* and *computation theory in networks* at t =1995 – 2004. This change reflects the link between the new clusters

and original communities: the genetic algorithm and computation theory in networks seem to be popular approaches to study neural networks.

Our observation over the structural changes have shown the benefits of the computational technique to analyze the history of science. First, it shows important information about science evolution, such as topic evolution, new topic or field emergence. Secondly, it is a good method to analyze cooperation between topics or fields, merge events enable to capture the cooperation between a priori distinct topics or fields. Finally, it is helpful to analyze how a new topic or filed emerges in views of community dynamics.

3.3.5 Robust cluster evolution and overlaps evolution

We use modular overlaps to discuss effects of overlapping nodes in structural changes in dynamic networks. Our approach is different from current academic work on structural properties. For instance, Balazs Vedres and David Stark [125] studied the contribution of overlapping nodes to the structural changes and demonstrated that the overlapping nodes are correlated with *interwoven lineages*, which are ongoing patterns of separation and reunification. Our studies on overlapping nodes are from an evolution perspective and the aim is to better understand how overlapping nodes may explain field evolution.

We study modular overlaps of each snapshot graph, whose results are shown in Tab. B.10, Tab. B.11, Tab. B.12 and Tab. B.13. We observe several modular overlaps over time. For example, the cluster *malaria transmission and mosquito* is shared by *ecosystems*, *molecular biology: serum and hormone* and *neuroscience* at t =1990 – 1999. The cluster *cellular automata* is shared by *chaos theory* and *timeless (gene)* at t =1995 – 2004. The cluster *genetic association* is shared by *molecular biology* and *biological psychology* at t =2000 – 2009. These modular overlaps enable to link different fields, such as cellular automata, it is a popular model used in chaos theory and gene studies.

The Fig. 3.9 shows the results of tracing robust cluster evolution when they contain modular overlaps evolution. We see that modular overlaps may change their community memberships in partitions, such as the modular overlaps *visual cortex*, which emerges and becomes one robust cluster of the community *neural networks* at t =1995 – 2004, and changes its community membership in the partition at t =2000 – 2009: it becomes one robust cluster of *neuroscience*. This case reveals that network evolution may change overlapping nodes performance. It also suggests us to consider overlapping nodes when studying community evolution.

Robust cluster evolution provides an excellent method for examining evolution of modular overlaps. By using robust cluster identification, we can follow their community membership evolution, and analyze the effects of modular overlaps in structural changes. In some cases, we can see that our method can provide reliable information for tracing community cores. For example, at t =1995 – 2004, the community *immunology* joins into the community *neuroscience*. At t =2000 – 2009, we can trace the evolution of the community *immunology* through its core, which is marked by "DCs" in Fig. 3.9.

3.3.6 Discussion and conclusion

Our empirical results show that structural changes can reveal the emergence of new topics or fields. For example, we observe the community *computation science* appearing at t =1995 – 2004. As the community *computation science* is the result of a split of *neural networks*, it implies the intrinsic existence of a link between *computation science* and *neural networks*. Through citation analysis in the cluster *computation theory in networks* like "algorithm", "stability", "networks", we learn that many computational algorithms are used to analyse networks. Although this result fails to capture how research topics or fields (*computation science, ecosystems*,...) are formed in the complex system science, it sheds lights on how to understand complex system science and history. Many topics or fields are used to describe experimental work like neural networks, self-organization criticality while new topics or fields come from modelling practices or theoretical applications like computation science.

Our framework supports modular overlaps and enables to trace their evolution. In the complex system science studies, modular overlaps refer to collaborations between distinct topic of fields, such as visual cortex which is a topic relating to neuroscience and neural networks. In other contexts of citation analysis such as the analysis on biology and social systems [75], modular overlaps may refer to interdisciplinary collaborations, which are essential scientific challenges. As our framework is able to trace modular overlaps evolution, it provides new insights in understanding the history of interdisciplinary evolution.

Many studies on co-citation networks endeavours to mine the evolution of science construction and expect to give insights into *field mobility* or *paradigm shift*. The *field mobility* describes how one author changes its topic over time [57]. The *paradigm shift* is proposed by Thomas Kuhn [67]. The author describes a discipline change in views of paradigm, where a paradigm is a scientific community. The results of field mobility is measured through a function of publications over time. The field mobility is defined as scientists moving into new research topics. Corresponding to the performance of our method, it seems that both paradigm shift and field mobility can be captured by analysing community dynamics.

Chapter 4

Conclusion

4.1 Summary

In this thesis, we have explored computational techniques to study community organization of complex networks with overlapping nodes. It is known that finding the communities within a network is a powerful tool for understanding the structure and the functioning of the network, and its dynamic mechanisms. In Section 1.1, we have described the definition of communities. Lately, we focus on the current problem in community detection. The two major problems concerning community detection are overlapping community detection and dynamic community detection.

In Chapter 1, we have discussed current research on the problem of community detection in dynamic networks, which have left us with a number of important open issues such as benchmark graphs. From our exposition it appears that current methods can be classified into three categories: two-stage methods, evolutionary clustering and coupling graph clustering. Different problems are raised by them, respectively.

Communities may overlap in real networks. In Chapter 2, we proposed a quality function for measuring the quality of covers and two definitions for overlapping nodes: granular overlaps and modular overlaps. For both definitions, we proposed a method called clique optimization to detect granular overlaps and also proposed a method named fuzzy detection to capture modular overlaps. Both methods have been applied to synthetic networks and real networks. The obtained results have shown that both methods can be used for characterizing overlapping nodes but in distinct and complementary views.

In Chapter 3, we have explored a mapping method and a visualization tool to study community evolution in dynamic networks. We have applied the definition of predecessor/successor relationship to track community evolution and identify community dynamics. The visualization tool of lineage diagrams has been introduced. It enables to show and explore the evolution of dynamic communities. We have conducted experiments with real data sets to assess the applicability of the proposed methods. The experiments have shown that the algorithms achieve the goals they are designed for.

4.2 Future works

This work is a first step in a more global research on dynamic networks. Next, we will apply our methods to the visual tool, such as in Fig. 4.1. (Communities are identified by colors. Among different communities, it is overlapping nodes that connect them such as the overlapping node labelled by "little".) Of course, many effects will be made. It is still a problem to visualize the evolution of overlapping communities, in particular the evolution of overlapping nodes.

Moreover, our method for detecting community evolution needs additional improvements and require further investigations. For example, our method is desirable to detect and analyse the evolution of communities in large, noisy networks that exhibit a high number of changes over time. But it fails to identify artificial community changes, which are caused by the community detection algorithm itself. If we want to improve the accuracy of our method, it is better to add more constraints to smooth the shifts of community members.

We hope to mine more time-dependent structural properties, in particular the structural properties about overlapping nodes. For instance, Asur *et al.* [3] have measured the *sociability index*, which gave high scores to nodes that were involved in interactions with different groups. Their analysis showed that the sociability index could be used to predict future co-occurrences of nodes in clusters. It is not difficult to measure the sociability index of overlapping nodes. Then we can analyse the influence of overlapping nodes to future community evolution. It is meaningful to take overlapping nodes into account for studying structural properties.

Several problems remain in community detection such as benchmark graphs. We have reviewed benchmarks [22, 31, 51] in Section 1.4. The current computer-generated benchmark graphs for community detection in dynamic graphs, are constructed by randomly changing interactions between nodes. In these benchmarks, most changes on topology correspond to a predefined probability, that is, the nodes belonging to the same community change their neighbours with the same probability. However, in real networks, changes on topologies should be heterogeneous.

Therefore, it is better to validate the proposed algorithm by applying it to real network benchmarks. In [88], some real networks are used as benchmark graphs. These real networks do not have any topology change. They only change the resolution scale by varying the resolution parameter. We need real network benchmarks whose community evolution is analysed and known a priori.

Offering benchmark graphs is a crucial problem in the area of community detection in dynamic networks.

