# Modular Structure of Complex Networks

by

Reihaneh Rabbany khorasgani

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

Doctor of Philosophy

Department of Computing Science

University of Alberta

# Abstract

Complex networks represent the relationships or interactions between entities in a complex system, such as biological interactions between proteins and genes, hyperlinks between web pages, co-authorships between research scholars. Although drawn from a wide range of domains, real-world networks exhibit similar structural properties and evolution patterns. A fundamental property of these networks is their tendency to organize according to an underlying modular structure, commonly referred to as clustering or community structure. This thesis focuses on comparing, quantifying, modeling, and utilizing this common structure in real-world networks.

First, it presents generalizations of well-established traditional clustering criteria and propose proper adaptations to make them applicable in the context of networks. This includes generalizations and extensions of 1) the well-known clustering validity criteria that quantify the goodness of a single clustering; and 2) clustering agreement measures that compare two clusterings of the same dataset. The former introduces a new set of measures for *quantifying* the goodness of a candid community structure, while the latter establishes a new family of clustering distances suitable for *comparing* two possible community structures of a given network. These adapted measures are useful in both defining and evaluating the communities in networks.

Second, it discusses generative network models and introduces an intuitive and flexible model for synthesizing modular networks that closely comply with the characteristics observed for real-world networks. This network synthesizer is particularly useful for generating benchmark datasets with built-in modular structure, which are used in evaluation of community detection algorithms.

Lastly, it investigates how the modular structure of networks can be utilized in different contexts. In particular, it focuses on an e-learning case study, where the network modules can effectively outline the collaboration groups of students, as well as the topics of their discussions; which is used to monitor the participation trends of students throughout an online course. Then, it examines the interplay between the attributes of nodes and their memberships in modules, and present how this interplay can be leveraged for predicting (missing) attribute values; where alternative modular structures are derived, each in better alignment with a given attribute.

# Preface

This thesis incorporates parts from previously published papers. I enumerate them below, link each one to its relevant chapter, and explain my contributions in that publication. Except for those in which I am the first author, which are my original work, conducted under supervision of Prof. Osmar Zaïane, and in collaboration with the cited co-authors. In other words, unless stated otherwise, I was the author responsible for formulating the problem, implementations, theoretical and experimental evaluations, and writing the paper.

1. Journal Articles
   - 1.1. A. Abnar, M. Takaffoli, R. Rabbany, and O. R. Zaïane, *"SSRM: structural social role mining for dynamic social networks"*, Social Network Analysis and Mining Journal (SNAM), 5(1): 1869-5450, Springer, Sep. 2015. I was involved in this paper from the early stages, contributing to the high-level discussions, formulating the problem, and writing the paper. Mentioned in Chapter 5.
   - 1.2. R. Rabbany, O. R. Zaïane, *"Generalization of Clustering Agreements and Distances for Overlapping Clusters and Network Communities"*, Data Mining and Knowledge Discovery (DAMI), 29(5): 1458-1485, Springer, Jul. 2015. Covered in Chapter 3.
   - 1.3. C. Largeron, P-N. Mougel, R. Rabbany, O. R. Zaïane, *"Generating Attributed Networks with Communities"*, Public Library of Science (PLoS ONE) 10(4), Apr. 2015. My main contributions in this paper was performing the literature review, and writing the related parts in the manuscript. Mentioned in Chapter 4.
   - 1.4. R. Rabbany, M. Takaffoli, J. Fagnan, O. R. Zaïane, and R. Campello, *"Communities Validity: Methodical Evaluation of Community Mining Algorithms"*, Social Network Analysis and Mining (SNAM), 3(4): 1039-1062, Springer, Oct. 2013. Covered in Chapter 2.

2. Refereed Book Chapters
   - 2.1. R. Rabbany, O. R. Zaïane, *"Evaluation of Community Mining Algorithms in the Presence of Attributes"*, Trends and Applications in Knowledge Discovery and Data Mining, LNCS 9441: 152-163, Springer, Nov. 2015. Covered in Chapter 5.
   - 2.2. R. Rabbany, S. ElAtia, M. Takaffoli, O. R. Zaïane, *"Collaborative Learning of Students in Online Discussion Forums: A Social Network Analysis Perspective"*, Educational Data Mining: Applications and Trends, in Studies in Computational Intelligence Series, 524: 441-466, Springer, Nov. 2014. Covered in Chapter 5.
   - 2.3. R. Rabbany, M. Takaffoli, J. Fagnan, O. R. Zaïane, and R. Campello, *"Relative Validity Criteria for Community Mining Algorithms"*, Encyclopedia of Social Network Analysis and Mining, 1562-1576, Springer, Oct. 2014. Covered in Chapter 2.

3. Conference and Workshop Papers
   - 3.1. R. Rabbany, O. R. Zaïane, *"Evaluation of Community Mining Algorithms in the Presence of Attributes"*, Proceedings of the 4th International Workshop on Quality issues, Measures of Interestingness and Evaluation of Data Mining Models (QIMIE) at the 19th Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD), May 2015. Covered in Chapter 5.
   - 3.2. J. Fagnan, R. Rabbany, M. Takaffoli, E. Verbeek, O. R. Zaïane, *"Community Dynamics: Event and Role Analysis in Social Network Analysis"*, Proceedings of the 10th International Conference on Advanced Data Mining and Applications (ADMA), LNCS 8933: 85-97, Dec 2014. Similar contribution as 1.2. Mentioned in Chapter 5.
   - 3.3. S. Ravanbakhsh, R. Rabbany, R. Greiner, *"Augmentative Message Passing for Traveling Salesman Problem and Graph Partitioning"*, Proceedings of the Advances in Neural Information Processing Sys-

tems (NIPS), pp. 289-297, Dec. 2014. In this paper, I was responsible for designing and conducting the experiments on the graph partitioning application. Mentioned in Chapter 2.

3.4. R. Rabbany, S. ElAtia, O. R. Zaïane, *"Mining Large Scale Data from National Educational Achievement Tests: A Case Study"*, Proceedings of the workshop on Data Mining for Educational Assessment and Feedback (ASSESS) at the ACM SIGKDD conference on Knowledge Discovery and Data Mining (KDD), Aug. 2014. Mentioned in Chapter 5.

3.5. M. Takaffoli, R. Rabbany, O. R. Zaïane, *"Community Evolution Prediction in Dynamic Social Networks"*, Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) , pp. 9-16, Aug. 2014. I was involved in the discussions on the evaluation strategies, designing the experiments and plotting the results, as well as writing the paper. Mentioned in Chapter 5.

3.6. A. Abnar, M. Takaffoli, R. Rabbany, O. R. Zaïane, *"SSRM: Structural Social Role Mining for Dynamic Social Networks"*, Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 289-296, Aug. 2014. Similar contribution as 1.2. Mentioned in Chapter 5.

3.7. M. Takaffoli, R. Rabbany, O. R. Zaïane, *"Incremental Local Community Identification in Dynamic Social Networks"*, Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 90-94, Aug. 2013. I was involved in the high-level discussions on how to formulate and evaluate the method, as well as writing the paper. Mentioned in Chapter 5.

3.8. R. Rabbany, M. Takaffoli, J. Fagnan, O. R. Zaïane, and R. Campello , *"Relative Validity Criteria for Community Mining Algorithms"*, Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 258-265, Aug. 2012. Covered in Chapter 2.

3.9. R. Rabbany and O. R. Zaïane,*"A Diffusion of Innovation-Based Closeness Measure for Network Associations"*, Proceedings of the Workshop on Mining Communities and People Recommenders (COMMPER) at the IEEE International Conference on Data Mining (ICDM), pp. 381-388, Dec. 2011. Mentioned in Chapter 2.

3.10. R. Rabbany , E. Stroulia and O. R. Zaïane,*"Web Service Matching for RESTfulWeb Services"*, Proceedings of the 13th IEEE International Symposium on Web Systems Evolution (WSE), pp. 115-124, Sep. 2011. Mentioned in Chapter 5.

3.11. R. Rabbany, M. Takaffoli and O. R. Zaïane,*"Analyzing Participation of Students in Online Courses Using Social Network Analysis Techniques"*, Proceedings of the 4th International Conference on Educational Data Mining (EDM), pp. 21-30, Jul. 2011. Mentioned in Chapter 5.

4. Invited Articles

4.1. R. Rabbany, M. Takaffoli and O. R. Zaïane, *"Social Network Analysis and Mining to Support the Assessment of Online Student Participation"*, ACM SIGKDD Explorations Newsletter, 13 (2): 20-29, 2011. Mentioned in Chapter 5.

5. Under Review

5.1. R. Rabbany, O. R. Zaïane, *"Clustering Agreement Index for Disjoint and Overlapping Clusters"*. Covered in Chapter 3.

5.2. J. Fagnan, A. Abnar, R. Rabbany, O. R. Zaïane, *"FARZ: Benchmarks for Community Detection Algorithms"*. Covered in Chapter 4.

To my mother,

and the memories of

my grandmother, and my great-grandmother.

# Acknowledgements

I feel very fortunate for having many wonderful people around me during the course of my PhD. First and foremost, I am grateful to my supervisor, Osmar Zaïane, for his invaluable insights and constant support. He perfectly guided me through my graduate research, by initiating promising research problems, giving me enough freedom, and also timely nudges when I was well off-track. I feel very lucky to have worked with him, not only because he was a great academic supervisor, but also because of his remarkable personality, and to have him as a mentor and role model.

I would like to extend my gratitude to my supervisory committee: Dale Schuurmans, Mohammad Salavatipour, and also Jörg Sander, and my external examiner, Martin Ester, for the valuable discussions we had in my committee meetings.

I am further thankful to my two external collaborators: Ricardo Campello and Samira ElAtia; as well as my amazing teammates from the Meerkat project: Eric Verbeek, Justin Fagnan, Matt Gallivan, Xiaoxiao Li, Afra Abnar, and Mansoureh Takaffoli. In particular, I like to acknowledge many enjoyable collaborations I had with Mansoureh, whom I consider as my academic sister.

I would also like to thank Alberta Innovates Center for Machine Learning (AICML) for funding our project and my graduate research; with special thanks to Leslie Acker for her warm smile and her hassle free help with the administrative tasks.

On a personal level, I am thankful to my caring and supportive friends: Bahar Aameri, Aida Nematzadeh, Nezam Bozorgzadeh, Amin Tootoonchian, Fatemeh Miri, Meysam bastani, Yavar Naddaf, Niousha Bolandzadeh, Pirooz chubak, Arezoo Elion, Amir Afshar, Maysam Heydari, Fariba Mahdavifard, Arash Afkanpour, Mariana Eret, Saman Vaisipour, Sadaf Matinkhoo, Hootan Nakhost, Farzaneh Mirzazadeh, Babak Behsaz, Neda Mirian, Babak Bostan, Amin Jorati, AmirAli Sharifi, Kiana Hajebi, Shiva Shams, Saina Lajevardi, Farzaneh Saba, Katayoon Navabi, Amir-massoud Farahmand, Parisa Naeimi, and many more. I am particularity grateful for the long lasting friendships of Bahar and Aida, thank you both for always being there for me. I would also like to thank Stuart and Maria Embleton for having me at their lovely home for many sushi and game nights; and also Sharon and Kent for joining in and making it more fun.

Last but not the least, I wish to express my deep gratitude to my parents, and my two younger sisters, Mahnaz and Parisa, for their love and their confidence in me; as well as my husband, Siamak, for his constant encouragement and support.

# Contents

# List of tables

# List of figures

# List of Acronyms and Glossary

**Complex network** models relationships in data usually represented by a graph.

**Small-world networks** have a diameter relatively smaller than their size.

**Scale-free networks** have a heavy-tail degree distribution.

**Modular networks** have regions of densely connected nodes.

**Clustering coefficient** measures the portion of connected neighbours, *i.e.*, closed triangles.

**Assortative mixing** similar nodes have a higher probability of being linked.

**Preferential attachment** probability of getting new links is proportional to the current degrees.

**Community structure** clustering or modular structure underlying a modular network.

**Modularity Q** quantifies the goodness of a given community structure.

**Internal evaluation** matches a clustering with the structure of data.

**Relative evaluation** ranks different clusterings of the same dataset.

**External evaluation** compares a clustering with the ground-truth.

**NMI (Normalized Mutual Information)** measures similarity of two clusterings.

**ARI (Adjusted Rand Index)** measures similarity of two clusterings based on pair counts.

# Chapter 1

# Introduction

Networks model the relationships in complex systems, such as biological interactions between proteins and genes, hyperlinks between web pages, co-authorships between research scholars, friendships between people, co-purchases between products, and many more. Although drawn from a wide range of domains, the real-world networks exhibit similar properties, such as small diameter, and heavy tail degree distribution [103]; as well as similar evolution patterns, such as shrinking diameter, and densification power laws [84]. Figure 1.1 illustrates the four basic characteristics observed for a typical random graph sample, and two real world instances.

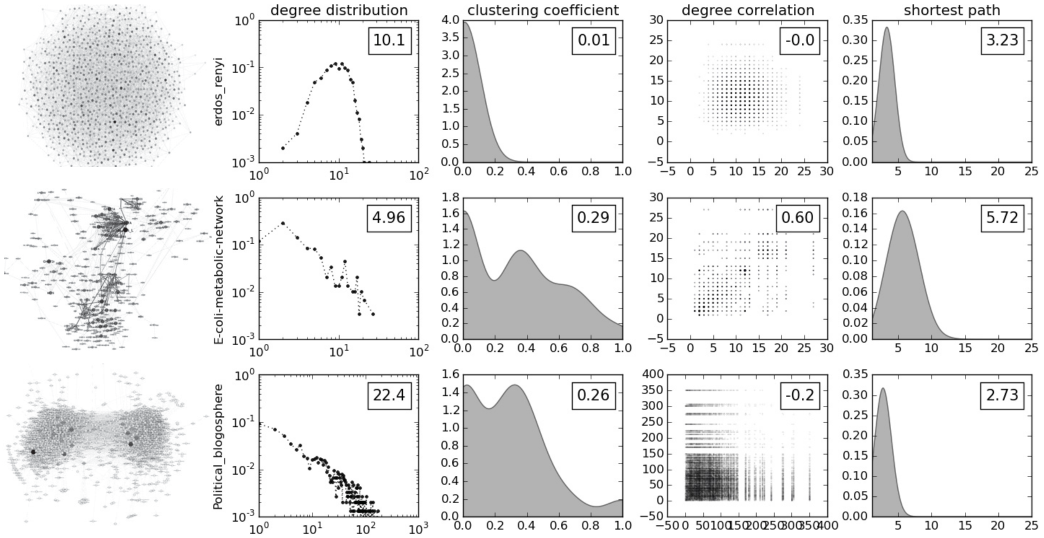
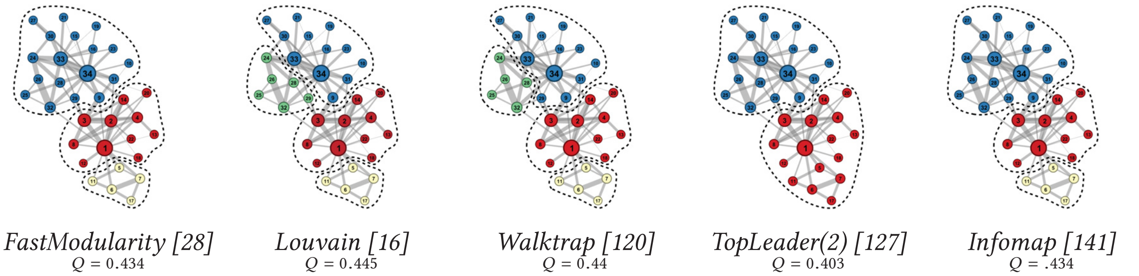


*__Figure 1.1:__ Properties of a random Erdős and Rényi [45] graph, a real-world biological network, and a real-world social network. All three have a small diameter (small-world), however unlike the random network, the real-world networks also have power law degree distribution (scale-free), relatively high transitivity (clustering coefficient), and degree correlation between connecting nodes ((dis-)assortative mixing).*

## 1.1 Problem Definition and Motivation

One fundamental property of the real world networks is that they tend to organize according to an *underlying modular structure*, commonly referred to as clustering or community structure [107]. Analyzing this structure provides insights into the mesoscopic characteristics of these networks. Therefore, module identification in networks, a.k.a. community detection, has been applied in wide range of domains, including biology, marketing, epidemiology, sociology, criminology, zoology, etc. For example in biology, the study of the modular structure in metabolic network of Homo sapiens [172] revealed that there are core modules that perform the basic metabolism functions and behave cohesively in evolution, and periphery modules that only interact with few other modules and accomplish specialized functions, which have a higher tendency to be gained/lost together through the evolution. As another biological example, the discovered modules in the yeast protein-protein interaction network studied in [147], are shown to outline the protein complexes (proteins that interact to carry out a task as a single complex unit, *e.g.*, RNA splicing), and dynamic functional units (proteins that bind at different time to participate in a cellular process, *e.g.*, communicating a signal from the surface of the cell to the nucleus). We expand this discussion on the applications of community detection in Chapter 5.