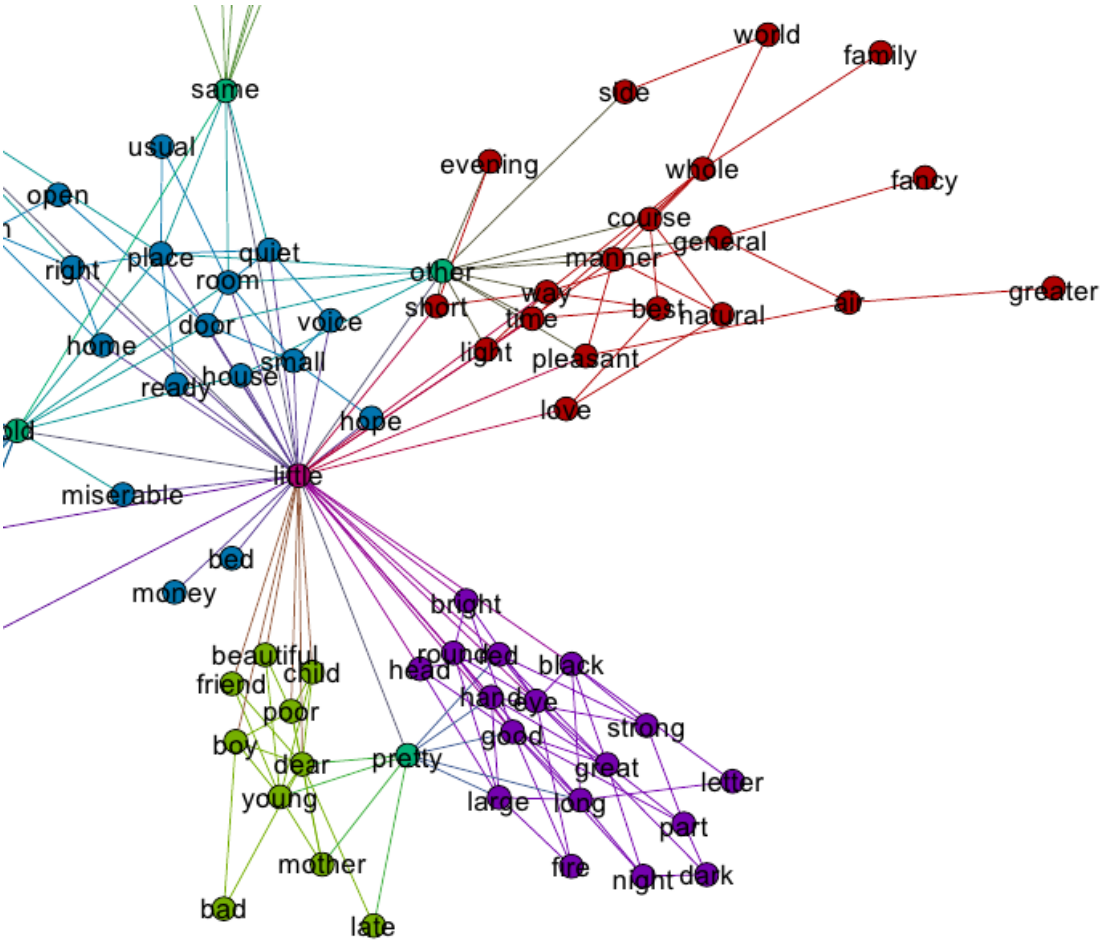


Figure 4.1: The community structure of adjacency network of common adjectives and nouns in the novel David Copperfield by Charles Dickens [92]. The observed overlapping communities are detected by our clique optimization. The community membership is shown by color such that the nodes belonging to the same community are in the same colour. For the visualisation, we only show internal edges. Between different communities, all observed connections are adjacent to overlapping nodes. Moreover, the color of overlapping nodes correspond to the number of community memberships: if a node is in green such as the node labelled by "pretty", it is shared by two communities; if it is in rose, it is shared by 4 communities. We observe that the overlapping node in rose labelled by "little" combines several communities such as the community in red describing manner ("manner", "way", "natural", *etc.*), the community in blue describing place ("place","room","door", *etc.*), the community in green describing people ("child","boy","mother", *etc.*) and the community in violet describing eye and hand ("eye", "hand", "black", "strong", *etc.*). Therefore, "little" seems important for these descriptions.

Bibliography

[1] J. I. Alvarez-hamelin, A. Barrat, and A. Vespignani. Large scale networks fingerprinting and visualization using the k-core decomposition. In *Advances in Neural Information Processing Systems 18*, pages 41–50. MIT Press, 2006.

[2] A. Arenas and C. J. Diaz-Guilera, Albert Perez-Vicente. Synchronization reveals topological scales in complex networks. 96:114102, 2006.

[3] S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 921. ACM, 2007.

[4] T. Aynaud. *Détection de communautés dans les réseaux dynamiques*. PhD thesis, DOCTEUR DE L'UNIVERSITÉ PIERRE ET MARIE CURIE, 2011.

[5] T. Aynaud and J.-L. Guillaume. Long range community detection. In *Latin American Workshop on Dynamic Networks*, 2010.

[6] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 54, 2006.

[7] D. A. Bader and K. Madduri. Gtgraph: A synthetic graph generator suite. 2006.

[8] Y. Bar-Yam. *Dynamics of complex systems*. Addison-Wesley stydies in nonlinearity. Westview Press, 2003.

[9] A.-L. Barabasi and E. Bonabeau. Scale-free networks. *Sci Am*, 288(5):60–69, May 2003.

[10] V. Batagelj, P. Doreian, and A. Ferligoj. Generalized blockmodeling of two-mode network data. *Social Networks*, 26(1):29–53, 2004.

[11] J. Baumes, M. Goldberg, and M. Magdon-Ismail. Efficient identification of overlapping communities. *Intelligence and Security Informatics, Proceedings*, 3495:27–36, 2005.

[12] M. Beiró and J. Busch. Visualizing communities in dynamic networks. In *Latin American Workshop on Dynamic Networks*, volume 1, 2010.

[13] R. Bekkerman, A. Mccallum, and G. Huang. Automatic categorization of email into folders: Benchmark experiments on enron and sri corpora. In *In Technical Report, Computer Science department, IR-418*, pages 4–6.

[14] M. Bengtsson and P. Roivainen. Using the potts glass for solving the clustering problem. *Int J Neural Syst*, 6(2):119–132, Jun 1995.

[15] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[16] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics-Theory and Experiment*, 2008.

[17] K. P. Burnham and D. R. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*, volume 172. Springer, 2002.

[18] J. Camacho, R. Guimerá, and L. A. N. Amaral. Robust patterns in food web structure. *Phys Rev Lett*, 88(22):228102, Jun 2002.

[19] D. Chakrabarti, S. Papadimitriou, D. S. Modha, and C. Faloutsos. Fully automatic cross-associations. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 79–88, New York, NY, USA, 2004. ACM.

[20] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-mat: A recursive model for graph mining. In *In SDM*, 2004.

[21] J. Chen and B. Yuan. Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics*, 22(18):2283–2290, Sep 2006.

[22] Z. Chen, K. a. Wilson, Y. Jin, W. Hendrix, and N. F. Samatova. Detecting and Tracking Community Dynamics in Evolutionary Networks. *2010 IEEE International Conference on Data Mining Workshops*, pages 318–327, Dec. 2010.

[23] Y. Chi, S. Zhu, X. Song, J. Tatemura, and B. L. Tseng. Structural and temporal analysis of the blogosphere through community factorization. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, page 163, 2007.

[24] H. H. Clark and S. A. Brennan. Grounding in communication. In L. B. Resnick, J. M. Levine, and S. D. Teasley, editors, *Perspectives on socially shared cognition.* 1991.

[25] A. Clauset, C. Moore, and M. E. J. Newman. Structural inference of hierarchies in networks. In *Proceedings of the 2006 conference on Statistical network analysis*, ICML'06, pages 1–13, Berlin, Heidelberg, 2007. Springer-Verlag.

[26] A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18:116–140, 1999.

[27] K. Crawford. Six provocations for big data. *Computer*, pages 1–17, 2011.

[28] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008–P09008, Sept. 2005.

[29] T. Dinh, I. Shin, N. Thai, and M. Thai. A General Approach for Modules Identification in Evolving Networks. *Dynamics of Information*, 40(4):83–100, 2010.

[30] Y. Dourisboure, F. Geraci, and M. Pellegrini. Extraction and classification of dense communities in the web. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 461–470, New York, NY, USA, 2007. ACM.

[31] D. Duan, Y. Li, Y. Jin, and Z. Lu. Community mining on dynamic weighted directed graphs. In *Proceeding of the 1st ACM international workshop on Complex networks meet information and knowledge management*, page 1118. ACM, 2009.

[32] H. Ebel, L.-I. Mielsch, and S. Bornholdt. Scale-free topology of e-mail networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 66(3 Pt 2A):035103, Sep 2002.

[33] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap.* Chapman & Hall, New York, 1993.

[34] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, 95(25):14863–14868, Dec 1998.

[35] N. B. Ellison, C. Steinfield, and C. Lampe. The benefits of facebook friends: Social capital and college students use of online social network sites. *Journal of Computer-Mediated Communication*, 12(4):1143–1168, July 2007.

[36] M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental clustering for mining in a data warehousing environment. In A. Gupta, O. Shmueli, and J. Widom, editors, *VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA*, pages 323–333. Morgan Kaufmann, 1998.

[37] M. Ester, H. peter Kriegel, J. S, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[38] T. S. Evans and R. Lambiotte. Line graphs, link partitions, and overlapping communities. *Physical Review E*, 80(1), 2009.

[39] T. Falkowski, A. Barth, and M. Spiliopoulou. Studying community dynamics with an incremental graph mining algorithm. In *Proc. of the 14 th Americas Conference on Information Systems (AMCIS 2008)*, pages 1–11, 2008.

[40] T. Falkowski and M. Spiliopoulou. Data mining for community dynamics. *Kunstliche Intelligenz*, 3:23–29, 2007.

[41] T. Falkowski and M. Spiliopoulou. Users in volatile communities: Studying active participation and community evolution. *Lecture Notes in Computer Science*, 4511:47, 2007.

[42] T. Falkowski, M. Spiliopoulou, and J. Bartelheimer. Community dynamics mining. In *Proceedings of 14th European Conference on Information Systems (ECIS 2006)*, Goteborg, 2006. Citeseer.

[43] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[44] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, 104(1):36–41, 2007.

[45] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of The Twentieth International Joint Conference for Artificial Intelligence*, pages 1606–1611, Hyderabad, India, 2007.

[46] D. Gfeller, J.-C. Chappelier, and P. De Los Rios. Finding instabilities in the community structure of complex networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 72(5 Pt 2):056135, Nov. 2005.

[47] C. L. Giles, K. D. Bollacker, and S. Lawrence. Citeseer: An automatic citation indexing system. pages 89–98. ACM Press, 1998.

[48] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99,:7821–7826, 2002.

[49] P. Gongla and C. R. Rizzuto. Where did that community go? - communities of practice that disappear. In P. Hildreth and C. Kimble, editors, *Knowledge Networks: Innovation through Communities of Practice*. Idea Group Publishing, Hershey, PA, USA, 2004.

[50] R. Görke, P. Maillard, and C. Staudt. Modularity-Driven Clustering of Dynamic Graphs. *Experimental Algorithms*, Cl(1), 2010.

[51] D. Greene and D. Doyle. Tracking the evolution of communities in dynamic social networks. In *Advances in Social Networks Analysis and Mining (ASONAM)*, volume 2010, pages 1–13. IEEE, 2010.

[52] S. Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):103018, 2010.

[53] P. D. Grnwald, I. J. Myung, and M. A. Pitt. *Advances in Minimum Description Length: Theory and Applications (Neural Information Processing)*. The MIT Press, 2005.

[54] R. Guimerá and L. A. N. Amaral. Cartography of complex networks: modules and universal roles. *J Stat Mech*, 2005(P02001):P02001, Feb 2005.

[55] R. Guimerá, M. Sales-Pardo, and L. A. N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 70(2 Pt 2):025101, Aug 2004.

[56] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402(6761):–47, 1999.

[57] I. Hellsten, R. Lambiotte, A. Scharnhorst, and M. Ausloos. Self-citations, co-authorships and keywords: A new approach to scientists' field mobility? *Scientometrics*, 72(3):469–486, 2007.

[58] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Tracking evolving communities in large linked networks. In *National Academy of Sciences of the United States of America*, volume 101, page 5249. National Acad Sciences, 2004.

[59] J. P. Hugot, M. Chamaillard, H. Zouali, S. Lesage, J. P. Cézard, J. Belaiche, S. Almer, C. Tysk, C. A. O'Morain, M. Gassull, V. Binder, Y. Finkel, A. Cortot, R. Modigliani, P. Laurent-Puig, C. Gower-Rousseau, J. Macry, J. F. Colombel, M. Sahbatou, and G. Thomas. Association of nod2 leucine-rich repeat variants with susceptibility to crohn's disease. *Nature*, 411(6837):599–603, May 2001.

[60] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1988.

[61] M. James and W. D. R. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review*, 68(1):103–127, 2003.

[62] M. B. Jdidia, C. Robardet, and E. Fleury. Communities detection and analysis of their dynamics in collaborative networks. In *ICDIM*, pages 744–749. IEEE, 2007.

[63] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, May 2001.

[64] E. M. Jin, M. Girvan, and M. E. J. Newman. Structure of growing social networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 64(4 Pt 2):046132, Oct 2001.

[65] M. M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14(1):10–25, 1963.

[66] A. E. Krause, K. A. Frank, D. M. Mason, R. E. Ulanowicz, and W. W. Taylor. Compartments revealed in food-web structure. *Nature*, 426(6964):282–285, Nov 2003.

[67] T. S. Kuhn. *The Structure of Scientific Revolutions*. University Of Chicago Press, 3rd edition, Dec. 1996.

[68] R. Kumar, A. Tomkins, and D. Chakrabarti. Evolutionary clustering. In *In Proc. of the 12th ACM SIGKDD Conference*, 2006.

[69] J. M. Kumpula, M. Kivela, K. Kaski, and J. Saramaki. A sequential algorithm for fast clique percolation. May 2008.

[70] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *ArXiv e-prints*, 2009.

[71] A. Lancichinetti, S. Fortunato, and J. Kertesz. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11, 2009.

[72] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 93046110, 2008.

[73] A. Lancichinetti, F. Radicchi, and S. Ramasco, José Javier Fortunato. Finding statistically significant communities in networks. 2010.

[74] A. Lancichinetti and J. J. Radicchi, Filippo Ramasco. Statistical significance of communities in networks. 81,:046110, 2009.

[75] D. Lazer, A. Pentland, L. Adamic, S. Aral, A.-L. Barabási, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, T. Jebara, G. King, M. Macy, D. Roy, and M. Van Alstyne. Computational social science. *Science*, 323(5915):721–723, 2009.

[76] C. Lee, F. Reid, A. McDaid, and N. Hurley. Detecting highly overlapping community structure by greedy clique expansion. *ArXiv e-prints*, feb 2010.

[77] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Oct. 2008.

[78] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 631–640, New York, NY, USA, 2010. ACM.

[79] J. V. Limbergen, R. K. Russell, E. R. Nimmo, L. Torkvist, C. W. Lees, H. E. Drummond, L. Smith, N. H. Anderson, P. M. Gillett, P. McGrogan, K. Hassan, L. T. Weaver, W. M. Bisset, G. Mahdi, I. D. Arnott, U. Sjoqvist, M. Lordal, S. M. Farrington, M. G. Dunlop, D. C. Wilson, and J. Satsangi. Contribution of the nod1/card4 insertion/deletion polymorphism +32656 to inflammatory bowel disease in northern europe. *Inflamm Bowel Dis*, 13(7):882–889, Jul 2007.

[80] R. D. LUCE and A. D. PERRY. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, Jun 1949.

[81] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, Dec. 2007.

[82] Y. Malhotra. *Knowledge management and virtual organizations*. Idea Group Publishing, 2000.

[83] R. N. Mantegna. Hierarchical structure in financial markets. *European Physical Journal B*, 11:193–197, 1999.

[84] C. P. Massen and J. P. K. Doye. Thermodynamics of Community Structure. *eprint arXiv:cond-mat/0610077*, Oct. 2006.

[85] F. McSherry. Spectral partitioning of random graphs. In *Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, FOCS '01, pages 529–, Washington, DC, USA, 2001. IEEE Computer Society.

[86] F. Michon and M. Tummers. The dynamic interest in topics within the biomedical scientific community. *PLoS ONE*, 4(8):e6544, 08 2009.

[87] D. M. Moody James and S. Bender-deMoll. Visualizing network dynamics. *American Journal of Sociology*, January 2005.

[88] P. Mucha, T. Richardson, K. Macon, and M. A. Porter. Community structure in time-dependent, multiscale, and multiplex networks. *science*, 876:10–13, 2010.

[89] T. Nepusz, A. Petroczi, L. Negyessy, and F. Bazso. Fuzzy communities and the concept of bridgeness in complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 77(1 Pt 2):016107, Jan 2008.

[90] M. E. J. Newman. Analysis of weighted networks. *Phys. Rev. E*, 70:056131, Nov 2004.

[91] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 69(6 Pt 2):066133, Jun 2004.

[92] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74:036104, Sep 2006.

[93] M. E. J. Newman. Modularity and community structure in networks. *Proc Natl Acad Sci U S A*, 103(23):8577–8582, Jun 2006.

[94] M. E. J. Newman, A. L. Barabási, and D. J. Watts, editors. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[95] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):26113, 2004.

[96] H. Ning, W. Xu, Y. Chi, Y. Gong, and T. Huang. Incremental spectral clustering with application to monitoring of evolving blog communities. *In SIAM Int. Conf. on Data Mining*, 2007.

[97] G. Palla, A.-L. Barabasi, and T. Vicsek. Quantifying social group evolution. *Nature*, 446:664–667, 2007.

[98] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814, 2005.

[99] R. Pastor-Satorras, A. Vázquez, and A. Vespignani. Dynamical and correlation properties of the internet. *Phys Rev Lett*, 87(25):258701, Dec 2001.

[100] R. Pastor-Satorras and A. Vespignani. *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press, New York, NY, USA, 2004.

[101] D. Pauleen. *Virtual teams: projects, protocols and processes*. Idea Group Pub., 2004.

[102] P. Pons and M. Latapy. Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, 10:191–218, 2006.

[103] S. Pu, J. Wong, B. Turner, E. Cho, and S. J. Wodak. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Res*, 37(3):825–831, Feb 2009.

[104] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proc Natl Acad Sci U S A*, 101(9):2658–2663, Mar 2004.

[105] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, Aug 2002.

[106] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Phys. Rev. E*, 74(1):016110, Jul 2006.

[107] E. J. Robert S. Weiss. A method for the analysis of the structure of complex organizations. *American Sociological Review*, 20:661, 1955.

[108] M. Rosvall. Mapping change in large networks. *PLoS One*, pages 1–9, 2010.

[109] M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. In *Proceedings of the 15th European Conference on Machine Learning (ECML 2004). Lecture Notes in Artificial Intelligence*, pages 371–383. Springer-Verlag, 2004.

[110] M. Sales-Pardo, R. Guimera, and L. A. N. Moreira, Andra A Amaral. Extracting the hierarchical organization of complex systems. *Proc Natl Acad Sci U S A*, 104(39):15224–15229, 2007.

[111] J. Scott. *Social network analysis: a handbook*. SAGE Publications, 2000.

[112] H. W. Shen, X. Q. Cheng, and J. F. Guo. Quantifying and identifying the overlapping community structure in networks. *Journal of Statistical Mechanics-Theory and Experiment*, 2009.

[113] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997.

[114] G. Simmel. The persistence of social groups. *American Journal of Sociology*, 3 (1897): 662-698.

[115] H. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24(4):265–269, 1973.

[116] S.-W. Son, H. Jeong, and J. D. Noh. Random field ising model and community structure in complex networks. *The European Physical Journal B*, 50:431, 2006.

[117] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult. Monic: modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 706–711. ACM New York, NY, USA, 2006.