*FastModularity [28]*
$Q = 0.434$

*Louvain [16]*
$Q = 0.445$

*Walktrap [120]*
$Q = 0.44$

*TopLeader(2) [127]*
$Q = 0.403$

*Infomap [141]*
$Q = .434$

***Figure 1.2:*** *Modular structure of a classic dataset (Zachary's Karate Club), discovered by five different community detection algorithms. Colours correspond to the discovered modules, a.k.a. communities or clusters.*

The problem of finding the modular structure of networks is not well-defined. Many algorithms have been proposed to detect communities in a given network; whereas a community is loosely defined as a group of nodes that have relatively more links between themselves than to the rest of the network. The most common implementations consider a community as a group of nodes that (i) the number of links between them is more than chance [16, 28]; (ii) within them a random walk is more likely to trap [120]; (iii) have structural similarity [165]; (iv) follow the same leader node [127]; (v) coding based on them gives efficient compression of the graph [139, 141]; (vi) are separated from the rest by minimum cut, or conductance [88].

Different community mining algorithms discover communities from different perspectives (see Figure 1.2 for an example), outperform each-others in different settings, and have different computational complexities [47]. Therefore, an important (and less explored) research direction is how to evaluate and compare different community mining algorithms. The general theme of this thesis is the *evaluation of community detection algorithms*, *i.e.*, it studies different evaluation practices, objective criteria, comparison measures, and benchmark models for the community detection task. Although of significant importance on its own, one should note that the findings and conclusions presented in this thesis have a much broader impact than the evaluation; since there is a congruence relation between defining communities and evaluating community mining results. For instance, the well-known modularity Q by Newman and Girvan [108] which is commonly optimized as an objective function for the community detection task (*e.g.*, in the two biological studies mentioned earlier *i.e.*, [56, 147]), was originally proposed for quantifying the goodness of the community structure, and is still used for evaluating the community detection algorithms [28, 139].

## 1.2 Thesis Statements and Organization

This thesis starts with categorizing possible evaluation practices for community detection task, into internal, relative, and external evaluation approaches; which are discussed in Chapter 2. The thesis hypothesis here is that:

*Thesis Statement* 1. *The evaluation practices for the community detection task can be categorized, using the same classification from the traditional clustering literature, into internal, relative, and external evaluation.*

The *internal evaluation* practice measures the significance of the matching between the clustering structure produced by an algorithm and the underlying structure of the data. The aforementioned modularity Q falls within this category. Similarly, a relative evaluation criteria quantifies and compares different clustering solutions of the same dataset. The *external evaluation* practice, on the other hand, validates a community detection algorithm by comparing its results against the known ground-truth in benchmark datasets [38, 76].

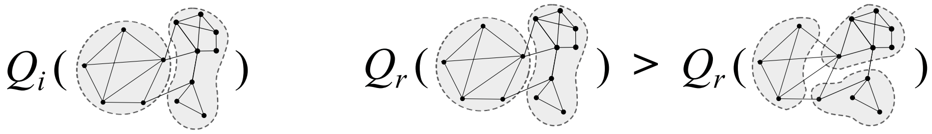


**Figure 1.3:** *Internal ($Q_i$) and Relative ($Q_r$) Quality Functions*

The external evaluation is the most common practice in the community mining evaluation. The relative evaluation, on the other hand, is less explored, although many relative clustering criteria exist in the traditional clustering literature. Hence the second thesis hypothesis is:

*Thesis Statement* 2. *The relative clustering criteria could be adapted for quantifying network communities, by generalizing them to use graph distances, and an appropriate notion of center.*

Chapter 2 introduces an extensive set of general quality functions for the internal and relative evaluation of community detection algorithms; see Figure 1.3 for an abstract illustration. These quality functions or criteria are adapted from the clustering literature, examples are: Variance Ratio Criterion, Silhouette Width Criterion, Dunn index, etc. These criteria are compared experimentally, and different factors which affect their performance are studied, including the hardness of the problem. To summarize, this chapter compares alternative measures which *quantify the goodness of community structure in networks*, concludes that their performance ranking depends on the experimental settings, and emphasizes that choosing the relative or internal evaluation criterion encompasses the same non-triviality and difficulty as of the community mining task itself. The results from this chapter are published in [130, 131, 133].

Two non-trivial factors which affect the results in Chapter 2 are the choices of the clustering agreement measure and benchmark datasets, which are used to assess the performance of the quality functions. This motivates the next two chapters of this thesis, *i.e.*, Chapter 3 and Chapter 4, which look deeper into each of these aspects. In more details, the following thesis statements are addressed in these two chapters, respectively.

*Thesis Statement* 3. *The clustering agreement measures could be adapted for comparing network communities, by generalizing them to incorporate the overlaps and structure in the data.*

*Thesis Statement* 4. *The external evaluation of community detection can be improved, by using realistic generative models for synthesizing benchmarks which comply with the characteristics and evolution patterns of real-world networks.*

Chapter 3 is focused on clustering agreement indexes which measures the similarity between two given clusterings (see Figure 1.4). The clustering agreement indexes are used mainly in the external evaluation, to compare the clustering results with the ground-truth. This chapter introduces novel generalizations of the well-known clustering agreement measures, and introduces a family of clustering agreement indexes, which can be used to derive new indexes. In particular, overlapping variations of the clustering agreement indexes are derived using this generalization, which are applicable to the general cases of clustering and are not constrained to the disjoint clusterings.

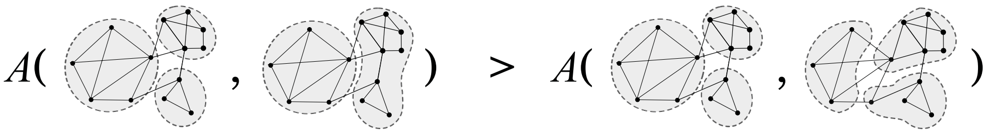


***Figure 1.4:*** *Agreement Measure in External Evaluation*

Chapter 3 further highlights that the clustering agreement indexes only compare memberships of data-points in the two clusterings, and overlook any relations between the data-points or any attributes associated with them. It then discusses the effect of neglecting these relations, *i.e.*, links in the networks, and derives extensions of the clustering agreement measures which incorporate the structure of the data when measuring the agreements between communities. This chapter has been published in [123].

The external evaluation is not applicable in real-world networks, as the ground-truth is not available. However, we assume that the performance of an algorithm on the benchmark datasets is a predictor of its performance on real networks. On the other hand, there are few and typically small real world benchmarks with known communities available for external evaluation; therefore the external evaluation is usually performed on synthetic benchmarks or on large networks with explicit or predefined communities, which are discussed respectively in Chapter 4 and Chapter 5.

Chapter 4 first studies different ways to improve the common generator models which are used for synthesizing benchmarks for community detection task. Then, it presents a realistic and flexible benchmark generator, called FARZ, which models modular networks, and incorporates intuitive parameters with meaningful interpretation that are easy to tune to control the experimental settings. Figure 1.5 visualizes example modular networks generated by this model. Common community detection algorithms are ranked from different perspectives using the FARZ benchmarks, and the resulted rankings are significantly different from the rankings obtained from the previous unrealistic benchmark networks. This new model, hence, enables a more thorough comparison of community detection algorithms. This work is submitted and currently under review.

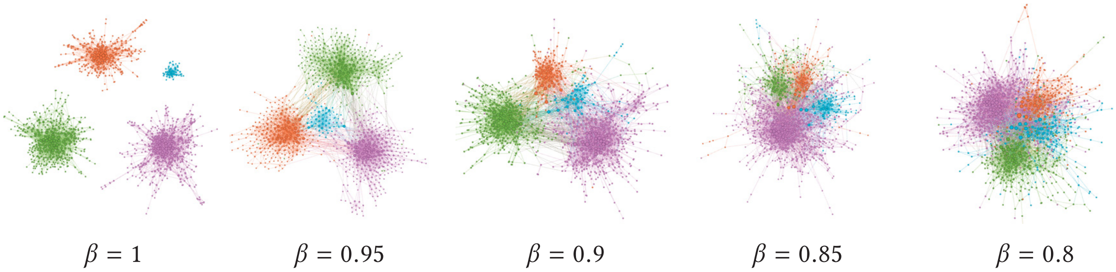
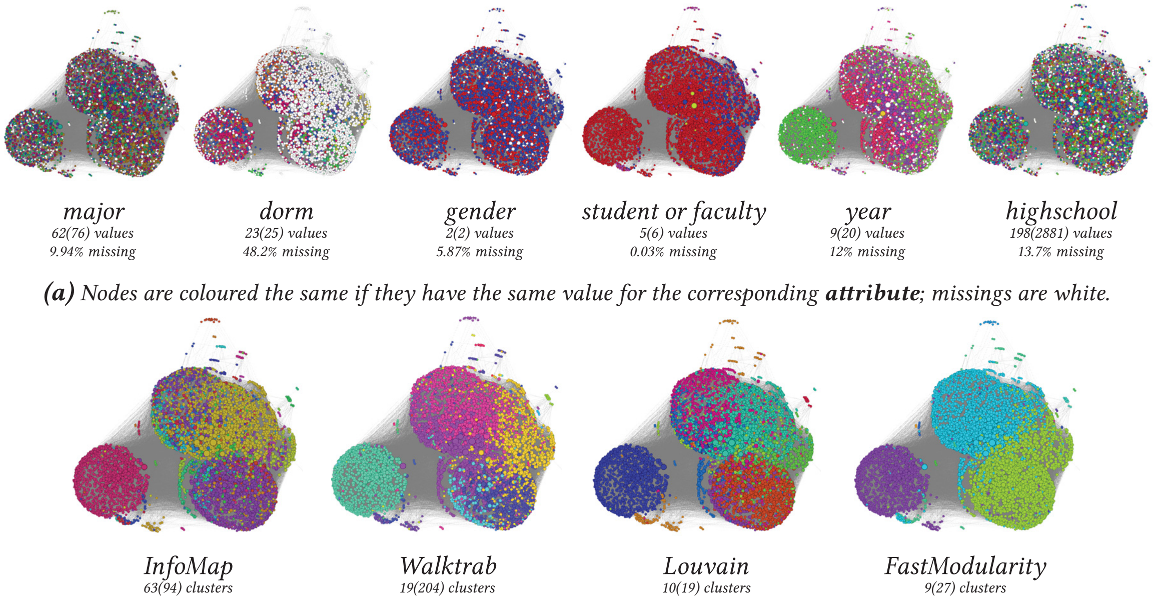


***Figure 1.5:*** *Example of networks generated by FARZ, with varying strength of the community structure.*

Alternative to generating benchmarks for the community detection task, large real world benchmarks are often used where the ground-truth communities are defined based on the explicit properties/attributes of the nodes. For instance in a collaboration network of authors obtained form DBLP, venues are considered as the ground-truth communities, or in the Amazon product co-purchasing network, product categories are considered as the ground-truth [166]. In general, there exists an interplay between the characteristics of nodes and the structure of the networks [33, 75], and in some contexts attributes or characteristics of nodes act as the primary organizing principle of the underlying communities [152]. However, this notion of ground-truth communities is imperfect and incomplete [83]. Chapter 5 discusses this in depth, and suggests to treat these attributes as another source of information. Hence, the last thesis statement considered here is:

*Thesis Statement* 5. *The community structure of a network is correlated with different attributes associated to the nodes in that network, and this correlation can be utilized to guide a community detection algorithm to find a community perspective that best corresponds with each given attribute.*

In particular, Chapter 5 utilizes the attributes associated to the nodes in the given network to guide a community detection algorithm, *i.e.*, to refine the communities and tune parameters, which is referred to as *community guidance by attributes*. Using this approach, different high quality community perspectives can be discovered where each best correspond with the selected set of attributes. The results of this chapter are published in [126]. Figure 1.6 visualizes the correlation between different attributes and different community results in an example dataset.



***(a)*** *Nodes are coloured the same if they have the same value for the corresponding* ***attribute****; missings are white.*

***(b)*** *Nodes are coloured the same if they belong to the same* ***community*** *in the results of corresponding algorithm.*

***Figure 1.6:*** *Correlations between attributes and communities for the American75 dataset from* Facebook 100 dataset *[153]. This network has 6386 nodes and 217662 friendships edges.*

# Chapter 2

# Quantifying Modular Structure of Networks

This chapter investigates different clustering quality criteria applied for relative and internal evaluation of clustering data points with attributes, and incorporates proper adaptations to make them applicable in the context of interrelated data. The adopted measures quantify a given community/modular structure of the network, which are useful in both defining and evaluating communities. The performances of the proposed adapted criteria are compared through an extensive set of experiments focusing on the evaluation of community mining results in different settings. The results from this chapter are published in [130, 131, 133].

## 2.1 Introduction

The recent growing trend in the Data Mining field is the analysis of structured/interrelated data, motivated by the natural presence of relationships between data points in a variety of present-day applications. The structures in these interrelated data are typically modeled by a graph of interconnected nodes, known as complex networks or information networks. Examples of such networks are hyperlink networks of web pages, citation or collaboration networks of scholars, biological networks of genes or proteins, trust and social networks of humans among others. These networks exhibit common statistical and structural properties (see Chapter 4 for more details), including having an underlying modular structure, which consists of regions of densely connected nodes, known as communities. Discovering this modular structure, commonly referred to as network clustering or community mining, is one of the principal tasks in the analysis of complex networks. The community mining algorithms evolved from simple heuristic approaches to more sophisticated optimization based methods that are explicitly or implicitly trying to maximize the goodness of the discovered communities. Although there have been many methods proposed for community mining, little research has been done to explore the evaluation and validation methodologies. Similar to the well-studied clustering validity methods in the Machine Learning field, we can consider three classes of approaches to evaluate community mining algorithms: external, internal and relative evaluation. The first two are statistical tests that measure the degree to which a clustering confirms a-priori specified scheme. The third approach compares and ranks clusterings of a same dataset discovered by different parameter settings [60]. In this chapter, we investigate the evaluation approaches for the community mining algorithms considering the same classification framework. We classify the common community mining evaluation practices into external, internal and relative approaches, and further extend these by introducing a new set of criteria adapted from the clustering literature. More specifically, these evaluation approaches are defined based on different clustering validity criteria. We propose proper adaptions that these measures require to handle comparison of community mining results. These criteria not only can be used as means to measure the goodness of discovered communities, but also as objective functions to detect communities.

The remainder of this chapter is organized as follows. In the next section, we first present some background, where we briefly introduce the well-known community mining algorithms, and the related work regarding evaluation of these algorithms. We continue the background with an elaboration on the three classes of evaluation approaches incorporating the common evaluation practices. In the subsequent section, we overview the clustering validity criteria, and introduce our proposed generalizations and adaptions of these measures for the context of interrelated data. Then, we extensively compare and discuss the performance of these adapted validity criteria through a set of carefully designed experiments on real and synthetic networks. Finally, we conclude with a brief analysis of the results.

## 2.2 Background and Related Works

A community is roughly defined as a group of “densely connected" nodes that are “loosely connected” to others outside their group. Different community detection algorithms have different interpretations for this definition. Basic heuristic approaches detect communities by assuming a set of heuristics by which the network divides naturally into some subgroups. For instance, the Clique Percolation Method [117] finds groups of nodes that can be reached via chains of k-cliques. The more recent optimization based approaches mine communities by defining and maximizing the overall “goodness” of the result. This optimization is often computationally expensive, which itself calls for different approximation algorithms. For example, the optimization of the infamous modularity Q [108] is proved to be $NP$-hard [17]; and several community detection algorithms have been proposed which optimize modularity Q as their objective [16, 26, 91, 104, 135]. Here, we first briefly overview the most well-known algorithms for discovering communities, then we review and classify the common practices for the evaluation of community mining algorithms.

### 2.2.1 Overview of Community Detection Methods

The most notable community mining method is the divisive hierarchical clustering of Girvan and Newman [51], which repeatedly removes the edge with the highest betweenness (often measured as the number of pairwise shortest paths that pass through an edge) from the networks, and constructs a dendrogram as the output. The modularity Q is proposed [108] to determine where to cut this dendrogram to get a sensible (flat) community structure. To put simply, modularity Q measures the difference between the fraction of edges that are within the communities and the expected such fraction if the edges were randomly distributed, *i.e.*, when the degree of nodes are fixed and the community structure is ignored. More formally, we have:

$$Q = \sum_{i} (e_{ii} - a_i^2) \tag{2.1}$$

where $e_{ij}$ denotes the fraction of edges with one endpoint in community $i$ and the other in community $j$; and $a_i = \sum_j e_{ij}$. In a later work, Newman [105] directly optimizes the modularity Q in an agglomerative hierarchical clustering algorithm. This greedy optimization starts by putting all nodes in their own community; then, repeatedly merges communities that result in the highest gain in the modularity Q , which is computed as $\Delta Q = 2(e_{ij} - a_i a_j)$ for communities $i$ and $j$.

The well-known FastModularity method [29], is an efficient heap based implementation of this algorithm, which only keeps track of the $\Delta Q$ matrix, hence reduces the time complexity of the original algorithm from $O(n(m+n))$ to $O(m \log^2 n)$; where $n$ and $m$ denote the number of nodes and edges in the graph, respectively. Blondel et al. [16] point out that the agglomerative method tends to produce super-communities, *i.e.*, communities that include a large fraction of the nodes in the network. As an alternative, they propose the Louvain method to optimize the modularity Q [16], which is highly scalable and one of the best performing community detection methods. Louvain starts with considering every node as a singleton community, and then iterates over all nodes, and moves each node to a community that results in the largest increase in the modularity Q . In more detail, we can rewrite Equation 2.1 as:

$$Q = \sum_{i} \sum_{u,v \in i} (w_{uv} - w_{u.} w_{v.}) \tag{2.2}$$

where $w_{uv}$ denotes the normalized weight of the edge from node $u$ to node $v$, and $w_{u.} = \sum_v w_{uv}$ *i.e.*, the weighted degree of $u$. Then, the gain of adding node $u$ to community $i$ is computed as $\Delta Q = 2 \sum_{v \in i} (w_{uv} - w_{u.} w_{v.})$. The efficiency of the Louvain algorithm is rooted in the simplicity of this formula. Using this formula, nodes are considered repeatedly until there is no such movement that increases the modularity, *i.e.*, a local maximum is reached. Then the resulted communities are aggregated to construct a new network; in which each community is a node, the edges between nodes are the sum of the edges between the members of their corresponding communities, and the sum of the edges within each community forms a self-loop. The above process repeats on the aggregated network, until there is no increase in the modularity, and hence the end result has a hierarchical structure. This structure is favourable since modularity Q is shown to have a resolution limit [48], *i.e.*, it tends to merge small (relative to the size of the overall network) communities into bigger modules. Fortunato and Barthélemy [48] show that, for an extreme instance, even merging two cliques connected with only one edge would increase the modularity Q , if they each have less than $\sqrt{m/2}$ edges; and more generally, modularity Q is biased against the communities with smaller than $\sqrt{2m}$ edges. Another well-known modularity optimization is based on the simulated annealing [57, 137] when community detection is modeled as finding the ground state of a spin system. In more detail, the module to which node $u$ belongs to is denoted by a variable $\sigma_u$, which represents a spin state in a spin glass, or Potts model, the energy of which is derived in [137] as a simplified Hamiltonian, with couplings of $J_{uv} = A_{uv} - \gamma p_{uv}$, as:

$$\mathcal{H}(\{\sigma\}) = -\sum_{u \neq v} (A_{uv} - \gamma p_{uv}) \delta(\sigma_u, \sigma_v) \tag{2.3}$$

where $A$ is the adjacency matrix, $p_{uv}$ is defined by the null model, $\gamma$ balances the effect of internal links/nonlinks, and $\delta$ is the Kronecker delta. The modularity Q of Equation 2.1 is a special case of this formula, *i.e.*, $Q = -\frac{1}{m}\mathcal{H}(\{\sigma\})$ when $\gamma = 1$ ("natural partition"), and $p_{uv} = \frac{A_{u.}A_{v.}}{2m}$; where $A_{u.}$ denotes the degree of node $u$. Hence modularity Q is maximized by finding a spin configuration that minimizes this Hamiltonian, a.k.a. its ground state. This minimization is performed using simulated annealing and based on the local update rules derived from the change in energy when a spin changes. More importantly, these studies [57, 137] highlight the fact that high modularity Q does not always indicate a community structure, and the need to examine the statistical significance of the obtained modularity Q . In particular, Guimera et al. [57], show how to obtain partitions with high modularity Q in random graphs, both Erdős and Rényi [45] and Albert and Barabási [6] scale-free models, which by definition do not have a modular structure. There are also spectral optimization techniques proposed for the modularity matrix [102, 105, 106]. In particular, the $n \times n$ modularity matrix $B$ is defined as $B_{uv} = A_{uv} - \frac{1}{2m} A_{u.} A_{v.}$; from which the modularity Q for when there are only two communities can be rewritten as:

$$Q = \frac{1}{4m} \sum_{uv} B_{uv} \sigma_u \sigma_v \tag{2.4}$$

where $\sigma_u$ is either +1 or $-1$, which indicates the membership of node $u$ in the two communities. Newman [102] shows that a relaxed version (*i.e.*, $\sigma_u \in [-\sqrt{n}, \sqrt{n}]$) of this problem could be solved by finding the second-highest eigenvalue for the normalized Laplacian of the network, *i.e.*, $L = D^{-1/2} A D^{-1/2}$; hence showing that for the bipartitioning case, the modularity maximization is similar to the normalized-cut graph partitioning. There also exists a whole body of community detection methods which are not based on optimizing modularity Q . We previously proposed a k-medoid based community mining approach, called TopLeaders [127]. TopLeaders (implicitly) maximizes the overall closeness of followers and leaders, assuming that a community is a set of followers congregating around a potential leader. A closely related family of methods are based on label propagation [15, 42, 134]. For instance, Raghavan et al. [134] consider a label for each node that denotes its community. Then these labels are propagated iteratively, where in each step a node chooses to join the community of majority of its neighbours. The “Chinese whisper” algorithm [15] is a similar approach proposed for the graph partitioning. Another notable family of methods mines communities by utilizing information theory concepts such as compression by Rosvall and Bergstrom [140], and entropy by Kenley and Cho [68]. For instance, the Infomap method proposed in [140] finds communities that if the network is coded based on them, one can optimally describe any random walk. Their objective for “goodness” of communities is defined in terms of the Shannon entropy of the random walk within and between the clusters. In more detail, they derive the following map equation which measures the average number of bits per step to describe a random walk on the given partitioned network.

$$L = \sum_i q_i \log(\sum_i q_i) - 2\sum_i q_i \log(q_i) - \sum_u p_u \log(p_u) + \sum_i (q_i + \sum_{u\in i} p_u) \log(q_i + \sum_{u \in i} p_u) \quad (2.5)$$

where $p_u$ is the ergodic node visit frequency computed for node $u$, by a random surfer which uses teleportation, and $q_i$ is the exit probability of module $i$ which is derived from the node visit frequencies; refer to [140] for more details. Different from the methods mentioned earlier, Ahn et al. [4] propose a community detection algorithm which groups edges instead of nodes. They define a similarity measure between edges, based on the neighbourhood overlap of their incident nodes, and use a single-linkage hierarchical algorithm to derive a clustering dendrogram; which is then cut where the average density of modules is maximized. Their method can put a node into different clusters, and hence generates overlapping communities. Finding overlapping communities is in fact one of the main extensions of the community detection problem [55, 81, 94, 167]. Other notable extensions are local[25, 91, 150], and dynamic[148, 149] communities. *Local community mining algorithms*, in particular, are developed for large networks in which the global information on the whole network is not available or computationally expensive. These methods are based on a locally defined quality function on a subset of nodes in the network, *e.g.*, local variants of the modularity Q [25, 91], where the current local community expands by identifying its boundary nodes, and according to their ratio of internal and external edges. Leskovec et al. [88] compare a local variation of modularity Q with different alternative local objectives, including ratio cut, normalized cut, conductance, *etc.* . In their implementation, they start with a seed node, and score the nodes based on their proximity to the seed using a random walk, then the community is expanded from the closest node, and the objective is computed for each expansion; whereas the local optima of the objective correspond to the detected community. One should however note that obtaining the global clustering structure of the network using a local method is not a straightforward task. For instance, Tepper and Sapiro [150] highlight the challenges of this task, and present a consensus approach to integrate local communities, which are discovered when considering each node in the network as the seed, to reach the global community structure. More comprehensive surveys on community detection methods are available in [32, 47, 49, 121].

### 2.2.2 Classification of Common Evaluation Practices

Fortunato [47] shows that the different community mining algorithms discover communities from different perspective and may outperform others in specific classes of networks and have different computational complexities. Therefore, an important research direction is to evaluate and compare the results of different community mining algorithms, and select the one providing more meaningful clustering for each class of networks. An intuitive practice is to validate the results partly by a human expert [91]. However, the community mining problem is $NP$-hard [17]; and the human expert validation is limited, since it is based on narrow intuition rather than on an exhaustive examination of the relations in the given network, specially for large real networks. To validate the result of a community mining algorithm, one can consider three approaches: *external evaluation*, *internal evaluation*, and *relative evaluation*; which are described in the following.

***External evaluation*** compares the discovered clustering against a prespecified structure, often called ground-truth. There are few and typically small real world benchmarks with known ground-truth communities available for external evaluation of community mining algorithms. Hence, there exists benchmark generators which synthesize benchmarks with built-in communities. However, in a real-world application the interesting communities that need to be discovered are hidden in the structure of the network, thus, the discovered communities can not be validated based on the external evaluation. This motivate investigating the other two alternatives approaches – internal and relative evaluation. Before describing these evaluation approaches, we first review the main studies relevant to the external evaluation in community detection.

Girvan and Newman [51] propose the first synthetic network generator for community evaluation, called GN benchmarks. Their benchmark generates graphs with 128 nodes, and expected degree of 16, which are divided into four groups of equal sizes; where the probabilities of the existence of a link between a pair of nodes of the same group and of different groups are $z_{in}$ and $1 - z_{in}$, respectively. However due the simplicity of its structure, most of the algorithms perform well on these benchmarks. Lancichinetti et al. [79] amend the GN benchmark and propose the well-known LFR benchmarks. LFR considers power law distributions for the degrees of nodes and community sizes, which corresponds better with properties observed for real-world networks. Here, each node shares a fraction $1 - \mu$ of its links with the other nodes of its community and a fraction $\mu$ with the other nodes of the network. For a more elaborate discussion on the synthetic benchmark generators please refer to Chapter 4.

Apart from many papers that use the external evaluation to assess the performance of their proposed algorithms, there are recent studies specifically on comparison of different community mining algorithms using the external evaluation approach. For instance, Gustafsson et al. [59] compare hierarchical and k-means community mining on real networks and also synthetic networks generated by the GN benchmark. Lancichinetti and Fortunato [76] compare a total of a dozen community mining algorithms; where the performance of the algorithms is compared against the network generated by both GN and LFR benchmark. Orman et al. [115] compare a total of five community mining algorithms on the synthetic networks generated by LFR benchmark. They first assess the quality of the different algorithms by their difference with the ground-truth. Then, they perform a qualitative analysis of the identified communities by comparing their size distribution with the community size distribution of the ground-truth.

***Internal evaluation*** techniques verify whether the clustering structure produced by a clustering algorithm matches the underlying structure of the data, using only information inherent in the data. These techniques are based on an internal criterion that measures the correlation between the discovered clustering structure and the structure of the data, represented as a proximity matrix –a square matrix in which the entry in cell $(i, j)$ is some measure of the similarity (or distance) between the items $i$, and $j$. The significance of this correlation is examined statistically based on the distribution of the defined criteria, which is usually not known and is estimated using Monte Carlo sampling method [151]. An internal criterion can also be considered as a quality index to compare different clusterings which overlaps with relative evaluation techniques. [1] The well-known modularity Q of Newman [107] can be considered as such, which is used both to validate a single community mining result and also to compare different community mining results [28, 139]. Modularity is defined as the fraction of edges within communities, *i.e.*, the correlation of adjacency matrix and the clustering structure, minus the expected value of this fraction that is derived based on the configuration model [107]. Another work that could be considered in this class is the evaluation of different community mining algorithms studied in [88]. Where the authors propose network community profile (NCP) to characterize the quality of communities as a function of their size, then the shape of the NCPs are compared for different algorithms over random and real networks.

***Relative evaluation*** compares alternative clustering structures based on an objective function or quality index. This evaluation approach is the least explored in the community mining context. Defining an objective function to evaluate community mining is non-trivial. Aside from the subjective nature of the community mining task, there is no formal definition on the term community. Consequently, there is no consensus on how to measure "goodness" of the discovered communities. Nevertheless, the well-studied clustering methods in the Machine Learning field are subject to similar issues and yet there exists an extensive set of validity criteria defined for clustering evaluation, such as Davies-Bouldin index [39], Dunn index [44], and Silhouette [142]; for a survey refer to [155]. In the next section, we describe how these criteria could be adapted to the context of community mining in order to compare results of different community mining algorithms. Also, these criteria can be used as alternatives to modularity Q to design novel community mining algorithms.

[1] One should note that while any internal evaluation metric could be used also for the relative evaluation, the reverse it not the case; *i.e.*, the relative measures could not necessarily provide an internal evaluation [151].

## 2.3 Community Quality Criteria

Here, we overview several validity criteria that could be used as relative indexes for comparing and evaluating different partitionings of a given network, *i.e.*, a disjoint/non-overlapping clustering. All of these criteria are generalized from well-known clustering criteria. The clustering quality criteria are originally defined with the implicit assumption that data points consist of vectors of attributes. Consequently their definition is mostly integrated or mixed with the definition of the distance measure between data points. The commonly used distance measure is the Euclidean distance, which cannot be defined for graphs. Therefore, we first review different possible proximity measures that could be used in graphs. Then, we present generalizations of criteria that could use any notion of proximity.

### 2.3.1 Proximity Between Nodes

We consider the following extensive set of distance or similarity measures, to compute the proximity between nodes $i$ and $j$, which is denoted by $p_{ij}$. Since similarity is more natural in the context of networks, we directly plug-in similarities in the relative criteria definitions. For those criteria which can not use the similarities in a straightforward way, we keep the original distance based form and use the corresponding dissimilarity/distance (*e.g.*, inverse of the similarity) [2].