[118] J. Sun, C. Faloutsos, S. Papadimitriou, and P. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 687–696. ACM New York, NY, USA, 2007.

[119] L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In *International Conference on Knowledge Discovery and Data Mining*, page 8, 2008.

[120] C. Tantipathananandh and T. Berger-Wolf. Constant-factor approximation algorithms for identifying dynamic communities. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 827–836, New York, NY, USA, 2009. ACM.

[121] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, page 717, 2007.

[122] A. Traud, E. Kelsic, P. Mucha, and M. Porter. Community Structure in Online Collegiate Social Networks. In *APS March Meeting Abstracts*, page H9013, Mar. 2009.

[123] B. L. Tseng, Y.-R. Lin, Y. Chi, S. Zhu, and H. Sundaram. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. *Social Networks*, pages 685–694, 2008.

[124] B. L. Tseng, Y.-R. Lin, Y. Chi, S. Zhu, and H. Sundaram. Analyzing communities and their evolutions in dynamic social networks. *ACM Transactions on Knowledge Discovery from Data*, 3(2):1–31, 2009.

[125] B. Vedres and D. Stark. Structural folds: Generative disruption in overlapping groups. *American Journal of Sociology*, January 2010.

[126] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks. *CoRR*, abs/cs/0702048, 2007.

[127] X. H. Wang, L. C. Jiao, and J. S. Wu. Adjusting from disjoint to overlapping community detection of complex networks. *Physica a-Statistical Mechanics and Its Applications*, 388(24):5045–5056, 2009.

[128] Y. Wang, B. Wu, and N. Du. Community Evolution of Social Network: Feature, Algorithm and Model. *Science And Technology*, (60402011), 2008.

[129] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences)*. Structural analysis in the social sciences, 8. Cambridge University Press, 1 edition, Nov. 1994.

[130] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, Jun 1998.

[131] S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. In *SDM*, pages 43–55, 2005.

[132] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin. A bayesian approach toward finding communities and their evolutions in dynamic social networks. In *SIAM Conference on Data Mining (SDM)*, 2009.

[133] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropologica*, 1(33):452–473, 1977.

Appendix A

List of publications

A.1 International Conferences

- Q.Wang and E.Fleury, *Detecting overlapping communities in graphs*, European conference on Complex Systems 2009, University of Warwick, UK
- Q.Wang and E.Fleury, *Uncovering Overlapping Community Structure*, 2nd Workshop on Complex Networks, Brazil, 2010
- Q.Wang and E.Fleury, *Mining time-dependent communities*, Latin-American Workshop on Dynamic Networks, Argentina, 2010
- Q.Wang and E.Fleury, *Community detection with fuzzy community structure*, The First Workshop on Social Network Analysis in Applications, ASONAM 2011:International Conference on Advances in Social Networks Analysis and Mining,Taiwan, 2011 (Best paper award)
- Q.Wang and E.Fleury, *Understanding community evolution in Complex systems science*, 1st International Workshop on Dynamicity, December 12, Collocated with OPODIS 2011, Toulouse, France

A.2 Poster

- Q.Wang and E.Fleury, *Community detection with fuzzy community*, Interdisciplinary Workshop on Information and Decision in Social Networks, MIT, 2011

A.3 Journals

- Q.Wang and E.Fleury, *Fuzziness and overlapping community structure in complex networks*, J.UCS (Journal of Universal Computer Science) special issue (en review) One chapter of the book

- Q.Wang and E.Fleury, *Fuzzy community structure and modular overlaps*, SNSI2012 (Studies in Mining Social Networks and Security Informatics by Springer Verlag) (en review)
- Thomas Aynaud, Eric Fleury, Jean-Loup Guillaume, Qinna Wang, *Communities in evolving networks: definitions, detection and analysis techniques*, en preparation
- Q.Wang and E.Fleury, *Overlapping time-dependent community detection in dynamic networks*, en preparation

A.4 Seminars

- Q.Wang and E.Fleury, *Fuzziness and overlapping communities in large-scale networks* Journées non thématique Octobre, Paris, 2011.
- Q.Wang and E.Fleury, *Mining time-dependent communities* Journées automnales ResCom, Lyon, 2010

Appendix B

Past history complex system science DATA set & keywords

Community	Highest frequent topic keywords	High frequent topic keywords
Biological Psychology	Brain	Brain, Neurons, Long-Term Potentiation, Association, Expression, Performance, Disease, Model, Synaptic Plasticity, Activation, Complex, Children, Central-Nervous-System, Rat
Chaos Theory	Chaos	Chaos, Dynamics, Systems, Model, Stability, Complexity, Synchronization, Time-Series, Bifurcation, Self-Organization
Spectroscopy	Complexes	Complexes, Self-Organization, Crystal-Structure, Chemistry, Derivatives, Behavior, Films, Polymers, Systems, Phase-Transition, Spectroscopy, Dynamics, Thin-Films, Molecules, Nonlinear-Optical Properties
Complex Networks	Complex Networks	Complex Networks, Dynamics, Small-World Networks, Model, Internet, Evolution, Systems, Organization, Topology, Scale-Free Networks, Metabolic Networks, Web, Graphs
Ecosystems	Ecology	Ecology, Systems, Model, Complexity, Evolution, Dynamics, Management, Growth, Behavior, Self-Organization, Patterns, Simulation, Biodiversity, Models
Molecular Biology	Expression	Expression, Complex, Gene-Expression, Protein, In-Vivo, Activation, Saccharomyces-Cerevisiae, Identification, Gene, Escherichia-Coli, Cells, In-Vitro, Binding, Crystal-Structure, Messenger-Rna, Phosphorylation, Proteins
Semiconductor Superlattice Materials And Growth Technology	Growth	Growth, Gaas, Islands, Molecular-Beam Epitaxy, Self-Organization, Quantum Dots, Surfaces, Films, Photoluminescence, Silicon, Nanostructures, Si(001)
Clinical Psychology	Management	Management, Therapy, Trauma, Experience, Hemorrhage, Surgery, Inhibitors, Optimization, Recombinant Factor Viia, Damage Control, Mortality, Cancer
Systems Neuroscience	Neuralnetwors	Neural Networks, Model, Systems, Classification, Optimization, Algorithm, Identification, Design, Prediction, Self-Organizing Maps
Soc	Self-Organized Criticality	Self-Organized Criticality, Model, Dynamics, Econophysics, Evolution, Systems, Fluctuations, Behavior, Growth, Turbulence, Noise, Transport, Avalanches, Earthquakes, Patterns, Time-Series
Computer Science	Systems	Systems, Design, Performance, Channels, Algorithm, Networks, Capacity, Ofdm, Stability, Optimization, Fading Channels, Algorithms, Model, Signals, Codes, Transmission
Dynamics Turbulence	Turbulence	Turbulence,Model, Flow, Simulation, Dynamics, Behavior, Large-Eddy Simulation, Complex Terrain, Plasticity, Flows, Boundary-Layer

Table B.1: Results of communities in the partition. The results shown in high frequent keywords are sorted in descending order and each keywords are contained by at least 20 articles.

	Complex networks	Neural networks	Semiconductor superlattice materials and growth technology	Ecosystems	SOC
Molecular biology		**SACCHAROMYCES-Cerevisiae**, Identification, Yeast, Complex Networks, Gene-Expression, Patterns, Database, Cell-Cycle, Organization, Self-Organizing Maps			**Complex Networks**, Organization, Dynamics, Model, Evolution, Metabolic Networks, Systems, Topology, Small-World Networks, Escherichia-Coli
Chaos theory	**Synchronization**, Systems, Dynamics, Model, Complex Networks, Chaos, Self-Organized Criticality, Stability, Complexity, Econophysics	**Chaos**, Systems, Dynamics, Model, Complexity, Neural Networks, Time-Series, Synchronization, Stability, Networks		**Dynamics**, Chaos, Model, Systems, Complexity, Self-Organization, Stability, Evolution, Cellular Automata, Patterns	**Dynamics**, Systems, Self-Organized Criticality, Chaos, Time-Series, Complexity, Model, Complex Networks, Econophysics, Synchronization
Neuroscience: biological psychology	**Complex Networks**	**Neurons**, Model, Primary Visual-Cortex, Complex Cells, Visual-Cortex, Epistasis, Receptive-Fields, Multifactor-Dimensionality Reduction, Cortex, Association			
Chemistry: spectroscopy		**Dynamics**	**SELF-Organization**, Superlattices, Nanoparticles, Clusters, Quantum Dots, Nanocrystals, Total-Energy Calculations, Particles, Self-Organized Growth		

Table B.2: Results of clique optimization at k=5: ten high frequent topic keywords contained by granular overlaps between pairs of communities. The shown high frequent topic keywords are sorted in descending order and each topic keyword is contained in at least 20 articles. The highest frequent topic keywords are shown in bold font.