***Shortest Path (SP)*** distance between two nodes is the length of the shortest path between them, which could be computed using the well-known Dijkstra's Shortest Path algorithm.

***Adjacency (A)*** similarity between the two nodes $i$ and $j$ is considered their incident edge weight, $p^A_{ij} = A_{ij}$; where $A$ denotes the (weighted) adjacency matrix. Accordingly, the distance between these nodes is derived as:

$$d^A_{ij} = A_{max} - p^A_{ij} \tag{2.6}$$

where $A_{max}$ is the maximum edge weight in the graph; *i.e.*, $A_{max} = \max_{ij} A_{ij}$.

***Adjacency Relation (AR)*** distance between two nodes measures their structural dissimilarity, which is computed by the difference between their immediate neighbourhoods [161] as:

$$d^{AR}_{ij} = \sqrt{\sum_{k \neq j,i} (A_{ik} - A_{jk})^2} \tag{2.7}$$

This definition does not consider the existence of an edge between the two nodes $i$ and $j$. To remedy this, Augmented AR ($\hat{AR}$) is also defined; *i.e.*,

$$d^{\hat{AR}}_{ij} = \sqrt{\sum_{k} (\hat{A}_{ik} - \hat{A}_{jk})^2} \tag{2.8}$$

[2] To avoid division by zero, we always have $P_{ij} = \min(P_{ij}, \epsilon)$ where $\epsilon$ is a very small number, *i.e.*, 10E-9.

where $\hat{A}$ denotes the adjacency matrix augmented by self-loops, *i.e.*, $\hat{A}_{ij}$ is equal to $A_{ij}$ if $i \neq j$ and is $A_{max}$ when $i = j$.

***Neighbour Overlap (NO)*** similarity between two nodes is the ratio of their shared neighbours [47], and is defined as:

$$p_{ij}^{NO} = \frac{|\aleph_i \cap \aleph_j|}{|\aleph_i \cup \aleph_j|} \tag{2.9}$$

where $\aleph_i$ denotes the set of nodes directly connected to node $i$, *i.e.*, $\aleph_i = \{k|A_{ik} \neq 0\}$. The corresponding distance is derived as $d_{ij}^{NO} = 1-p_{ij}^{NO}$. There is a close relation between this measure and the previous one, since $d^{AR}$ can also be computed as: $d_{ij}^{AR} = \sqrt{|\aleph_i \cup \aleph_j| - |\aleph_i \cap \aleph_j|}$. We can also derive $d_{ij}^{\hat{AR}}$ from this formula, if we consider the neighbourhoods closed, *i.e.*, when $\hat{\aleph}_i = \{n_k|\hat{A}_{ik} \neq 0\}$. Hence, we also consider the closed neighbour overlap similarity, $p^{\hat{NO}}$, with the same analogy that two nodes are more similar if directly connected. The closed overlap similarity, $p^{\hat{NO}}$, could be rewritten in terms of the adjacency matrix, which then straightforwardly generalizes for weighted cases.

$$p_{ij}^{\hat{NO}} = \frac{\sum_k \hat{A}_{ik}\hat{A}_{jk}}{\sum_k [\hat{A}_{ik}^2 + \hat{A}_{jk}^2 - \hat{A}_{ik}\hat{A}_{jk}]} \tag{2.10}$$

We also consider the following variation:

$$p_{ij}^{N\hat{O}V} = \frac{\sum_k (\hat{A}_{ik} + \hat{A}_{jk})(\hat{A}_{ik} + \hat{A}_{jk}) - \sum_k (\hat{A}_{ik} - \hat{A}_{jk})(\hat{A}_{ik} - \hat{A}_{jk})}{\sum_k (\hat{A}_{ik} + \hat{A}_{jk})(\hat{A}_{ik} + \hat{A}_{jk}) + \sum_k (\hat{A}_{ik} - \hat{A}_{jk})(\hat{A}_{ik} - \hat{A}_{jk})} \tag{2.11}$$

***Topological Overlap (TP)*** similarity measures the normalized overlap size of the neighbourhoods [136], which we generalize as:

$$p_{ij}^{TP} = \frac{\sum_{k \neq j,i} (A_{ik}A_{jk}) + A_{ij}^2}{min(\sum_k A_{ik}^2, \sum_k A_{jk}^2)} \tag{2.12}$$

and the corresponding distance is derived as $d_{ij}^{TO} = 1 - p_{ij}^{TO}$.

***Pearson Correlation (PC)*** coefficient between two nodes is the correlation between their corresponding rows of the adjacency matrix, *i.e.*, :

$$p_{ij}^{PC} = \frac{\sum_k (A_{ik} - \mu_i)(A_{jk} - \mu_j)}{N\sigma_i\sigma_j} \tag{2.13}$$

where $N$ is the number of nodes, and for the average $\mu_i$ and the variance $\sigma_i$ we have:

$$\mu_i = (\sum_k A_{ik})/N \quad , \quad \sigma_i = \sqrt{\sum_k (A_{ik} - \mu_i)^2/N}$$

This correlation coefficient lies between −1 (when the two nodes are most similar) and 1 (when the two nodes are most dissimilar). Most relative clustering criteria are defined assuming distance is positive, therefore we also consider the normalized version of this correlation, *i.e.*, $p^{NPC} = (p^{PC}_{ij} + 1)/2$. Then, the distance between two nodes is computed as $d^{(N)PC}_{ij} = 1 - p^{(N)PC}_{ij}$.

In all the above proximity measures, the iteration over all other nodes can be limited to iteration over the nodes in the union of neighbourhoods. More specifically, in the formulae, one can use $\sum_{k \in \hat{\aleph}_i \cup \hat{\aleph}_j}$ instead of $\sum_{k=1}^{N}$. This will make the computation local and more efficient, especially in case of large networks. This strategy will not work for the current definition of the Pearson correlation, however, it can be applied if we reformulate it as follows:

$$p^{PC}_{ij} = \frac{\sum_k A_{ik}A_{jk} - (\sum_k A_{ik})(\sum_k A_{jk})/N}{\sqrt{((\sum_k A^2_{ik}) - (\sum_k A_{ik})^2/N)((\sum_k A^2_{jk}) - (\sum_k A_{jk})^2/N)}} \tag{2.14}$$

We also consider this correlation based on $\hat{A}$, which gives $p^{\hat{PC}}$, in which the existence of an edge between the two nodes, increases their correlation similarity. Note that since we are assuming a self edge for each node, $\hat{N} = N + 1$ should be used. The above formula can be further rearranged as follows:

$$p^{PC}_{ij} = \frac{\sum\limits_k \left[A_{ik}A_{jk} - (\sum\limits_{k'} A_{ik'})(\sum\limits_{k'} A_{jk'})/N^2\right]}{\sqrt{(\sum\limits_k \left[A^2_{ik} - (\sum\limits_{k'} A_{ik'})^2/N^2\right])(\sum\limits_k \left[A^2_{jk} - (\sum\limits_{k'} A_{jk'})^2/N^2\right])}} \tag{2.15}$$

Where if the index $k$ iterates over all nodes, it is equal to the original Pearson correlation. This is not the case if $k$ only iterates over the union of neighbourhoods, $\sum_{k \in \hat{\aleph}_i \cup \hat{\aleph}_j}$, which also consider and call Pearson overlap ($NPO$).

***Number of Paths (NP)*** between two nodes is the sum of all the paths between them, which is a notion of similarity. For the sake of time complexity, we consider paths of up to a certain number of hops *i.e.*, 2 and 3. The number of paths of length $l$ between nodes $i$ and $j$ can be computed as $np^l_{ij} = (A^l)_{ij}$. More specifically we have: $np^1_{ij} = A_{ij}$, $np^2_{ij} = \sum_k A_{ik}A_{jk}$, and $np^3_{ij} = \sum_{kl} A_{ik}A_{kl}A_{jl}$. We consider the follwoing combinations of these as different proximity measures:

$$p^{NP^2} = np^1 + np^2, \quad \textit{and} \quad p^{NP^3} = np^1 + np^2 + np^3 \tag{2.16}$$

$$p^{NP^3_L} = np^1 + \frac{np^2}{2} + \frac{np^3}{3}, \quad \textit{and} \quad p^{NP^3_E} = np^1 + \sqrt[2]{np^2} + \sqrt[3]{np^3} \tag{2.17}$$

***Modularity (M)*** similarities are defined inspired by the modularity Q [107] as:

$$p^M_{ij} = A_{ij} - \frac{(\sum_k A_{ik})(\sum_k A_{jk})}{\sum_{kl} A_{kl}}, \quad \textit{and} \quad p^{MD}_{ij} = \frac{A_{ij}}{\frac{(\sum_k A_{ik})(\sum_k A_{jk})}{\sum_{kl} A_{kl}}} \tag{2.18}$$

The distance is derived as $1 - p^{M(D)}$.

***ICloseness (IC)*** similarity between two nodes is computed as the inverse of the connectivity between their scored neighbourhoods:

$$p_{ij}^{IC} = \frac{\sum_k s_{ki}s_{kj}}{\sum_k s_{ki}^2 + \sum_k s_{kj}^2 - \sum_k s_{ki}s_{kj}} \tag{2.19}$$

where $s_{ki}$ denotes the neighbouring score of node $k$ to $i$; for complete formulation refer to [124]. This score is comouted for a neighbourhood of specified depth; here we consider 3 variations: direct neighbourhood (IC1), neighbourhood of depth 2 *i.e.*, neighbours up to one hop apart (IC2), and neighbourhood of depth 3 *i.e.*, neighbours up to two hops apart (IC3). We also consider the following variation:

$$p_{ij}^{ICV} = \frac{\sum\limits_k (s_{ki}+s_{kj})(s_{ki}+s_{kj}) - \sum\limits_k (s_{ki}-s_{kj})(s_{ki}-s_{kj})}{\sum\limits_k (s_{ki}+s_{kj})(s_{ki}+s_{kj}) + \sum\limits_k (s_{ki}-s_{kj})(s_{ki}-s_{kj})} \tag{2.20}$$

The distance is then derived as $d^{IC(V)} = 1 - p^{IC(V)}$.

### 2.3.2 Community Centroid

In addition to the notion of proximity measure, most of the cluster validity criteria use averaging between the numerical data points to determine the centroid of a cluster. The averaging is not defined for nodes in a graph, therefore we modify the criteria definitions to use a generalized centroid notion, in a way that, if the centroid is set as averaging, we would obtain the original criteria definitions, but we could also use other alternative notions for centroid of a group of data points. Averaging data points results in a point with the least average distance to the other points. When averaging is not possible, using medoid is the natural option, which is perfectly compatible with graphs. More formally, the centroid of the community $C$ can be obtained as:

$$\overline{C} = \underset{m \in C}{\arg\min} \sum_{i \in C} d(i, m) \tag{2.21}$$

### 2.3.3 Relative Validity Criteria

Here, we present our generalizations of well-known clustering validity criteria defined as quality measures for internal or relative evaluation of clustering results. All these criteria are originally defined based on distances between data points, which in all cases is the Euclidean or other inner product norms of difference between their vectors of attributes; refer to [155] for comparative analysis of these criteria in the clustering context. We alter the formulae to use a generalized distance, so that we can plug in our graph proximity measures. The other alteration is generalizing the mean over data points to a general centroid notion, which can be set as averaging in the presence of attributes and the *medoid* in our case of dealing with graphs and in the absence of attributes.

In a nutshell, in every criterion, the average of points in a cluster is replaced with a generalized notion of centroid, and distances between data points are generalized from Euclidean/norm to a generic distance. Consider a partitioning $C = \{C_1, C_2, ...C_k\}$ of $N$ data points, where $\overline{C_l}$ denotes the (generalized) centroid of data points belonging to $C_l$ and $d(i, j)$ denotes the (generalized) distance between point $i$ and point $j$. The quality of $C$ can be measured using one of the following criteria.

***Variance Ratio Criterion (VRC)*** measures the ratio of the between-cluster/community distances to within-cluster/community distances which could be generalized as follows:

$$VRC = \frac{\sum_{l=1}^{k} |C_l| d(\overline{C}_l, \overline{C})}{\sum_{l=1}^{k} \sum_{i \in C_l} d(i, \overline{C}_l)} \times \frac{N-k}{k-1} \tag{2.22}$$

where $\overline{C}_l$ is the centroid of the cluster $C_l$, and $\overline{C}$ is the centroid of the entire data/network. Consequently $d(\overline{C}_l, \overline{C})$ is measuring the distance between centroid of cluster $C_l$ and the centroid of the entire data, while $d(i, \overline{C}_l)$ is measuring the distance between data point $i$ and its cluster centroid. The original clustering formula proposed by Calinski and Harabasz [19] for attributes vectors is obtained if the centroid is fixed to averaging of vectors of attributes and distance to (square of) Euclidean distance. Here we use this formula with one of the proximity measures mentioned in the preious section; if it is a similarity measure, we either transform the similarity to its distance form and apply the above formula, or we use it directly as a similarity and inverse the ratio to within/between while keeping the normalization, the latter approach is distinguished in the experiments as $VRC'$.

***Davies-Bouldin index (DB)*** calculates the worst-case within-cluster to between-cluster distances ratio averaged over all clusters/communities [39]:

$$DB = \frac{1}{k} \sum_{l=1}^{k} \max_{m \neq l} \frac{\overline{d}_l + \overline{d}_m}{d(\overline{C}_l, \overline{C}_m)} \ , \ where \quad \overline{d}_l = \frac{1}{|C_l|} \sum_{i \in C_l} d(i, \overline{C}_l)$$

If used directly with a similarity measure, we change the max in the formula to min and the final criterion becomes a maximizer instead of minimizer, which is denoted by $DB'$.

***Dunn index*** considers both the minimum distance between any two clusters/communities and the length of the largest cluster/community diameter (*i.e.*, the maximum or the average distance between all the pairs in the cluster/community) [44]:

$$Dunn = \min_{l \neq m} \{ \frac{\delta(C_l, C_m)}{max_p \Delta(C_p)} \} \tag{2.23}$$

where $\delta$ denotes distance between two communities and $\Delta$ is the diameter of a community. Different variations of calculating $\delta$ and $\Delta$ are available; $\delta$ could be single, complete or average linkage, or only the difference between the two centroids. Moreover, $\Delta$ could be maximum or average distance between all pairs of nodes, or the average distance of all nodes to the centroid. For example, the single linkage for $\delta$ and maximum distance for $\Delta$ are $\delta(C_l, C_m) = \min_{i \in C_l, j \in C_m} d(i, j)$ and $\Delta(C_p) = \max_{i,j \in C_p} d(i, j)$. Therefore, we have different variations of Dunn index in our experiments, each indicated by two indexes for different methods to calculate $\delta$ (*i.e.*, single(0), complete(1), average(2), and centroid(3)) and different methods to calculate $\Delta$ (*i.e.*, maximum(0), average(1), average to centroid(3)).

***Silhouette Width Criterion (SWC)*** measures the average silhouette scores, which is computed individually for each data point. The silhouette score of a point shows the goodness of the assignment of this point to the community it belongs to, by calculating the normalized difference between the distance to its nearest neighbouring community and the distance to its own community [142]. Taking the average one has:

$$SWC = \frac{1}{N} \sum_{l=1}^{k} \sum_{i \in C_l} \frac{\min_{m \neq l} d(i, C_m) - d(i, C_l)}{\max \{\min_{m \neq l} d(i, C_m), d(i, C_l)\}} \tag{2.24}$$

where $d(i, C_l)$ is the distance of point $i$ to community $C_l$, which is originally set to be the average distance, *i.e.*, $1/|C_l| \sum_{j \in C_l} d(i, j)$, which we call SWC0. The $d(i, C_l)$ could also be the distance to the centroid, *i.e.*, $d(i, \overline{C_l})$, which we call SWC1. An alternative formula for Silhouette is proposed in [155] :

$$ASWC = \frac{1}{N} \sum_{l=1}^{k} \sum_{i \in C_l} \frac{\min_{m \neq l} d(i, C_m)}{d(i, C_l)} \tag{2.25}$$

Similar to *DB*, if used directly with a similarity proximity measure, we change the min to max and the final criterion becomes a minimizer instead of maximizer, which is denoted by $(A)SWC'$.

***PBM*** criterion is based on the within-community distances and the maximum distance between centroids of communities[116]:

$$PBM = \frac{1}{k} \times \frac{\max_{l,m} d(\overline{C}_l, \overline{C}_m)}{\sum_{l=1}^{k} \sum_{i \in C_l} d(i, \overline{C}_l)} \tag{2.26}$$

Again similar to *DB*, here also if used directly with a similarity measure, we change the max to min and consider the final criterion as a minimizer instead of maximizer, which is denoted by $PBM'$.

***C-Index*** criterion compares the sum of the within-community distances to the worst and best case scenarios [36]. The best case scenario is where the within-community distances are the shortest distances in the graph, and the worst case scenario is where the within-community distances are the longest distances in the graph.

$$CIndex = \frac{\theta - \min \theta}{\max \theta - \min \theta}\ , \ where \quad \theta = \frac{1}{2} \sum_{l=1}^{k} \sum_{i,j \in C_l} d(i, j) \tag{2.27}$$

The $\min\theta/\max\theta$ is computed by summing the $\Theta$ smallest/largest distances between every two points, where $\Theta = \frac{1}{2}\sum_{l=1}^{k}|C_l|(|C_l|-1)$. C-Index can be directly used with a similarity measure as a maximization criterion, whereas with a distance measure it is a minimizer. This is also true for the two following criteria.

***Z-Statistics*** criterion is defined similar to C-Index [65]:

$$ZIndex = \frac{\theta - E(\theta)}{\sqrt{var(\theta)}}\ , \quad where \tag{2.28}$$

$$\bar{d} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} d(i,j)\ , \quad E(\theta) = \Theta \times \bar{d}\ , \quad Var(\theta) = \frac{1}{4}\sum_{l=1}^{k}\sum_{i,j\in C_l}(d(i,j)-\bar{d})^2$$

***Point-Biserial (PB)*** This criterion computes the correlation of the distances between nodes and their cluster co-membership which is dichotomous variable [97]. Intuitively, nodes that are in the same community should be separated by shorter distances than those which are not:

$$PB = \frac{M_1 - M_0}{S}\sqrt{\frac{m_1 m_0}{m^2}} \tag{2.29}$$

where $m$ is the total number of distances *i.e.*, $N(N-1)/2$ and $S$ is the standard deviation of all pairwise distances *i.e.*, $\sqrt{\frac{1}{m}\sum_{i,j}(d(i,j)-\frac{1}{m}\sum_{i,j}d(i,j))^2}$, while $M_1$, $M_0$ are respectively the average of within and between-community distances, and $m_1$ and $m_0$ represent the number of within and between community distances. More formally:

$$m_1 = \sum_{l=1}^{k}\frac{N_l(N_l-1)}{2}\ ,\ m_0 = \sum_{l=1}^{k}\frac{N_l(N-N_l)}{2}\ ,\ M_1 = 1/2\sum_{l=1}^{k}\sum_{i,j\in C_l}d(i,j)\ ,\ M_0 = 1/2\sum_{l=1}^{k}\sum_{\substack{i\in C_l\\ j\notin C_l}}d(i,j)$$

***Modularity*** is the well-known criterion proposed by Newman and Girvan [108] specifically for the context of community mining. Let $E$ denote the number of edges in the network *i.e.*, $E = \frac{1}{2}\sum_{ij}A_{ij}$, then Q-modularity is defined as:

$$Q = \frac{1}{2E}\sum_{l=1}^{k}\sum_{i,j\in C_l}[A_{ij} - \frac{\sum_k A_{ik}\sum_k A_{kj}}{2E}] \tag{2.30}$$

### 2.3.4 Computational Complexity Analysis

The computational complexity of different clustering validity criteria is provided in the previous work by Vendramin et al. [155]. For the adapted criteria, the time complexity of the indexes is affected by the cost of the chosen proximity measure. All the proximity measures we introduced here can be computed in linear time, $O(n)$, except for the $A$ (adjacency) which is $O(1)$, the $NP$ (number of paths) which is $O(n^2)$ and the $IC$ (Icloseness) which is $O(E)$. However, for the case of sparse graphs and using a proper graph data structure such as incidence list, this complexity can be reduced to $O(\hat{d})$, where $\hat{d}$ is the average degree in the network, *i.e.*, the average neighbours of a node in the network. For example, let us revisit the formula for $AR$ (adjacency relation): $d_{ij}^{AR} = \sqrt{\sum_{k\neq j,i} (A_{ik} - A_{jk})^2}$. In this formula we can change $\sum_k$ to $\sum_{k\in\aleph_i\cup\aleph_j}$ since the expression $(A_{ik} - A_{jk})^2$ is zero for other values of $k$, *i.e.*, for nodes that are not neighbour to either $i$ or $j$ and therefore have $A_{ik} = A_{jk} = 0$. The same strategy could be applied to other proximity measures.

The other cost that should be considered is the cost of computing the medoid of $m$ data points, which is $O(pm^2)$, where $p$ is the cost of the proximity measure. Therefore the $VRC$ criterion that require computing the overall centroid, is in order of $O(pn^2)$. This is while the $VRC$ for traditional clustering is linear with respect to the size of the dataset, since it uses averaging for computing the centroid which is $O(n)$. Similarly, any other measure that requires computing all the pairwise distances will have the $\Omega(pn^2)$. This holds for the adapted *Dunn* index which is in order of $O(pn^2)$, becuase for finding the minimum distances between any two clusters, it requires to compute the distances between all pair of nodes. Similarly, the $ZIndex$ computes all the pairwise distances, and is in order of $O(pn^2)$. The same also holds for the *PB*. The $CIndex$ is even more expensive since it not only computes all the pairwise distances but also sorts them, and hence is in order of $O(n^2(p + logn))$. These orders (except for $VRC$) are along the computational complexities previously reported in Vendramin et al. [155], where the cost of the $p$ is the size of the feature vectors there.

The adapted *DB* and *PBM*, on the other hand, do not require computing the medoid of the whole dataset nor all pairwise distances. Instead they only compute the medoid of each cluster, which makes them in $\Omega(pk\hat{m}^2)$, where $k$ is the number of clusters and the $\hat{m}$ is the average size of the clusters. Consequently, this term will be added to the complexity of these criteria, giving them the order of $O(p(n + k^2 + k\hat{m}^2))$. Finally for the silhouette criterion, the $(A)SWC0$ that uses the average distance, has the order of $O(pn^2)$, however the order for $(A)SWC1$ is simplified to $O(kp(n + \hat{m}^2))$ since it uses the distance to centroid instead of averaging. The latter is similar to the order for modularity $Q$ which is $O(k(n + \hat{m}^2))$. To sum up, none of the adapted criteria is significantly superior or inferior in terms of its order, therefore one should focus on which criterion is more appropriate according to its performance which is demonstrated in the experiments.

## 2.4 Comparison Methodology and Results

In this section, we first describe our experimental settings. Then, we report the performances of the proposed community quality criteria in relative evaluation of communities.

### 2.4.1 Experiment Settings

We have used three sets of benchmarks as our datasets: Real, GN and LFR. The Real dataset consists of five well-known real-world benchmarks: Karate Club (weighted) by Zachary [171], Sawmill Strike data-set [111], NCAA Football Bowl Subdivision [51], and Politician Books from Amazon

[73]. The GN and LFR datasets, each include 10 realizations of the GN and LFR synthetic benchmarks [79], which are the benchmarks widely used for community mining evaluation. For each realization, we generate different partitionings to sample the space of all possible partitionings. For doing so, given the ground-truth, we generate different randomized versions of the true partitioning by randomly merging and splitting communities and swapping nodes between them. The sampling procedure is described in more detail in Appendix A. The set of the samples obtained covers the partitioning space in a way that it includes very poor to perfect samples.

### 2.4.2 Comparison Methodology

The performance of a criterion could be examined by how well it could rank different partitionings of a given dataset. More formally, consider for the dataset $d$, we have a set of $m$ different possible partitionings: $P(d) = \{p_1, p_2, \ldots, p_m\}$. Then, the performance of criterion $c$ on dataset $d$ could be determined by how much its values, $I_c(d) = \{c(p_1), c(p_2), \ldots, c(p_m)\}$, correlate with the "goodness" of these partitionings. Assuming that the true partitioning (*i.e.*, ground-truth) $p^*$ is known for dataset $d$, the "goodness" of partitioning $p_i$ could be determined using partitioning agreement measure $a$. Hence, for dataset $d$ with set of possible partitionings $P(d)$, the external evaluation provides $E(d) = \{a(p_1, p^*), a(p_2, p^*), \ldots, a(p_m, p^*)\}$, where $(p_1, p^*)$ denotes the "goodness" of partitioning $p_1$ comparing to the ground-truth. Then, the performance score of criterion $c$ on dataset $d$ could be examined by the correlation of its values $I_c(d)$ and the values obtained from the external evaluation $E(d)$ on different possible partitionings. Finally, the criteria are ranked based on their average performance score over a set of datasets. The following procedure summarizes our comparison approach.

```
$D \leftarrow \{d_1, d_2, \ldots, d_n\}$
for all dataset $d \in D$ do
    $P(d) \leftarrow \{p_1, p_2, \ldots, p_m\}$        {generate $m$ possible partitionings}
    $E(d) \leftarrow \{a(p_1, p^*), a(p_2, p^*), \ldots, a(p_m, p^*)\}$        {compute the external scores}
    for all $c \in Criteria$ do
        $I_c(d) \leftarrow \{c(p_1), c(p_2), \ldots, c(p_m)\}$        {compute the internal scores}
        $score_c(d) \leftarrow correlation(E, I)$        {compute the correlation}
$score_c \leftarrow \frac{1}{n} \sum_{d=1}^{n} score_c(d)$        {rank criteria based on their average scores}
```

External scores are obtained using a clustering agreement measure. There are several choices for this agreement measure, we consider four commonly used ones: Jaccard Coefficient [5], Adjusted Rank Index (ARI) [64], Normalized Mutual Information (NMI) [38], and Adjusted Mutual Information (AMI) [157]. There are also different ways to compute the correlation between two vectors, most notably Pearson Product Moment coefficient and the Spearman's Rank correlation coefficient. The reported results in our experiments are based on the Spearman's Correlation, since we are interested on the correlation of rankings that a criterion provides for different partitionings and not the actual values of that criterion. However, the reported results mostly agree with the results obtained by using Pearson correlation.

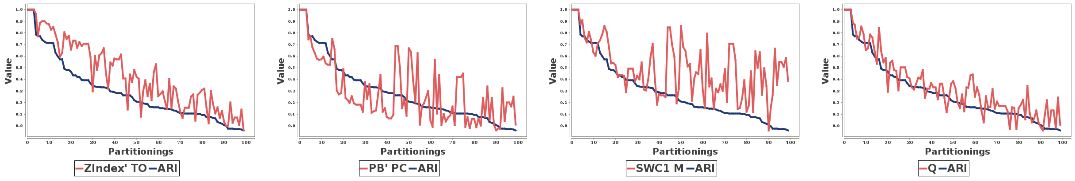
### 2.4.3 Results on Real World Datasets

Table 2.1 shows general statistics of our real world datasets and their generated samples. We can see that the randomized samples cover the space of partitionings according to their external index range.

| Dataset | $K^*$ | # | $\overline{K}$ | $\overline{ARI}$ |
|---|---|---|---|---|
| strike | 3 | 100 | 3.2±1.08∈[2,7] | 0.45±0.27∈[0.01,1] |
| polboks | 3 | 100 | 4.36±1.73∈[2,9] | 0.43±0.2∈[0.03,1] |
| karate | 2 | 100 | 3.82±1.51∈[2,7] | 0.29±0.26∈[-0.04,1] |
| football | 11 | 100 | 12.04±4.8∈[4,25] | 0.55±0.22∈[0.16,1] |

***Table 2.1:*** *Statistics for sample partitionings of each real world dataset. For example, for the Karate Club dataset which has 2 communities in its ground-truth, we have generated 100 different partitionings with average 3.82±1.51 clusters ranging from 2 to 7 and the "goodness" of the samples is on average 0.29±0.26 in terms of their ARI agreement.*

Figure 2.1 exemplifies how different criteria exhibit different correlations with the external index. It visualizes the correlation between few selected relative indexes and an external index for one of our datasets listed in Table 2.1.



*ZIndex with Topological Overlap* *Point-Biserial with Pearson Correlation* *Silhouette with Modularity Proximity* *Q modularity*

***Figure 2.1:*** *Visualization of correlation between an external agreement measure and some relative quality criteria for Karate dataset. The x axis indicates different random partitionings, and the y axis indicates the value of the index. While, the blue/darker line represents the value of the external index for the given partitioning and the red/lighter line represents the value that the criterion gives for the partitioning. Please note that the value of criteria are not generally normalized and in the same range as the external indexes, in this figure ARI. For the sake of illustration therefore, each criterion's values are scaled to be in the same range as of the external index.*

Similar analysis is done for all 4 datasets × 645 criteria (combination of relative indexes and distances variations) × 5 external indexes, which produced over 12900 such correlations. The top ranked criteria based on their average performance over these datasets are summarized in Table 2.2. Based on these results, *ZIndex* when used with almost all of the proximity measures, such as Topological Overlap (*TO*), Pearson Correlation Similarity (*PC*) or Intersection Closeness (*IC*); has a higher correlation with the external index comparing to the modularity *Q*. And this is true regardless of the choice of *ARI* as the external index, since it is ranked above *Q* by other external indexes, *e.g.*, , *NMI* and *AMI*. Other criteria, on the other hand, are all ranked after the modularity *Q*, except the*CIndex SP*. One may conclude based on this experiment that *ZIndex* is a more accurate evaluation criterion comparing to *Q*.