	Complex networks	Systems neuroscience	Semiconductor superlattice materials and growth technology	Ecosystems	SOC
Molecular biology		**Saccharomyces-Cerevisiae**, Identification, Yeast, Gene-Expression, Patterns, Cell-Cycle, Database, Complex Networks, Neural-Network, Self-Organizing Maps			**Complex Networks**, Dynamics, Organization, Saccharomyces-Cerevisiae, Model, Evolution
Chaos theory	**Synchronization**, Systems, Dynamics, Complex Networks, Chaos, Model, Self-Organized Criticality, Stability, Chaotic Systems, Oscillators	**Chaos**, Systems, Neural Networks, Dynamics, Complexity, Time-Series, Model, Stability, Communication, Synchronization		**Dynamics**, Self-Organization, Model, Complexity, Chaos, Systems, Stability, Patterns	**Dynamics**, Self-Organized Criticality, Time-Series, Systems, Chaos, Complexity, Model, Complex Networks, Econophysics, Synchronization
Biological Psychology		**Neurons**, Model, Epistasis, Multifactor-Dimensionality Reduction, Primary Visual-Cortex, Receptive-Fields, Association, Complex Cells, Cortex, Natural Images			
Spectroscopy			**Self-Organization**, Superlattices, Nanoparticles, Clusters, Quantum Dots, Total-Energy Calculations, Nanocrystals, Self-Organized Growth, Nanostructures, Wave Basis-Set		

Table B.3: Results of clique optimization at k=6: ten high frequent topic keywords contained by granular overlaps between pairs of communities. These high frequent topic keywords are contained in at least 20 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

Cluster	High frequent topic keywords	Involving communities
Genetic association	**Association**, Susceptibility, Polymorphism, Linkage Disequilibrium, Disease, Major Histocompatibility Complex, Linkage, Complex Traits, Risk, Population	Molecular Biology, Biological Psychology
Discrete-Event Systems	**Systems**, Supervisory Control, Petri Nets, Complexity, Discrete-Event Systems, Verification, Design, Automata, Synchronization, Discrete Event Systems	Computer Science, Ecosystems
Computational Complexity	**Complexity**, Algorithms, Computational Complexity, Algorithm, Networks, Optimization, Time, Systems, Search, Computational-Complexity	Computer Science, Ecosystems
Astronomy-Ism(Interstellar Medium)	**Turbulence**, Ism : Clouds, Star-Formation, Stars : Formation, Molecular Clouds, Ism : Structure, Ism : Kinematics And Dynamics, Evolution, Radio Lines : Ism, Intergalactic Medium	Dynamics Turbulence, Clinical Psychology
Multi-Agent Systems	**Systems**, Multi-Agent Systems, Multiagent Systems, Design, Agents, Architecture, Multi-Agent System, Framework, Model, Intelligent Agents	Computer Science, Ecosystems
Visual Cortex	**Complex Cells**, Lateral Geniculate-Nucleus, Cat Striate Cortex, Primary Visual-Cortex, Striate Cortex, Cortical-Neurons, Receptive-Fields, Contrast, Orientation Selectivity, Simple Cells	Biological psychology, Systems neuroscience

Table B.4: Results of fuzzy detection: ten high-frequent topic keywords contained by modular overlaps between pairs of communities. These high frequent topic keywords are contained in at least 20 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

Community	Cluster	High frequent topic keywords
Dynamics Turbulence	Flow over Complex Terrain	**Turbulence**, Model, Flow, Simulation, Complex Terrain, Large-Eddy Simulation, Flows, Behavior, Boundary-Layer, Plasticity
	Astronomy-ISM (Interstellar Medium)	**Turbulence**, Ism : Clouds, Star-Formation, Stars : Formation, Ism : Structure, Molecular Clouds, Ism : Kinematics And Dynamics, Evolution, Radio Lines : Ism, Intergalactic Medium
Computer Science: Communication Systems	Telecommunication System	**Systems**, Performance, Channels, Synchronization, Fading Channels, Capacity, Ofdm, Equalization, Networks, Multiuser Detection
	Control Theory	**Systems**, Stability, Design, Robust Control, Optimization, Linear-Systems, Model-Predictive Control, Stabilization, H-Infinity Control, Model Predictive Control
	Wireless Network	**Ad Hoc Networks**, Sensor Networks, Wireless Sensor Networks, Self-Organization, Networks, Wireless Networks, Clustering
	Cryptography	**Stream Ciphers**, Cryptanalysis, Linear Complexity, Stream Cipher, Sequences
Molecular Biology	Expression	**Expression**, Complex, Gene-Expression, Protein, Saccharomyces-Cerevisiae, Gene, Activation, In-Vivo, Identification, In-Vitro
	Dendritic Cells	**Dendritic Cells**, In-Vivo, Expression, T-Cells, Infection, Complex, Mice, Activation, Major Histocompatibility Complex, Antigen
	Crystal Structure Of Escherichia Coli	**Crystal-Structure**, Complex, Escherichia-Coli, Binding, Protein, Recognition, Mechanism, Proteins, Molecular-Dynamics, Complexes
	Gene Expression In Escherichia Coli	**Escherichia-Coli**, Gene-Expression, Systems, Expression, Model, Networks, Systems Biology, Protein, Transcription, Rhythms
	Atherosclerosis	**Atherosclerosis**, Inflammation, Expression, Disease, Myocardial-Infarction, In-Vivo, C-Reactive Protein, Smooth-Muscle-Cells, Activation, Low-Density-Lipoprotein
	Membrane Fusion And Exocytosis	**Membrane-Fusion**, Neurotransmitter Release, Exocytosis, Syntaxin, Snare, Complex, Protein, Snare Complex, Transmitter Release
	Proteomics	**Identification**, Proteomics, Mass-Spectrometry, Proteins, Peptides, Protein Identification
	Chaotic Dynamics	**Chaos**, Dynamics, Systems, Complexity, Stability, Model, Time-Series, Synchronization, Nonlinear Dynamics,

		Bifurcation
Chaos Theory	Quantum Chaos And Universality	**Universality**, Quantum Chaos, Systems, Chaos, States, Model, Random-Matrix Theory, Complex Systems, Fluctuations, Spectra
	Chaos In Population Dynamics	**Chaos**, Stability, Dynamics, Population, Permanence, Models, Systems, Bifurcation, Predator-Prey System, Birth Pulses
Neuroscience: Biological Psychology	Neuroplasticity	**Rat**, Neurons, Plasticity, Hippocampus, Brain, Central-Nervous-System, Synaptic Plasticity, Long-Term Potentiation, Food-Intake, Memory
	Long-Term Potentiation	**Long-Term Potentiation**, Synaptic Plasticity, Plasticity, Hippocampus, Nmda Receptor, Glutamate Receptors, Expression, Neurons, In-Vivo, Hippocampal-Neurons
	Genetic Association	**Association**, Susceptibility, Polymorphism, Linkage Disequilibrium, Disease, Major Histocompatibility Complex, Linkage, Complex Traits, Risk, Population
	Pre-Botzinger Complex	**Pre-Botzinger Complex**, In-Vitro, Prebotzinger Complex, Brain-Stem, Respiratory Rhythm Generation, Rhythm Generation, Rat, Control Of Breathing, Neurons, Pacemaker Neurons
	Prefrontal Cortex	**Performance**, Attention, Fmri, Children, Prefrontal Cortex, Brain, Working-Memory, Cortex, Memory, Activation
	Diabetes Mellitus	**Mellitus**, Glycemic Control, Complications, Hypertension, Randomized Controlled-Trial, Diabetes, Therapy, Risk, Diabetes Mellitus, Management
Chemistry: Spectroscopy	Crystal Structure	**Complexes**, Self-Organization, Crystal-Structure, Derivatives, Chemistry, Polymers, Behavior, Films, Nonlinear-Optical Properties, Phase-Transition
	Anodic Alumina	**Fabrication**, Arrays, Films, Anodic Alumina, Anodization, Self-Organization, Growth, Self-Organized Formation, Hexagonal Pore Arrays, Titanium
Soc	Soc	**Self-Organized Criticality**, Model, Dynamics, Econophysics, Evolution, Systems, Fluctuations, Models, Behavior, Turbulence
Ecosystems	Innovation Management	**Management**, Innovation, Economics, Performance, Model, Complexity, Systems, Technology, Firm, Knowledge
	Discrete-Event Systems	**Systems**, Supervisory Control, Petri Nets, Complexity, Discrete-Event Systems, Verification, Design, Automata, Discrete Event Systems, Synchronization
	Computational Complexity	**Complexity**, Algorithms, Computational Complexity, Algorithm, Networks, Optimization, Time, Systems,

		Search, Computational-Complexity
	Ecosystems	**Ecology**, Dynamics, Evolution, Biodiversity, Patterns, Diversity, Growth, Model, Management, Conservation
	Absorption	**Adsorption**, Sorption, Speciation, Complexation, Humic Substances, Water, Natural-Waters, Kinetics, Ph, Copper
	Cellular Automaton	**Cellular Automata**, Systems, Simulation, Self-Organization, Model, Cellular-Automata, Flow, Cellular-Automaton Model, Traffic Flow, Dynamics
	Multi-Agent Systems	**Systems**, Multi-Agent Systems, Multiagent Systems, Design, Agents, Architecture, Multi-Agent System, Framework, Model, Intelligent Agents
	Division Of Labor In Insect Societies	**Self-Organization**, Behavior, Division-Of-Labor, Hymenoptera, Ants, Colonies, Formicidae, Social Insects, Swarm Intelligence, Evolution
	Complex Adaptive Systems	**Complexity**, Self-Organization, Chaos, Emergence, Science, Complex Adaptive Systems, Complexity Theory
	Malaria	**Malaria**, Culicidae, Identification, Transmission, Complex, Diptera, Africa, Mosquitos, Anopheles-Gambiae Complex, Gambiae Complex
Neural Networks	Neural Networks	**Neural Networks**, Classification, Systems, Model, Self-Organizing Map, Neural Network, Algorithm, Identification, Artificial Neural Networks, Prediction
	Genetic Algorithm	**Optimization**, Genetic Algorithms, Genetic Algorithm, Design, Systems, Neural Networks, Model, Algorithm, Algorithms, Simulation
	Simulated Annealing	**Optimization**, Simulated Annealing, Algorithm, Model
	Gene Expression Patterns	**Patterns**, Self-Organizing Maps, Gene-Expression, Microarray, Identification, Gene Expression, Saccharomyces-Cerevisiae, Cancer, Expression, Classification
Complex Systems	Complex Systems	**Complex Networks**, Dynamics, Small-World Networks, Model, Internet, Networks, Evolution, Scale-Free Networks, Systems, Organization