| Rank | Criterion | $ARI_{corr}$ | Rand | Jaccard | NMI | AMI |
|---|---|---|---|---|---|---|
| 1 | $ZIndex'$ TO | 0.925±0.018 | 9 | 148 | 9 | 7 |
| 2 | $ZIndex'$ $\hat{P}C$ | 0.923±0.012 | 2 | 197 | 2 | 2 |
| 3 | $ZIndex'$ $N\hat{P}C$ | 0.923±0.012 | 3 | 198 | 1 | 1 |
| 4 | $ZIndex'$ IC2 | 0.922±0.024 | 8 | 182 | 5 | 3 |
| 5 | $ZIndex'$ $\hat{T}O$ | 0.922±0.016 | 10 | 153 | 8 | 8 |
| 6 | $ZIndex'$ $N\hat{P}O$ | 0.921±0.014 | 6 | 204 | 3 | 4 |
| 7 | $ZIndex'$ ICV2 | 0.919±0.04 | 18 | 163 | 12 | 10 |
| 8 | $ZIndex'$ PC | 0.918±0.018 | 4 | 207 | 10 | 11 |
| 9 | $ZIndex'$ IC3 | 0.918±0.039 | 19 | 165 | 15 | 12 |
| 10 | $ZIndex'$ $N\hat{O}V$ | 0.915±0.014 | 11 | 213 | 6 | 9 |
| 11 | $ZIndex'$ IC1 | 0.912±0.02 | 5 | 235 | 13 | 20 |
| 12 | $ZIndex'$ NPE2 | 0.911±0.03 | 26 | 168 | 21 | 15 |
| 13 | $ZIndex'$ NOV | 0.91±0.023 | 12 | 225 | 18 | 21 |
| 14 | $ZIndex'$ ICV1 | 0.91±0.023 | 13 | 226 | 19 | 22 |
| 15 | $ZIndex'$ $N\hat{P}E2$ | 0.91±0.025 | 23 | 184 | 22 | 19 |
| 16 | $ZIndex'$ NPL2 | 0.909±0.02 | 24 | 202 | 14 | 13 |
| 17 | $ZIndex'$ M | 0.908±0.028 | 25 | 149 | 26 | 23 |
| 18 | $ZIndex'$ ICV3 | 0.908±0.057 | 29 | 176 | 28 | 25 |
| 19 | $ZIndex'$ NP2 | 0.907±0.021 | 20 | 212 | 16 | 14 |
| 20 | $ZIndex'$ $N\hat{P}L2$ | 0.906±0.022 | 21 | 216 | 17 | 17 |
| 21 | $ZIndex'$ $N\hat{P}2$ | 0.906±0.022 | 22 | 217 | 20 | 18 |
| 22 | $ZIndex'$ $\hat{N}O$ | 0.905±0.022 | 16 | 253 | 11 | 16 |
| 23 | $ZIndex'$ NO | 0.904±0.034 | 7 | 250 | 23 | 31 |
| 24 | $ZIndex'$ $\hat{M}M$ | 0.903±0.037 | 17 | 233 | 24 | 30 |
| 25 | $CIndex$ SP | 0.9±0.02 | 1 | 251 | 31 | 42 |
|  |  | ⋮ |  |  |  |  |
| 36 | $ZIndex'$ MD | 0.894±0.048 | 34 | 179 | 33 | 32 |
| 37 | $ZIndex'$ $\hat{A}$ | 0.891±0.05 | 27 | 241 | 37 | 37 |
| 38 | Q | 0.878±0.034 | 45 | 110 | 45 | 44 |
| 39 | $CIndex'$ NPE3 | 0.876±0.054 | 43 | 9 | 4 | 6 |
| 40 | $CIndex'$ ICV3 | 0.869±0.069 | 44 | 4 | 7 | 5 |

***Table 2.2:*** *Overall ranking of criteria on the real world datasets, based on the average Spearman's correlation of criteria with the ARI external index, $ARI_{corr}$. Ranking based on correlation with other external indexes is also reported. The full ranking of the top 50 criteria is reported in Appendix A.*

The correlation between a criterion and an external index depends on how close the randomized partitionings are from the true partitioning of the ground-truth. This can be seen in Figure 2.1. For example, *SWC1* (Silhouette with Criterion where distance of a node to a community is computed by its distance to the centroid of that community) with the Modularity *M* proximity agrees strongly with the external index in samples with higher external index value, *i.e.*, closer to the ground-truth, but not on further samples. We can also see the similar pattern in the Point-Biserial with *PC* proximity. With this in mind, we have divided the generated clustering samples into three sets of easy, medium and hard samples and re-ranked the criteria in each of these settings. Since

| Near Optimal Samples | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | $ZIndex'$ $N\hat{P}C$ | 0.851±0.081 | 1 | 3 | 4 | 5 |
| 2 | $ZIndex'$ $\hat{PC}$ | 0.851±0.081 | 2 | 4 | 3 | 3 |
| 3 | ZIndex SP | 0.847±0.084 | 18 | 2 | 8 | 8 |
| 4 | $ZIndex'$ $N\hat{P}O$ | 0.845±0.088 | 3 | 9 | 6 | 6 |
| 5 | *DB* ICV2 | 0.845±0.065 | 30 | 1 | 31 | 30 |
| 6 | $ZIndex'$ $NP\hat{E}$3.0 | 0.842±0.082 | 10 | 5 | 2 | 2 |
| 7 | $ZIndex'$ ICV3 | 0.839±0.084 | 4 | 20 | 20 | 21 |
| | | ⋮ | | | | |
| 37 | Q | 0.762±0.166 | 39 | 21 | 41 | 41 |
| 38 | *DB* ICV3 | 0.757±0.126 | 37 | 35 | 38 | 36 |
| 39 | *DB* IC3 | 0.753±0.176 | 35 | 36 | 39 | 39 |
| 40 | $PB'$ PC | 0.753±0.289 | 45 | 26 | 71 | 71 |
| Medium Far Samples | | | | | | |
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | $ZIndex'$ TO | 0.775±0.087 | 5 | 361 | 22 | 20 |
| 2 | $ZIndex'$ $\hat{TO}$ | 0.771±0.091 | 6 | 386 | 19 | 17 |
| 3 | $ZIndex'$ IC3 | 0.768±0.134 | 2 | 372 | 16 | 13 |
| 4 | $ZIndex'$ ICV2 | 0.766±0.124 | 3 | 370 | 2 | 2 |
| 5 | $ZIndex'$ NPL3.0 | 0.762±0.079 | 12 | 349 | 28 | 27 |
| 6 | $ZIndex'$ ICV3 | 0.757±0.12 | 4 | 376 | 21 | 19 |
| 7 | $ZIndex'$ NP3.0 | 0.756±0.085 | 15 | 354 | 29 | 28 |
| | | ⋮ | | | | |
| 30 | Q | 0.69±0.151 | 58 | 70 | 79 | 72 |
| | | ⋮ | | | | |
| 46 | $PB'$ PC | 0.623±0.06 | 112 | 28 | 200 | 157 |
| Far Far Samples | | | | | | |
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | $ZIndex'$ ICV2 | 0.724±0.066 | 36 | 520 | 4 | 9 |
| 2 | $ZIndex'$ IC3 | 0.72±0.062 | 40 | 523 | 11 | 19 |
| 3 | $ZIndex'$ ICV3 | 0.717±0.059 | 47 | 511 | 23 | 25 |
| 4 | $ZIndex'$ IC2 | 0.715±0.072 | 35 | 540 | 3 | 6 |
| 5 | $ZIndex'$ TO | 0.706±0.064 | 49 | 519 | 16 | 14 |
| 6 | $ZIndex'$ $N\hat{P}O$ | 0.704±0.076 | 44 | 547 | 1 | 3 |
| 7 | $ZIndex'$ $\hat{TO}$ | 0.704±0.062 | 51 | 522 | 13 | 5 |
| | | ⋮ | | | | |
| 30 | $ZIndex'$ IC1 | 0.655±0.132 | 43 | 566 | 34 | 40 |
| 31 | $ZIndex'$ $\hat{NO}$ | 0.651±0.106 | 52 | 567 | 22 | 26 |
| 32 | Q | 0.643±0.033 | 86 | 444 | 50 | 45 |
| | | ⋮ | | | | |
| 117 | $PB'$ PC | 0.372±0.126 | 197 | 170 | 159 | 129 |

***Table 2.3:*** *Difficulty analysis of the results: considering ranking for partitionings near optimal ground-truth, medium far and very far. Reported result are based on ARI and the Spearman's correlation.*

the external index determines how far a sample is from the optimal result, the samples are divided into three equal length intervals according to the range of the external index. Table 2.3 reports the rankings of the top criteria in each of these three settings. We can see that these average results support our earlier hypothesis, *i.e.*, when considering partitionings near or medium far from the true partitioning, $PB' PC$ is between top criteria, while its performance drops significantly for samples very far from the ground-truth.

### 2.4.4 Results on Synthetic Benchmarks Datasets

Similar to the last experiment, Table 2.5 reports the ranking of the top criteria according to their average performance on synthesized datasets of Table 2.4. Based on which, *ZIndex* overall outperforms other criteria including the modularity $Q$, this is more significant in ranking finner partitionings, near optimal; while it is less significant in ranking poor partitionings.

| Dataset | $K^*$ | # | $\overline{K}$ | $\overline{ARI}$ |
|---|---|---|---|---|
| network1 | 4 | 100 | 5.26±2.45∈[2,12] | 0.45±0.18∈[0.13,1] |
| network2 | 3 | 100 | 4±1.7∈[2,8] | 0.47±0.23∈[0.06,1] |
| network3 | 2 | 100 | 4±1.33∈[2,6] | 0.36±0.22∈[0.07,1] |
| network4 | 7 | 100 | 10.68±3.3∈[4,19] | 0.69±0.21∈[0.25,1] |
| network5 | 2 | 100 | 4.68±1.91∈[2,9] | 0.32±0.22∈[-0.01,1] |
| network6 | 5 | 100 | 5.98±2.63∈[2,14] | 0.52±0.21∈[0.12,1] |
| network7 | 4 | 100 | 6.62±2.72∈[2,12] | 0.52±0.22∈[0.11,1] |
| network8 | 5 | 100 | 5.8±2.45∈[2,12] | 0.55±0.22∈[0.15,1] |
| network9 | 5 | 100 | 6.54±2.08∈[3,11] | 0.64±0.2∈[0.25,1] |
| network10 | 6 | 100 | 8.88±2.74∈[4,15] | 0.59±0.19∈[0.21,1] |

***Table 2.4:*** *Statistics for sample partitionings of each synthetic dataset. The benchmark generation parameters: 100 nodes with average degree 5 and maximum degree 50, where size of each community is between 5 and 50 and mixing parameter is* 0.1.

The LFR generator can generate networks with different levels of difficulty for the partitioning task, by changing how well separated the communities are in the ground-truth. To examine the effect of this difficulty parameter, we have ranked the criteria for different values of this parameter. We observed that modularity Q becomes the overall superior criterion for synthetic benchmarks with higher level of mixed communities $(.3 \leq \mu \leq .5)$. Table 2.6 reports the overall ranking of the criteria for a difficult set of datasets that have high mixing parameter. We can see that although $Q$ is the overall superior criterion, *ZIndex* still significantly outperforms $Q$ in ranking finer partitionings. In short, the relative performances of different criteria depends on the difficulty of the network itself, as well as how far we are sampling from the ground-truth. Altogether, choosing the right criterion for evaluating different community mining results depends both on the application, *i.e.*, how well-separated communities might be in the given network, and also on the algorithm that produces these results, *i.e.*, how fine the results might be. For example, if the algorithm is producing high quality results close to the optimal, modularity $Q$ might not distinguish the good and bad partitionings very well. While if we are choosing between mixed and not well separated clusterings, it is the superior criterion.

| Overall Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | $ZIndex'$ ICV2 | 0.96±0.029 | 5 | 32 | 3 | 3 |
| 2 | $ZIndex'$ IC3 | 0.958±0.028 | 4 | 42 | 2 | 2 |
| 3 | $ZIndex'$ IC2 | 0.958±0.033 | 1 | 58 | 1 | 1 |
| 4 | $ZIndex'$ $\hat{PC}$ | 0.953±0.04 | 3 | 78 | 6 | 6 |
| 5 | $ZIndex'$ $\hat{NPC}$ | 0.953±0.04 | 2 | 79 | 7 | 7 |
| 6 | $ZIndex'$ ICV3 | 0.953±0.027 | 8 | 44 | 4 | 5 |
| 7 | $ZIndex'$ $\hat{NPO}$ | 0.951±0.041 | 6 | 83 | 9 | 9 |
| 8 | $ZIndex'$ $\hat{TO}$ | 0.949±0.045 | 13 | 60 | 17 | 17 |
| 9 | $ZIndex'$ $\hat{NOV}$ | 0.949±0.042 | 7 | 90 | 8 | 8 |
| | | ⋮ | | | | |
| 30 | Q | 0.893±0.046 | 33 | 33 | 26 | 22 |
| Near Optimal Results | | | | | | |
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | $ZIndex'$ IC2 | 0.826±0.227 | 2 | 10 | 4 | 6 |
| 2 | $CIndex'$ ICV2 | 0.822±0.132 | 7 | 1 | 11 | 7 |
| 3 | $ZIndex'$ IC3 | 0.821±0.232 | 1 | 16 | 5 | 9 |
| 4 | $CIndex'$ ICV3 | 0.818±0.237 | 4 | 9 | 3 | 5 |
| 5 | $ZIndex'$ ICV2 | 0.816±0.232 | 3 | 18 | 7 | 10 |
| 6 | $ZIndex'$ $\hat{A}$ | 0.813±0.225 | 5 | 19 | 2 | 2 |
| 7 | $CIndex'$ IC3 | 0.8±0.2 | 31 | 2 | 13 | 8 |
| 8 | $ZIndex'$ A | 0.795±0.177 | 30 | 20 | 6 | 4 |
| | | ⋮ | | | | |
| 207 | Q | 0.589±0.161 | 222 | 198 | 138 | 110 |
| Medium Far Results | | | | | | |
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | $ZIndex'$ ICV2 | 0.741±0.177 | 4 | 231 | 22 | 22 |
| 2 | $ZIndex'$ IC2 | 0.738±0.181 | 1 | 247 | 16 | 20 |
| 3 | $ZIndex'$ IC3 | 0.728±0.188 | 5 | 252 | 18 | 21 |
| 4 | $ZIndex'$ ICV3 | 0.721±0.177 | 8 | 258 | 21 | 23 |
| 5 | $ZIndex'$ $\hat{PC}$ | 0.719±0.204 | 3 | 285 | 30 | 35 |
| 6 | $ZIndex'$ $\hat{NPC}$ | 0.719±0.204 | 2 | 286 | 31 | 36 |
| 7 | $CIndex'$ ICV3 | 0.713±0.151 | 28 | 21 | 33 | 27 |
| 8 | $ZIndex'$ $\hat{NPO}$ | 0.709±0.205 | 7 | 278 | 32 | 38 |
| | | ⋮ | | | | |
| 37 | Q | 0.62±0.139 | 42 | 167 | 56 | 47 |
| Far Far Results | | | | | | |
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | $ZIndex'$ ICV2 | 0.834±0.062 | 9 | 464 | 5 | 3 |
| 2 | $ZIndex'$ IC3 | 0.832±0.06 | 7 | 469 | 4 | 2 |
| 3 | $ZIndex'$ TO | 0.825±0.098 | 22 | 423 | 29 | 27 |
| 4 | $ZIndex'$ ICV3 | 0.823±0.063 | 12 | 458 | 6 | 6 |
| 5 | $ZIndex'$ $\hat{TO}$ | 0.823±0.096 | 18 | 446 | 27 | 25 |
| 6 | $ZIndex'$ $\hat{NPC}$ | 0.822±0.083 | 2 | 502 | 11 | 10 |
| 7 | $ZIndex'$ $\hat{PC}$ | 0.822±0.083 | 3 | 501 | 12 | 11 |
| 8 | $ZIndex'$ PC | 0.817±0.09 | 11 | 479 | 23 | 19 |
| | | ⋮ | | | | |
| 31 | Q | 0.581±0.155 | 95 | 368 | 69 | 32 |

***Table 2.5:*** *Overall ranking and difficulty analysis of the synthetic results. Here communities are well-separated with mixing parameter of .1. Similar to the last experiment, reported result are based on AMI and the Spearman's correlation.*

| Overall Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | Q | 0.854±0.039 | 11 | 1 | 4 | 2 |
| 2 | $ZIndex'$ M | 0.839±0.067 | 2 | 5 | 1 | 1 |
| 3 | $ZIndex'$ A | 0.813±0.071 | 4 | 11 | 3 | 3 |
| 4 | $ZIndex'$ $\hat{MM}$ | 0.785±0.115 | 1 | 63 | 2 | 4 |
| 5 | $ZIndex'$ $\hat{A}$ | 0.767±0.101 | 3 | 86 | 5 | 5 |
| 6 | $ZIndex'$ $\hat{PC}$ | 0.748±0.19 | 5 | 108 | 7 | 7 |
| 7 | $ZIndex'$ $\hat{NPC}$ | 0.748±0.19 | 6 | 109 | 8 | 8 |
| 8 | $ZIndex'$ $\hat{NPO}$ | 0.745±0.191 | 7 | 110 | 9 | 9 |
| 9 | $ZIndex'$ $\hat{TO}$ | 0.738±0.197 | 13 | 88 | 16 | 15 |
| 10 | $ZIndex'$ $\hat{NOV}$ | 0.738±0.197 | 8 | 134 | 10 | 10 |

| Near Optimal Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | $ZIndex'$ M | 0.825±0.105 | 1 | 1 | 1 | 1 |
| 2 | $ZIndex'$ A | 0.8±0.184 | 2 | 2 | 2 | 2 |
| 3 | $ZIndex'$ $\hat{MM}$ | 0.768±0.166 | 3 | 4 | 3 | 3 |
| 4 | $ZIndex'$ $\hat{A}$ | 0.76±0.192 | 4 | 6 | 4 | 4 |
| 5 | Q | 0.72±0.209 | 34 | 3 | 34 | 34 |
| 6 | $ASWC0$ $\hat{NPL}2$ | 0.719±0.248 | 22 | 8 | 5 | 5 |
| 7 | $SWC0$ $\hat{NPL}2$ | 0.718±0.247 | 23 | 9 | 6 | 6 |
| 8 | $ZIndex'$ $\hat{NPE}2$ | 0.714±0.259 | 5 | 21 | 7 | 8 |
| 9 | $ASWC0$ SP | 0.71±0.286 | 28 | 5 | 29 | 26 |
| 10 | $ZIndex'$ $\hat{NPL}2$ | 0.702±0.261 | 6 | 29 | 13 | 18 |

| Medium Far Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | Q | 0.578±0.124 | 106 | 22 | 3 | 1 |
| 2 | $CIndex'$ $\hat{NPC}$ | 0.522±0.146 | 154 | 12 | 78 | 69 |
| 3 | $CIndex'$ $\hat{PC}$ | 0.521±0.146 | 155 | 13 | 79 | 70 |
| 4 | $CIndex'$ $\hat{NPO}$ | 0.519±0.142 | 176 | 5 | 120 | 100 |
| 5 | $CIndex'$ $\hat{NOV}$ | 0.501±0.14 | 209 | 4 | 142 | 135 |
| 6 | $ZIndex'$ M | 0.498±0.199 | 4 | 364 | 2 | 2 |
| 7 | $CIndex'$ IC2 | 0.492±0.146 | 227 | 9 | 176 | 173 |
| 8 | $CIndex'$ ICV2 | 0.483±0.193 | 149 | 79 | 119 | 115 |
| 9 | $CIndex'$ IC3 | 0.478±0.191 | 187 | 43 | 148 | 146 |
| 10 | $CIndex'$ TO | 0.478±0.175 | 179 | 31 | 204 | 203 |

| Far Far Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | $ZIndex'$ $\hat{PC}$ | 0.527±0.169 | 61 | 501 | 5 | 4 |
| 2 | $ZIndex'$ $\hat{NPC}$ | 0.527±0.169 | 62 | 502 | 6 | 5 |
| 3 | Q | 0.523±0.192 | 128 | 73 | 93 | 25 |
| 4 | $ZIndex'$ M | 0.522±0.121 | 77 | 465 | 8 | 2 |
| 5 | $ZIndex'$ $\hat{NPO}$ | 0.518±0.168 | 63 | 504 | 10 | 6 |
| 6 | $ZIndex'$ $\hat{NOV}$ | 0.515±0.166 | 60 | 518 | 11 | 7 |
| 7 | $ZIndex'$ $\hat{TO}$ | 0.489±0.171 | 78 | 485 | 15 | 9 |
| 8 | $ZIndex'$ $\hat{NPE}2$ | 0.481±0.168 | 79 | 491 | 24 | 14 |
| 9 | $ZIndex'$ $\hat{MM}$ | 0.48±0.15 | 30 | 553 | 2 | 3 |
| 10 | $ZIndex'$ $\hat{NO}$ | 0.48±0.17 | 43 | 552 | 7 | 8 |

***Table 2.6:*** *Overall ranking of criteria based on AMI & Spearman's Correlation on the synthetic benchmarks with the same parameters as in Table 2.4 but much higher mixing parameter, .4. We can see that in these settings, modularity Q overall outperforms the ZIndex while the latter is significantly better in differentiating finer results near optimal.*

## 2.5 Summary and Future Perspectives

In this chapter, we examined different approaches for evaluating community mining results. Particularly, we examined different relative measures for clustering validity and adapted these for community mining evaluation. Our main contribution is the generalization of the well-known clustering criteria, which are originally proposed for evaluating quality of clusters of data points represented by attributes. The first reason for this generalization is to adapt these criteria in the context of interrelated data, where the only commonly used criterion to evaluate the goodness of detected communities is the modularity Q . Providing a more extensive set of validity criteria enables researchers to better evaluate and compare community mining results in different settings. In our experiments, several of these adapted criteria exhibit high performances on ranking different partitionings of a given dataset, which makes them useful alternatives for the modularity Q . Particularly, the *ZIndex* criterion exhibits good performance almost regardless of the choice of the proximity measure. This makes *ZIndex* also an attractive objective for finding communities. This is an interesting direction for the future work.

Our results suggests that the performances of different criteria and their rankings changes in different settings. Here we examined the effects of how well-separated are the communities in the ground-truth and also the general distance of a clustering from the ground-truth. We further observed that the quality of different criteria is also affected by the choice of benchmarks: Synthetic v.s. Real benchmarks. This difference motivates further investigation in order to produce more realistic synthetic generators, we cover this in Chapter 4. Another direction is to *classify the criteria* according to their performance based on different network characteristics; Onnela et al. [112], Sallaberry et al. [144] provide examples of network characterization. Another factor which affected the ranking of criteria is the choice of the agreement measure. Although the ranking based on different agreement indexes correlates, there is also significant disagreement between these measures. This calls for a closer look into the behaviour and properties of these measures, which we address in Chapter 3.

Another line of work following this chapter is to provide extensions of the criteria and measures defined here for more general cases of community mining: *overlapping communities*, *dynamic communities* and also *local communities*. For example in the literature on cluster analysis, there are clustering algorithms and validation indexes specially designed to deal with data involving overlapping categories. In particular, fuzzy clustering algorithms produce clustering results in which data objects may belong to multiple clusters at different degrees [13, 43, 62]. In order to evaluate the results of such algorithms, a number of relative, internal, and external fuzzy clustering validation indexes have been proposed [20, 21, 30, 43, 60, 62].

# Chapter 3

# Comparing Modular Structure of Networks

A measure of distance between two clusterings has important applications, including clustering validation, ensemble clustering (to aggregate multiple clusterings), and robustness analysis (to assess changes in clusterings due to fluctuations). Generally, such distance measure provides navigation through the space of possible clusterings. Mostly used in cluster validation, a normalized clustering distance, a.k.a. agreement measure, compares a given clustering result against the ground-truth clustering. In this chapter, we study different clustering agreement indexes. The two widely-used clustering agreement measures are Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). Here, we present generalized formulations from which these two measures can be derived. Unlike the original formulation of these measures, our generalizations naturally extend to overlapping and/or structured cases. In other words, the extended measures from our formulations can incorporate the structure of the data, whilst being applicable to overlapping clusterings which are common in networks. This is in particular important in comparing modular structure of networks, *i.e.*, measuring the (dis)agreement of clusterings/communities in networks. The implications are however broader since our generalizations can be used to derive new indexes, hence introducing a family of clustering agreement indexes, which are not constrained to overlapping or disjoint cases. This chapter has been published in [123].

## 3.1 Introduction

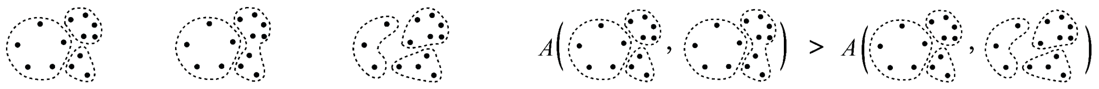
A cluster distance, accordance, similarity, or divergence has different applications. *Cluster validation* is the most common usage of cluster distance measures. In particular, in external evaluation, a clustering algorithm is validated on a set of benchmark datasets by comparing the similarity of its results against the ground-truth clusterings. Another notable application is *ensemble, or consensus clustering*, where results of different clustering algorithms on the same dataset are aggregated. A notion of distance between alternative clusterings is used in modeling and formulating this aggregation, *i.e.*, to find a clustering that has the minimum average distance to the alternative clusterings[1]. Another closely related application is multi-view clustering [35], where the objective is to find different clusterings of the same dataset, which are usually in different sub-spaces of the data, and could represent different views of that dataset. In the same context, one might be interested to find the sub-spaces that result in different/similar clusterings. The *stability and robustness* of clustering algorithms can also be assessed based on the similarity of their results when introducing noise/variations/sampling, and/or changing the order of the data [7].

Clustering distance measures are well-studied and widely-used in cluster validation, where they measure the (dis)agreement between clustering results and the ground-truth clustering. The clustering agreement measures are often classified into three main families of set matching, pair counting, and information theoretic indexes. The former class of indexes are less favoured as they suffer from the "problem of matching"[2] [96]; whereas the representatives of the latter two classes: Adjusted Rand Index (ARI) [64] and Normalized Mutual Information (NMI) [157], respectively, are the most commonly used indexes for comparing disjoint clusterings, a.k.a. partitionings. In this chapter, we demonstrate the latter two families are measuring the agreements based on the same principle. In more detail, these measures are all defined based on the contingency table of the two given clusterings, *i.e.*, their pair-wise cluster overlaps; and they measure the average dispersion in this table. More specifically, we present a *generalized clustering distance* defined based on the contingency table, which has a generative function $\varphi$. We show that, although classified differently, the representatives of both these families can be generated from the proposed generalized distance, using specific $\varphi$ functions. Moreover, unlike the original definitions, this generalized formula does not require the clusters to be disjoint and nor does it require them to cover all the data-points; the latter is particularly useful in case there are outliers, or missing cluster labels. The former, however, does not help extending to overlapping cases. In fact, there is an inherent difficulty in extending any contingency based formula for *overlapping clusters*; since the contingency table can not differentiate between the natural overlaps in the data and the cluster overlaps used for measuring the (dis)agreement. To tackle this issue, we propose to measure the agreements between two given clusterings directly based on the co-memberships of data-points in their clusters, instead of the overlaps between their clusters. More specifically, we define the *Clustering Co-Membership Difference Matrix ($\Delta$)*, based on which the clustering distance could be quantified. In particular, we present two normalized forms for $\Delta$, denoted by $RI_\delta$ and $ARI_\delta$, which are overlapping counterparts for *RI* and *ARI*, and will reduce to the original measures in case of disjoint clusters.

This algebraic overlapping ARI extension, although accurately extends the ARI for overlapping cases, is based on matrix representation of the data, and matrix multiplication, which makes

[1] Refer to Aggarwal and Reddy [3], Chapter 23 on clustering validation measures (in particular the section on external clustering validation measures); and Chapter 22 on cluster ensembles (in particular the section on measuring similarity between clustering solutions).

[2] Which is discussed in Section 3.3.2.



***(a)*** *Three clusterings of a dataset with 13 data points.* ***(b)*** *Comparing the pair-wise similarity of clusterings.*

***Figure 3.1:*** *A clustering agreement measure compares the pair-wise similarity of clusterings; visualized for a toy example. Based on A, the first pair of clusterings are more similar when compared to the second pair.*

this extension computationally expensive and not scalable to most cases. Hence, we further reformulate this measure, which evolves into a general clustering agreement formula that naturally extends to overlapping clusters. The algebraic overlapping ARI can be derived from this new general formula, called *CAI*; whereas unlike the previous measure, *CAI* is not based on a matrix representation of the data, and hence is much more efficient and practical. Moreover, *CAI* is more generalized and can be used to construct new agreement indexes. For instance, here we introduce a novel overlapping extension of *NMI* derived from *CAI*, which is more appropriate in comparing overlapping clusterings when compared to the currently available extensions proposed by Lancichinetti et al. [78] and McDaid et al. [95]. Our formulation reduces to the original formula if clusterings are disjoint, while it does not restrict any condition on the clusterings, hence works for disjoint, fuzzy, or crisp overlapping clusters.

This is in particular important since in the last decade the clustering agreement indexes have been applied extensively in the comparison of community mining algorithms [38, 59, 76]. Clusters in networks, a.k.a. communities, are shown to be *highly overlapping* [55, 88]. However the current extensions of the clustering agreement indexes for overlapping cases, which are used in the evaluation of overlapping community detection methods, are either inaccurate or inefficient. The two overlapping extensions for NMI proposed by Lancichinetti et al. [78] and McDaid et al. [95], are both defined based on the best matching between the clusters of the two clusterings, and hence fail to measure the agreement accurately particularly when the matching is not perfect. For instance in Figure 3.1, the original ARI and NMI indexes (which are defined only for disjoint cases) both agree with A, however the set matching measures, including the overlapping NMI extensions [78, 95], suggest the opposite. This "problem of matching", coined by Meilă [96], is present in all matching based agreement indexes. Other recent overlapping indexes that are defined based on the matching are the Balanced Error Rate with alignment introduced by Yang and Leskovec [167], as well as the average $F1$ score, and Recall measures used by Mcauley and Leskovec [93].

Last but not the least, a community mining algorithm clusters nodes in a given network, based on the "relationships" between them. However all the current clustering agreement measures only consider memberships of data-points in clusters, and overlook any relations between the data-points or any attributes associated with them. In this chapter, we also discuss the effect of neglecting these relations, *i.e.*, links in the networks, and derive extensions of our generalized formulae which incorporate the structure of the data in measuring clustering agreements.

## 3.2 Overview of Clustering Agreement Measures

There are several measures defined to examine the similarity, a.k.a. agreement, between two partitionings of the same dataset. More formally, let $D$ denote a dataset of $n$ data items, *i.e.*, $D = \{d_1, d_2, d_3 \dots d_n\}$. Consider $U$ and $V$ as two disjoint clusterings, a.k.a. partitionings, of $D$, that cluster $D$ into respectively $k$ and $r$ mutually disjoint groups, *i.e.*, $U = \{U_1, U_2 \dots U_k\}$; where $D = \cup_{i=1}^{k} U_i$ and $U_i \cap U_j = \emptyset,\ \forall i \neq j$; and similarly $V = \{V_1, V_2 \dots V_r\}$; where $D = \cup_{i=1}^{r} V_i$ and $V_i \cap V_j = \emptyset,\ \forall i \neq j$. Note that there is no constraint on the number of clusters by the two partitioning, *i.e.*, $k$ and $r$ might be, and are in most cases, different.