Table B.5: Results of fuzzy detection: ten high frequent topic keywords contained by robust clusters. These high frequent topic keywords are contained in at least 20 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

[1]Only 10-15 articles contain them

Cluster	High frequent topic keywords
Soc	**Self-Organized Criticality**, Model, Systems, Earthquakes, 1/F Noise, Dynamics, Noise, Avalanches, Fluctuations, Criticality
Chemistry	**Complexes**, Crystal-Structure, Chemistry, Model, Phase-Transition, Derivatives, Systems, Spectroscopy, Kinetics, Molecular-Structure
Chaos Theory	**Chaos**, Dynamics, Systems, Model, Strange Attractors, Turbulence, Stability, Time-Series, Behavior, Models
Ising Model	**Ising-Model**, Complex Zeros, Model, Systems, Dynamics, Phase-Transition, Partition-Function, Behavior, Phase-Transitions
Neuroscience	**Rat**, Neurons, Brain, Complex, Cat, Ecology, Rat-Brain, Responses, Growth, Horseradish-Peroxidase
Neural Networks	**Neural Networks**, Systems, Model, Algorithm, Design, Optimization, System, Artificial Intelligence, Neural Network, Self-Organization
Mhc	**Major Histocompatibility Complex**, Expression, Activation, Monoclonal-Antibodies, Complex, Mice, Cells, T-Cells, Induction, Antigen
Arc	**Aids-Related Complex**, Aids, Performance, Human-Immunodeficiency-Virus, Aids Dementia Complex, Infection, Children, Complex, Disease, Brain
Molecular Biology: Fission Yeast And Saccharomyces Cerevisiae	**Fission Yeast**, Cell-Cycle, Saccharomyces-Cerevisiae, Expression, Phosphorylation, Protein-Kinase, Activation, Messenger-Rna, M-Phase, Mitosis
Dynamics Turbulence	**Flow**, Turbulence, Model, Boundary-Layer, Equations, Velocity, Simulation, Evolution, Diffusion
Molecular Biology: Protein	**Complex**, Binding, Expression, Proteins, Purification, Cells, Protein, Identification, Activation, Metabolism
Photosystems	**Photosystem-Ii**, Complex(, Chloroplasts, Photosynthesis, Escherichia-Coli) [1]
Molecular Biology: Gene	**Expression**, Protein, Gene, Messenger-Rna, Gene-Expression, Escherichia-Coli, Complex, Transcription, Sequence, Dna

Table B.6: Partition results in past history of complex system science during 1985-1994: ten high-frequent topic keywords contained by disjoint communities. These high frequent topic keywords are contained in at least 15 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

Cluster	High frequent topic keywords
Chaos Theory	**Chaos**, Systems, Dynamics, Model, Time-Series, Strange Attractors, Stability, Turbulence, Behavior, Self-Organization
Neural Networks	**Neural Networks**, Systems, Model, Algorithm, Optimization, Design, Neural Network, Complexity, Networks, Recognition
Molecular Biology: Serum And Hormone	**Serum**, Complex, Invitro, Factor-I, Igf-I, Cells, Granulosa-Cells, Fertilization, Maturation, Hormone
Ecosystems	**Ecology**, Growth, Evolution, Systems, Performance, Complexity, Model, Behavior, Dynamics, Patterns
Immunology	**Expression**, Major Histocompatibility Complex, Complex, Mice, Activation, Disease, Monoclonal-Antibodies, T-Cells, Cells, Tumor-Necrosis-Factor
Dynamics Turbulence	**Model**, Flow, Turbulence, Transport, Simulation, Evolution, Adsorption, Behavior, Flows, Boundary-Layer
Molecular Biology: Saccharomyces Cerevisiae And Fission Yeast	**Saccharomyces-Cerevisiae**, Cell-Cycle, Expression, Protein, S-Phase, Cell Cycle, Fission Yeast, Gene, Mitosis, Complex
Metabolic Control Analysis	**Control Coefficients**, Metabolic Control Analysis, Metabolism, Skeletal-Muscle
Chemistry	**Complexes**, Crystal-Structure, Chemistry, Derivatives, Systems, Model, Complex, Dynamics, Spectroscopy, Phase-Transition
Neuroscience	**Rat**, Brain, Neurons, Complex, Cat, Rat-Brain, Cells, Hippocampus, Long-Term Potentiation, Central-Nervous-System
Soc	**Self-Organized Criticality**, Model, Dynamics, Systems, Earthquakes, Avalanches, 1/F Noise, Evolution, Noise, Growth
Semiconductor Superlattice Materials And Growth Technology	**Gaas**, Growth, Molecular-Beam Epitaxy, Quantum Dots, Photoluminescence, Islands, Self-Organized Growth, Self-Organization, Surfaces, Ingaas
Molecular Biology	**Expression**, Complex, Protein, Binding, Gene, Cells, Messenger-Rna, Escherichia-Coli, Identification, Gene-Expression

Table B.7: Partition results in the past history of complex system science during 1990-1999: ten high-frequent topic keywords contained by disjoint communities. These high frequent topic keywords are contained in at least 15 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

Cluster	High frequent topic keywords
Chemistry	**Complexes**, Crystal-Structure, Self-Organization, Chemistry, Derivatives, Complex, Behavior, Polymers, Dynamics, Systems
Molecular Biology	**Expression**, Complex, Protein, Gene, Gene-Expression, Activation, Saccharomyces-Cerevisiae, Cells, Identification, Messenger-Rna
Neural Networks	**Neural Networks**, Model, Classification, Neural Network, Systems, Algorithm, Recognition, Artificial Neural Networks, Networks, Identification
Genetic Algorithm	**Systems**, Optimization, Design, Model, Complexity, Genetic Algorithms, Simulation, Cellular Automata, Algorithm, Models
Chaos Theory	**Chaos**, Dynamics, Systems, Model, Stability, Time-Series, Complexity, Nonlinear Dynamics, Synchronization, Behavior
Quantum Chaos	**Systems**, Model, Ising-Model, Localization, Chaos, Quantum Chaos, States, Fluctuations, Universality, Density
Neuroscience	**Brain**, Rat, Neurons, Long-Term Potentiation, Plasticity, Hippocampus, Memory, Central-Nervous-System, Synaptic Plasticity, Cortex
Complex Networks	**Complex Networks**, Internet, Small-World Networks, Dynamics, Model, Networks, Evolution, Systems, Escherichia-Coli, Organization
Computation Theory In Networks	**Systems**, Design, Performance, Algorithm, Stability, Optimization, Identification, Networks, Simulation, Model
Soc	**Self-Organized Criticality**, Model, Dynamics, Systems, Evolution, Avalanches, Econophysics, Earthquakes, Noise, Growth
Dfb	**Dfb Lasers**, Distributed Feedback Lasers, Semiconductor-Lasers
Timeless (Gene)	**Transcription**, Light, Rhythms, Suprachiasmatic Nucleus, Protein, Clock, Timeless, Expression, Drosophila, Circadian Clock
Dynamics Turbulence	**Model**, Turbulence, Flow, Simulation, Evolution, Transport, Behavior, Dynamics, Flows, Adsorption
Semiconductor Superlattice Materials And Growth Technology	**Growth**, Gaas, Molecular-Beam Epitaxy, Quantum Dots, Islands, Self-Organization, Photoluminescence, Self-Organized Growth, Surfaces, Films
Ecosystems	**Ecology**, Evolution, Model, Management, Dynamics, Complexity, Growth, Behavior, Patterns, Systems
Immunology	**Expression**, Major Histocompatibility Complex, Complex, Disease, Mice, Association, Identification, T-Cells, Infection, Activation

Table B.8: Partition results in the past history of complex system science during 1995-2004: ten high-frequent topic keywords contained by disjoint communities. These high frequent topic keywords are contained in at least 15 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

Cluster	High frequent topic keywords
Dynamics Turbulence	**Turbulence**, Model, Flow, Simulation, Dynamics, Behavior, Large-Eddy Simulation, Complex Terrain, Plasticity, Flows
Network Systems	**Systems**, Design, Performance, Algorithm, Channels, Synchronization, Optimization, Networks, Stability, Fading Channels
Molecular Biology	**Expression**, Complex, Gene-Expression, Protein, In-Vivo, Activation, Saccharomyces-Cerevisiae, Identification, Gene, Escherichia-Coli
Chaos Theory	**Chaos**, Dynamics, Systems, Model, Stability, Complexity, Synchronization, Time-Series, Nonlinear Dynamics, System
Biological Psychology(Behavioral Neuroscience)	**Brain**, Long-Term Potentiation, Association, Rat, Synaptic Plasticity, Neurons, Plasticity, Expression, Performance, Children
Spectroscopy	**Complexes**, Self-Organization, Crystal-Structure, Chemistry, Derivatives, Behavior, Polymers, Films, Systems, Spectroscopy
Soc	**Self-Organized Criticality**, Model, Dynamics, Econophysics, Evolution, Systems, Fluctuations, Models, Behavior, Turbulence
Ecosystems	**Ecology**, Systems, Model, Complexity, Evolution, Dynamics, Management, Growth, Behavior, Self-Organization
Semiconductor Superlattice Materials And Growth Technology	**Growth**, Self-Organization, Gaas, Quantum Dots, Islands, Molecular-Beam Epitaxy, Nanostructures, Surfaces, Films, Self-Organized Growth
Neural Networks	**Neural Networks**, Model, Systems, Classification, Algorithm, Optimization, Identification, Design, Neural Network, Models
Complex Networks	**Complex Networks**, Dynamics, Small-World Networks, Model, Internet, Networks, Evolution, Scale-Free Networks, Systems, Organization
Clinical Psychology	**Management**, Therapy, Radiation-Therapy, Radiotherapy, Trauma, Experience, Hemorrhage

Table B.9: Partition results in the past history of complex system science during 2000-2009: ten high-frequent topic keywords contained by disjoint communities. These high frequent topic keywords are contained in at least 20 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

Cluster	High frequent topic keywords	Involving communities
Pathophysiology: coagulation and fibrinolysis	**PLASMA**, COAGULATION, FIBRINOLYSIS, ACTIVATION, ASSAY, FIBRINOPEPTIDE-A, COMPLEX	MHC, ARC

Table B.10: Results of fuzzy detection during 1985-1994: ten high-frequent topic keywords contained by modular overlaps between pairs of communities. These high frequent topic keywords are contained in at least 15 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

Cluster	High frequent topic keywords	Involving communities
Chemical: Adsorption	**Adsorption**, Complexation, Speciation, Sorption, Cadmium, Copper, Ph, Natural-Waters, Transport, Zinc	Ecosystems, Molecular Biology: Serum And Hormone, Dynamics Turbulence
Industrial And Organizational Psychology	**Management**, Organizations, Model, Performance, Economics, Organization, Innovation, United-States, Industry, Complexity	Ecosystems, Molecular Biology: Serum And Hormone, Neuroscience
Malaria Transmission And Mosquito	**Malaria**, Culicidae, Transmission, Diptera, Complex, Identification, Anopheles-Gambiae Complex, West-Africa	Ecosystems, Molecular Biology: Serum And Hormone, Dynamics Turbulence
Protein Expression: Binding	**Binding**, Expression, Complex, Protein, Cells, Rat, Messenger-Rna, Escherichia-Coli, Purification, Phosphorylation	Metabolic Control Analysis, Genetics
Structural And Molecular Biology	**Escherichia-Coli**, Crystal-Structure, Resolution, Binding, Mechanism, Complex, 3-Dimensional Structure	Molecular Biology: Saccharomyces Cerevisiae And Fission Yeast, Genetics
Protein Expression: Protein	**Proteins**, Expression, Complex, Purification, Protein, Identification, Binding, Cells, Escherichia-Coli, Gene	Metabolic Control Analysis, Genetics
Cell Physiology: Cell Signaling	**Signal Transduction**, Map Kinase, Phosphorylation, Signal-Transduction, Activation, Ras, Tyrosine Phosphorylation, Activated Protein-Kinase, Epidermal Growth-Factor, Cells	Molecular Biology: Saccharomyces Cerevisiae And Fission Yeast, Genetics
Molecular Biology: Metabolism And Mitochondria	**Metabolism**, Mitochondria, Complex, Brain, Binding, Rat, Cells, Expression, Inhibition, Liver	Metabolic Control Analysis, Genetics

Table B.11: Results of fuzzy detection during 1990-1999: ten high-frequent topic keywords contained by modular overlaps between pairs of communities. These high frequent topic keywords are contained in at least 15 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

Cluster	High frequent topic keywords	Involving communities
Hydrogeology	**Porous-Media**, Solute Transport, Flow, Stochastic-Analysis, Hydraulic Conductivity, Dispersion, Transport, Groundwater-Flow, Groundwater, Simulation	Dynamic Turbulence, Ecosystems
Chemistry: Adsorption	**Adsorption**, Sorption, Complexation, Speciation, Natural-Waters, Copper, Water, Seawater, Humic Substances, Cadmium	Dynamic Turbulence, Ecosystems
Cellular Automata	**Cellular Automata**, Self-Organization, Simulation, Cellular-Automata, Systems, Traffic Flow, Cellular-Automaton Model, Model, Jams, Jamming Transition	Chaos Theory, Timeless (Gene)
Self-Organization	**Model**, Self-Organization, Dynamics, Systems, Patterns, Oscillations, System, Chaos, Pattern-Formation, Evolution	Chaos Theory, Timeless (Gene)
Dynamics	**Dynamics**, Systems, Pattern-Formation, Turbulence, Stability, Ginzburg-Landau Equation, Instability, Chaos, Waves, Transition	Chaos Theory, Timeless (Gene)
Self-Organization In Chemistry	**Self-Organization**, Particles, Nanoparticles, Superlattices, Clusters, Size, Monolayers, Films, Optical-Properties, Growth	Semiconductor Superlattice Materials And Growth Technology, Chemistry
Nitric Oxide Synthase	**Nitric Oxide**, Nitric-Oxide, L-Arginine, Relaxing Factor, Synthase, Inhibition, Cells, Nitric-Oxide Synthase, Endothelial-Cells, Endothelium	Molecular Biology, Neuroscience
Celluar Neural Networks	**Cellular Neural Networks**, Cnn, Cellular Neural Network, Chaos, Neural Networks	Chaos Theory, Timeless (Gene)

Table B.12: Results of fuzzy detection during 1995-2004: ten high-frequent topic keywords contained by modular overlaps between pairs of communities. These high frequent topic keywords are contained in at least 15 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

Cluster	High frequent topic keywords	Involving communities
Genetic Association	**Association**, Susceptibility, Polymorphism, Linkage Disequilibrium, Disease, Major Histocompatibility Complex, Linkage, Complex Traits, Risk, Population	Molecular Biology, Biological Psychology
Discrete-Event Systems	**Systems**, Supervisory Control, Petri Nets, Complexity, Discrete-Event Systems, Verification, Design, Automata, Synchronization, Discrete Event Systems	Computer Science, Ecosystems
Computational Complexity	**Complexity**, Algorithms, Computational Complexity, Algorithm, Networks, Optimization, Time, Systems, Search, Computational-Complexity	Computer Science, Ecosystems
Astronomy-Ism(Interstellar Medium)	**Turbulence**, Ism : Clouds, Star-Formation, Stars : Formation, Molecular Clouds, Ism : Structure, Ism : Kinematics And Dynamics, Evolution, Radio Lines : Ism, Intergalactic Medium	Dynamics Turbulence, Clinical Psychology
Multi-Agent Systems	**Systems**, Multi-Agent Systems, Multiagent Systems, Design, Agents, Architecture, Multi-Agent System, Framework, Model, Intelligent Agents	Computer Science, Ecosystems
Visual Cortex	**Complex Cells**, Lateral Geniculate-Nucleus, Cat Striate Cortex, Primary Visual-Cortex, Striate Cortex, Cortical-Neurons, Receptive-Fields, Contrast, Orientation Selectivity, Simple Cells	Biological Psychology, Systems Neuroscience

Table B.13: Results of fuzzy detection during 2000-2009: ten high-frequent topic keywords contained by modular overlaps between pairs of communities. These high frequent topic keywords are contained in at least 15 articles and are shown in order of descending frequency. The highest frequent topic keywords are shown in bold font.

Résumé substantiel

L'organisation modulaire de réseaux complexes

Les réseaux complexes sont obtenus par la modélisation de systèmes réels avec des graphes. Ce paradigme est utilisé pour représenter une grande variété de systèmes dans des domaines différents, tels que Internet, World Wide Web, les réseaux de co-citations, les réseaux de collaborations, les réseaux métaboliques. Chaque citoyen, tel qu'une personne, peut construire un réseau social dont les nœuds sont reliés par les interactions sociales(professionnelles, amicales).

Les études dans les réseaux complexes deviennent un intérêt populaire pour la recherche. Ces études ont été déclenchées par deux articles sur les réseaux petits mondes et les réseaux scale-free. Ces articles ont présenté des propriétés non-triviales, qui ne se produisent pas dans les réseaux simples tels que de graphes de treillis ou des graphes aléatoires. Cela a provoqué un grand développement des études sur les propriétés des réseaux réels.

L'analyse comparative et massive des réseaux de plusieurs domaines a produit une série de résultats inattendus et impressionnants. Une question importante qui a émergé est la compréhension des structures en communautés. Les études empiriques sur des réseaux différents tels que World Wide Web, les réseaux protéines complexes, les réseaux de messageries, *etc.* , mettent en évidence que leurs distributions de degrés ont des caractéristiques particulières. Des études ont également constaté que la distribution des degrés des nœuds n'est pas seulement qu'hétérogène sur l'ensemble du graphes globale, mais aussi l'est aussi localement. En d'autres mots, les réseaux peuvent être décrire par des communautés, avec des connexions denses en leur sein et des connexions éparses entre eux.

La structure de la communauté d'un réseau réel n'est pas seulement le résultat de la topologie, mais aussi se réfère aux fonctions du système : dans les réseaux de protéines complexes, les communautés correspondent à des fonctions spécifiques ; dans le World Wide Web, ils lient des pages par des thèmes ; dans les réseaux trophiques, ils correspondent à des compartiments, *etc.* . Les études sur la structure de communautés devraient conduire à une meilleure compréhension des systèmes complexes.

Détection de communautés

Afin de détecter des structures communautaires, diverses techniques sont proposées et sont appliquées à des réseaux réels. Dès 1955, Weiss et Jacobson ont réalisé la première analyse de structures communautaires, qui était à la base des algorithmes de partitionnement de graphe. Ces algorithmes de partitionnement de graphe divisent les noeuds en communautés prédéfinies, telles que le nombre d'arêtes entre les groupes est faible.

Dans un article fondateur paru en 2001, Girvan et Newman ont proposé un nouvel algorithme, qui a identifié les arêtes située entre les communautés pour l'élimination successive jusqu'à l'isolement des communautés. Ce document a déclenché une grande activité dans le domaine, et de nombreuses nouvelles méthodes modernes ont été proposées. Par exemple, l'optimisation de la modularité est la méthode la plus populaire pour la détection de communautés dans les grands graphes, les algorithmes dynamiques sont basés sur des techniques physiques : spin models, les marches aléatoires et la synchronisation, et d'autres, comme les méthodes basées sur l'inférence statistique : inférence bayésienne, blockmodeling, la sélection du modèle et la théorie de l'information.

Ces méthodes fournissent de bonnes performances dans la détection de communautés, et ont été appliquées à de réseaux réels pour leur analyse. La détection de communautés mérite-t-elle encore d'autres études plus approfondies ? Au moins deux raisons ont motivé notre travail en profondeur.

La première des raisons est que les réseaux réels complexes deviennent de plus en plus complexes. A partir de l'analyse de petits réseaux statiques nous pouvons nous attacher à l'étude de systèmes avec des milliers ou des millions de nœuds, en portant particulièrement notre attention sur les propriétés des réseaux dynamiques. Par exemple, le réseau des communications de millions d'utilisateurs change ses interactions au fil du temps. La structure d'un réseau réel est le résultat de l'évolution continue des interactions qui correspondent aux fonctions du système. De sorte que la recherche sur les communautés dans les réseaux dynamiques conduirait à une meilleure connaissance des mécanismes de l'évolution du système, et à une meilleure cottoning sur les comportements dynamiques et fonctionnels. La plupart des méthodes de détection de communautés sont proposées pour les réseaux statiques. Il y a un besoin crucial d'algorithmes qui détectent les communautés dans les réseaux dynamiques.

La seconde raison est que la structure de communautés recouvrantes est toujours un problème. La plupart des méthodes de la détection de communautés sont proposées pour détecter les communautés disjointes sans nœuds recouvrants. Les nœuds recouvrants sont partagés par plusieurs communautés qui se recouvrent dans la structure communautaire. Ils sont intéressants à étudier, car ils jouent un rôle clé en tant qu'intermédiaire entre les communautés, avec un effet spécial dans la prédiction de comportements dynamiques des individuals dans les réseaux. Des études dans les histoires d'interactions sociales entreprises en Hongrie ont montré qu'il est possible de recombiner les adhésions aux groupes communautaires des nœuds recouvrants. Au cours du temps, une communauté peut survivre à ses membres. Certaines communautés ont été construites par le fractionnement d'une communauté mère et d'autres par le regroupement de plusieurs communautés mères. Ce phénomène, en effet, représente une caractéristique essentielle de nœuds recouvrants dans la compréhension de l'organisation structurelle des systèmes complexes. Des études de communautés recouvrantes seront utiles pour comprendre les mécanismes des systèmes dynamiques et prévoir les tendances futures. Nous explorons cette thèse pour faire face à l'analyse de la structure de communautés recouvrantes dans des réseaux différents ou/et dans des réseaux dynamiques.

Pour ce faire, deux méthodes différentes de la détection de communautés recouvrantes sont proposées. Nous présentons aussi des approches pour suivre l'évolution de structures

au fil du temps. Pour vérifier nos méthodes, nous les avons appliquées à données réelles différentes. Les résultats obtenus sont évalués.

Les contributions

Les contributions principales de cette thèse sont brièvement résumées ci-dessous :

- Deux points de vue différents sur la détection de nœuds recouvrants : Pour détecter les communautés recouvrantes et caractériser les nœuds recouvrants, nous avons proposé deux définitions de nœuds recouvrants : les nœuds recouvrants granulaires et les groupes recouvrants. Chaque nœud recouvrant granulaire se connecte à plusieurs communautés avec une forte cohésion. Chaque groupe recouvrant est un ensemble de nœuds ayant un haut degré d'appartenance à une communauté (la force du groupe de noeuds appartient à la communauté) avec au moins deux communautés.
  Pour la détection de nœuds recouvrants granulaires, nous avons proposé l'optimisation de cliques, ce qui détecte les cliques k-adjacentes aux communautés (Une clique qui n'appartient pas à la communauté, mais partage au moins k - 1 nœuds communs). Un nœud recouvrant granulaire dans un sens faible est le membre d'une clique, qui est adjacent à d'autres communautés auxquelles il n'appartient pas dans la partition. Un nœud recouvrant granulaire dans un sens fort est l'élément d'une clique, qui est adjacente à au moins deux communautés en même temps.
  Notre étude a pour but d'identifier les groupes recouvrants. Nous proposons la méthode nommée détection floue (fuzzy detection). En exécutant de l'algorithme de Louvain à plusieurs reprises, nous pouvons calculer la probabilité qu'une paire de noeuds apparaissent dans une même communauté. Il nous permet de détecter des groupes robustes, qui ont une grande stabilité contre les chocs aléatoires. Chaque paire de noeuds connectés d'un groupe robuste a une haute probabilité de co-comparution. En outre, nous sommes en mesure de détecter les noyaux de la communauté et les groupes recouvrants. Le noyau de la communauté est le groupe robuste maximum dans une communauté. Le groupe recouvrant est un groupe robuste ayant une haute probabilité de co-comparution avec plusieurs communautés. Les applications de ces deux méthodes à des graphes de référence sont accord avec les communautés connues. Nous les appliquons également à un réseau réel. Dans les expériences, nous observons que les deux méthodes donnent les résultats significatives mais différentes pour caractériser des nœuds recouvrants.

- Suivi de l'évolution des communautés et identification de la dynamique des communautés : Pour suivre l'évolution des communautés et identifier la dynamique des communautés, nous avons proposé une méthode en deux étapes : tout d'abord, nous appliquons notre détection floue à la détection de la structure communautaire à chaque pas de temps, et ensuite nous établissons la relation entre les communautés à des pas de temps différents. Comme la définition de la persistance de groupe est utilisée, nous sommes en mesure de caractériser la dynamique des communautés, même si certaines parties de la composition fluctue.

Afin de mieux analyser et d'explorer la dynamique des communautés, nous avons introduit une technique de visualisation appelées diagrammes de lignage. Les diagrammes de la lignée nous permettent d'observer le degré de stabilité des communautés définies par leurs membres au fil du temps et comment les changements de structure évoluent dans les communautés. Cette approche a été appliquée à un réseau dynamique. Un avantage important de notre méthode est son efficacité dans la détection de la dynamique des communautés dans des réseaux évoluant au court du temps. Par conséquent, notre méthode est permet de détecter et d'analyser l'évolution des communautés dans les grands réseaux, en particulier, ceux qui présentent un nombre élevé de changements au fil du temps.

Aperçu de cette thèse

La thèse est organisée comme suit. Le chapitre 1 est l'enquête de la détection communautaire dans des réseaux dynamiques. Nous décrivons la définition de la structure communautaire et les changements des communautés au fil du temps. La détection de communautés dans des réseaux dynamiques devient une question populaire. Ce problème est très difficile à résoudre. Nous passons en revue les algorithmes conçus pour les réseaux dynamiques, qui sont basés sur les techniques de la détection de communautés dans les réseaux statiques. Nous discutons également des questions cruciales comme la façon dont les méthodes doivent être testées et comparées entre elles.

Le chapitre 2 concerne la détection de communautés recouvrantes. Nous discutons l'importance de la structure de communautés recouvrantes dans l'analyse des réseaux et les limites des algorithmes existants. Ensuite, nous transformons le problème de la détection de communautés recouvrantes au problème de la détection de nœuds recouvrants, avec la définition de nœuds recouvrants granulaires et la définition de groupes recouvrants. Par conséquent, nous avons proposé deux méthodes distinctes : l'optimisation de clique et la détection floue. La première consiste à détecter les nœuds recouvrants granulaires et la seconde vise à détecter des groupes recouvrants. Les applications de deux méthodes dans les réseaux synthétiques et les réseaux réels ont de bonnes performances. En particulier, des applications dans le réseau de l'histoire des systèmes complexes fournissent un résultat impressionnant : les deux méthodes fournissent des informations sur les relations entre les communautés, mais quelque peu différentes.

Dans le chapitre 3 nous considérons la structure des communautés recouvrantes dans les réseaux dynamiques et proposons une méthode basée sur nos travaux décrits précédemment. Les applications dans les réseaux dynamiques telles que l'histoire de la science des systèmes complexes, révèlent que les nœuds recouvrants sont importants pour les fonctions et les interactions structurelles entre les communautés.

Enfin, nous terminons dans le chapitre 4 en concluant notre travail en matière de détection communautés et la discussion sur les perspectives que soulève ce travail.