### 3.2.1 Pair Counting Measures

Clustering agreement measures are originally introduced based on counting the pairs of data items that are in the same/different partition in $U$ and $V$. In more detail, each pair of data items, $(d_i, d_j)$, is classified into one of four groups based on their co-memberships in $U$ and $V$; which results in the following pair-counts.

| | Same in $V$ | Different in $V$ |
|---|---|---|
| Same in $U$ | $M_{11} = TP$ | $M_{10} = FP$ |
| Different in $U$ | $M_{01} = FN$ | $M_{00} = TN$ |

Here, $M_{11}/M_{00}$ counts the number of pairs that are in the same/different partitions in both $U$ and $V$. $M_{10}/M_{01}$ sums up those that belong to the same/different partitions in $U$ but are in different/same partitions according to $V$. Note that $M_{11} + M_{00} + M_{10} + M_{01} = \binom{n}{2}$. When one of these partitionings, for instance $V$, is the true partitioning, *i.e.*, the ground-truth, these pair-counts are also referred to as the true/false positive/negative scores[3]. Counting these pairs requires $O(n^2)$ operations, however, these pair-counts could be derived using the contingency table a.k.a. confusion table [64], which only looks at the pairwise overlaps of the clusters and hence is more efficient. The following $k \times r$ matrix represents the contingency table of $U$ and $V$, where the element at $(i, j)$ denotes the size of overlap between $U_i$ and $V_j$, *i.e.*, $n_{ij} = |U_i \cap V_j|$.

| | $V_1$ | $V_2$ | $\dots$ | $V_r$ | marginal sums |
|---|---|---|---|---|---|
| $U_1$ | $n_{11}$ | $n_{12}$ | $\dots$ | $n_{1r}$ | $n_{1.}$ |
| $U_2$ | $n_{21}$ | $n_{22}$ | $\dots$ | $n_{2r}$ | $n_{2.}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $U_k$ | $n_{k1}$ | $n_{k2}$ | $\dots$ | $n_{kr}$ | $n_{k.}$ |
| marginal sums | $n_{.1}$ | $n_{.2}$ | $\dots$ | $n_{.r}$ | $n$ |

The last row and column are the marginal sums, *i.e.*, $n_{i.} = \sum_j n_{ij}$, and $n_{.j} = \sum_i n_{ij}$. And since clusters are disjoint, we further have $n_{i.} = |U_i|$, and $n_{.j} = |V_j|$. The pair counts can then be

[3] These pair-counts also are denoted by $a$, $b$, $c$, $d$ letters for the notational convenience in some literature, *e.g.*, [64].

computed using the following formulae.

$$M_{10} = \sum_{i=1}^{k} \binom{n_{i.}}{2} - \sum_{i=1}^{k}\sum_{j=1}^{r} \binom{n_{ij}}{2}, \quad M_{01} = \sum_{j=1}^{r} \binom{n_{.j}}{2} - \sum_{i=1}^{k}\sum_{j=1}^{r} \binom{n_{ij}}{2}$$

$$M_{11} = \sum_{i=1}^{k}\sum_{j=1}^{r} \binom{n_{ij}}{2}, \quad M_{00} = \binom{n}{2} + \sum_{i=1}^{k}\sum_{j=1}^{r} \binom{n_{ij}}{2} - \sum_{i=1}^{k} \binom{n_{i.}}{2} - \sum_{j=1}^{r} \binom{n_{.j}}{2}$$

A variety of clustering agreement measures are defined based on these pair-counts [5, 92]. Albatineh et al. [5] provides a complete survey, and here we briefly cover the most common ones. Considering co-membership of data-points in the same or different clusters as a binary variable,

***Jaccard*** agreement between clustering $U$ and $V$ is defined as:

$$J = \frac{TP}{(FP + FN + TP)} = \frac{M_{11}}{(M_{01} + M_{10} + M_{11})} \tag{3.1}$$

***Rand Index*** is defined similarly to Jaccard, but it also values pairs that belong to different clusters in both partitionings, *i.e.*,

$$\begin{aligned} RI &= \frac{TP + TN}{TP + FP + FN + TN} = \frac{(M_{11} + M_{00})}{(M_{11} + M_{01} + M_{10} + M_{00})} \\ &= 1 + \frac{1}{n^2 - n}(2\sum_{i=1}^{k}\sum_{j=1}^{r} n_{ij}^2 - (\sum_{i=1}^{k} n_{i.}^2 + \sum_{j=1}^{r} n_{.j}^2)) \end{aligned} \tag{3.2}$$

***Mirkin Index*** is a transformation of the Rand Index, defined as $n(n-1)(RI-1)$, which is equivalent to $RI$ when comparing partitionings of the same dataset [164].

***F-measure*** is a weighted mean of the precision, and recall, *i.e.*,

$$P = \frac{M_{11}}{(M_{11} + M_{10})}, \quad R = \frac{M_{11}}{(M_{11} + M_{01})}, \quad F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{3.3}$$

where $\beta$ determines the importance of recall w.r.t. precision. The two common values for $\beta$ are 2 and $\frac{1}{2}$; the former weighs recall higher than precision while the latter favours the precision more. The precision and recall in F-measure are same as the two nonsymmetric Wallace [158] measures proposed for partition correspondence. While their geometric mean defines the Fowlkes and Mallows [50] measure, *i.e.*, $FM = \sqrt{PR}$.

A clustering agreement measure is desired to return the same value[4], usually zero, for agreement no better than random [64, 157]. *Correction for chance* is adjusting a measure to have a

[4] *i.e.*, a constant baseline, the expected value of agreements between two random clusterings of a same dataset. If not constant, an example of 0.7 agreement value can be both a strong (when baseline is 0.2) or a weak (when baseline is 0.6) agreement.

constant expected value for agreements due to chance. This adjustment is done based on an upper bound on the measure, $Max[M]$, and its expected value, $E[M]$, using the following formula:

$$AM = \frac{M - E[M]}{Max[M] - E[M]} \tag{3.4}$$

The ***Adjusted Rand Index (ARI)*** is proposed by Hubert and Arabie [64], in order to adjust RI (Equation 3.2) for chance. *ARI* assumes that the contingency table is constructed randomly when the marginals are fixed, *i.e.*, the size of the clusters in $U$ and $V$ are fixed. With this assumption, *RI* is a linear transformation of $\sum_{i,j}\binom{n_{ij}}{2}$, and $E\left(\sum_{i,j}\binom{n_{ij}}{2}\right) = \sum_i \binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}/\binom{n}{2}$. Hence, adjusting *RI* with upper bound 1 results in the following formula:

$$ARI = \frac{\sum\limits_{i=1}^{k}\sum\limits_{j=1}^{r}\binom{n_{ij}}{2} - \sum\limits_{i=1}^{k}\binom{n_{i.}}{2}\sum\limits_{j=1}^{r}\binom{n_{.j}}{2}/\binom{n}{2}}{\frac{1}{2}[\sum\limits_{i=1}^{k}\binom{n_{i.}}{2} + \sum\limits_{j=1}^{r}\binom{n_{.j}}{2}] - \sum\limits_{i=1}^{k}\binom{n_{i.}}{2}\sum\limits_{j=1}^{r}\binom{n_{.j}}{2}/\binom{n}{2}} \tag{3.5}$$

which returns 0 for agreements no better than random and ranges between $[-1, 1]$. There is also an approximate formulation [5, 64] for this expectation defined as $E(\sum_{i,j} n_{ij}^2) = \sum_i n_{i.}^2 \sum_j n_{.j}^2/n^2$, which results in a slightly different formula for the *ARI*, *i.e.*,

$$ARI' = \frac{\sum\limits_{i=1}^{k}\sum\limits_{j=1}^{r}{n_{ij}}^2 - \sum\limits_{i=1}^{k}{n_{i.}}^2\sum\limits_{j=1}^{r}{n_{.j}}^2/n^2}{\frac{1}{2}[\sum\limits_{i=1}^{k}{n_{i.}}^2 + \sum\limits_{j=1}^{r}{n_{.j}}^2] - \sum\limits_{i=1}^{k}{n_{i.}}^2\sum\limits_{j=1}^{r}{n_{.j}}^2/n^2} \tag{3.6}$$

There also exist several other variations of pair counting agreement measures, defined in terms of $M_{00}, M_{01}, M_{10}, M_{01}$, such as Gamma, Hubert, Pearson, etc. However, it has been shown that these measures become similar or even equivalent after correction for chance [5]. More specifically, Albatineh et al. [5] show that many of these measures are linear transformations of $\sum_{i,j} n_{ij}^2$, which are known as the $\mathcal{L}$ family, *i.e.*, each measure could be written as $\alpha + \beta\sum_{i,j} n_{ij}^2$, where $\alpha$ and $\beta$ depend on the marginal counts, $n_{i.}$ or $n_{.j}$, but not on the $n_{ij}$. For example for the Rand Index we have: $\alpha = 1 - \frac{1}{n(n-1)}(\sum_i n_{i.}^2 + \sum_j n_{.j}^2)$, and $\beta = 2/n(n-1)$. They further prove that the $\mathcal{L}$ family measures become equivalent if their $\frac{1-\alpha}{\beta}$ ratio is the same, since their corrected for chance formula will all be as:

$$\frac{\sum_{i,j} n_{ij}^2 - E(\sum_{i,j} n_{ij}^2)}{\frac{1-\alpha}{\beta} - E(\sum_{i,j} n_{ij}^2)}$$

It is worth to mention that the pair counting measures are closely related to the *statistical inter-rater agreement* indices. The inter-rater agreement indices in statistics are defined to measure the agreement between different coders, or judges on categorizing the same data. Examples are the goodness of fit, chi-square test, the likelihood chi-square, kappa measure of agreement, Fisher's exact test, Krippendroff's alpha; which are used in a different settings based on the number of coders, number of data-points, number of categories, balance in the categories, etc. [31]. These statistical tests are also defined based on the contingency table which displays the multivariate frequency distribution of the (categorical) variables. Warrens [159] shows that the pair counting clustering agreement measures become equivalent to one of the statistical inter-rater agreement indices after correction for chance. In particular, the equivalence of *Cohen's kappa*, one the most widely used inter-rater agreement index, and the *ARI* is proved by Warrens [160]. Cohen's kappa is a chance corrected index of association defined for assessing the agreement between two raters, who categorize data into $k$ categories, which is formulated as:

$$\kappa = \frac{[\sum_{i=j}^{k} n_{ij} - \sum_{i=j}^{k} E_{ij}]}{[n - \sum_{i=j}^{k} E_{ij}]}\ , \quad where \quad E_{ij} = \frac{n_{i.}n_{.j}}{n} \tag{3.7}$$

### 3.2.2 Information Theoretic Measures

Another commonly used family of clustering agreement measures are the information theoretic based measures, defined based on *mutual information* between the two clusterings. These measures consider the overlaps between clusters in $U$ and $V$, as a joint distribution of two random variables, *i.e.*, the cluster memberships in $U$ and $V$. The entropy of cluster $U$, $H(U)$, the joint entropy of $U$ and $V$, $H(U,V)$, and their mutual information, $I(U,V)$ are then defined as:

$$H(U) = -\sum_{i=1}^{k} \frac{n_{i.}}{n} \log(\frac{n_{i.}}{n}), \quad H(V) = -\sum_{j=1}^{r} \frac{n_{.j}}{n} \log(\frac{n_{.j}}{n})$$
$$H(U,V) = -\sum_{i=1}^{k}\sum_{j=1}^{r} \frac{n_{ij}}{n} \log(\frac{n_{ij}}{n}), \quad I(U,V) = \sum_{i=1}^{k}\sum_{j=1}^{r} \frac{n_{ij}}{n} \log(\frac{n_{ij}/n}{n_{i.}n_{.j}/n^2})$$

Particularly, Meilă [96] proposed the ***Variation of Information (VI)***, for comparing two different clusterings as:

$$VI = \sum_{i=1}^{k}\sum_{j=1}^{r} \frac{n_{ij}}{n} \log(\frac{n_{i.}n_{.j}/n^2}{n_{ij}^2/n^2}) \tag{3.8}$$

The pair counting measures overviewed above, except Mirkin, have a fixed range of $[0,1]$. The information theoretic measures, however, do not have a fixed range; for instance the mutual information varies between $(0, logk]$, and the variation of information between $[0, 2\log\max(k,r)]$ [164]. We often require to compare and average agreements of different clustering methods over different datasets, and therefore a *normalized* index is preferred. Consequently, different normalized forms for the mutual information are defined for comparing clusterings; refer to [157] for a survey. The most commonly used normalization forms are:

$$NMI_{\Sigma} = \frac{2\,I(U,V)}{H(U)+H(V)} \quad \text{and} \quad NMI_{\surd} = \frac{I(U,V)}{\sqrt{H(U)H(V)}} \tag{3.9}$$

As we can see in the experiments in Section 3.6.1, these two variations exhibit the same behaviour in practice. Correction for chance of the information theoretic measures is discussed by Vinh et al. [156]. More specifically, they propose *Adjusted Mutual Information*, using Equation 3.4, as:

$$AMI = \frac{I(U,V) - E[I(U,V)]}{Max[I(U,V)] - E[I(U,V)]} \tag{3.10}$$

where different forms of *AMI* are derived using different upper bounds on $I$ as $Max[I]$, which are:

$$I(U,V) \leq \min(H(U),H(V)) \leq \sqrt{H(U)H(V)} \leq \frac{H(U)+H(V)}{2} \leq \max(H(U),H(V)) \leq H(U,V)$$

In particular, $AMI_{\sum}$, *i.e.*, *AMI* with upper bound of $\frac{1}{2}(H(U)+H(V))$, is equivalent to the Adjusted form for Variation of Information. On the other hand, the expected value, $E[I]$, is derived assuming the sizes of the clusters are fixed, *i.e.*, similar to the *ARI*'s assumption on the hypergeometric model of randomness, as:

$$E[I(U,V)] = \sum_{i,j} \sum_{\mathfrak{m}=\max(n_{i.}+n_{.j}-n,1)}^{min(n_{i.},n_{.j})} \frac{\mathfrak{m}}{n} \log(\frac{n\mathfrak{m}}{n_{i.}n_{.j}}) \frac{n_{i.}!n_{.j}!(n-n_{i.})!(n-n_{.j})!}{n!\mathfrak{m}!(n_{i.}-\mathfrak{m})!(n_{.j}-\mathfrak{m})!(n-n_{i.}-n_{.j}+\mathfrak{m})!}$$

This formulation includes big factorials, therefore is computationally complex; which makes *AMI* less practical when compared to the *ARI*.

All the measures discussed above are only valid when clusters are disjoint, and also ignore any structure in the data. In the following, we first discuss how these measures are related, and then discuss how to extend them for overlapping and structured cases.

## 3.3 Generalization of Clustering Agreement Measures

Both families of pair counting and information theoretic measures quantify the agreement between two clusterings based on their contingency table. Here, we generalize them, and show that they are both measuring the (normalized) sum of the divergences in rows (and columns) of this table; whereas the perfect agreement occurs if the sum is zero[5]. Our generalization is symmetric and is defined based on the relation between the Rand Index (*RI*) and the Variation of Information ($VI$), which are respectively a representative for the pair counting and information theoretic families.

[5] Which happens when the clusterings are identical, and only the order of the clusters is permuted, *i.e.*, the distribution of overlaps in each row/column of the contingency table has a single spike on the matched cluster and is zero elsewhere.

**Proposition 3.3.1.** *VI (RI) of two partitionings is proportional to the conditional entropies (variances) of memberships in them,* i.e., $VI(U,V) = H(U|V) + H(V|U)$ *and* $RI(U,V) \propto Var(U|V) + Var(V|U)$*; see Appendix B.1.1 for proof.*

Based on this proposition, we define the generalized distance for clusterings as:

**Definition 3.3.1.** Generalized Clustering Distance ($\mathcal{D}$)

$$\mathcal{D}^{\eta}_{\varphi}(U,V) = \mathcal{D}^{\eta}_{\varphi}(U||V) + \mathcal{D}^{\eta}_{\varphi}(V||U), \quad \mathcal{D}^{\eta}_{\varphi}(U||V) = \sum_{v\in V}\left[\varphi(\sum_{u\in U}\eta_{uv}) - \sum_{u\in U}\varphi(\eta_{uv})\right]$$

where $\eta_{uv}$ quantifies the similarity between the two clusters of $u \in U$ and $v \in V$, *i.e.*, $\eta : 2^V \times 2^U \rightarrow \mathbb{R}$; and $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ is a non-linear function, such that $\varphi(\sum x) \neq \sum \varphi(x)$, which is used to quantify the divergence or dispersion in a set of numbers.

This generalized formula can be extended to define novel clustering distances, using the flexibility that the $\varphi$ and $\eta$ functions provide. For example, we introduce an extension of this generalization for clusterings of nodes in graphs, a.k.a. communities, in the following section. More specifically, $\eta$ is any function that transforms the two given clusterings into a contingency table. The distance between the clusterings is then measured by adding up the dispersion in each row (and column) of this table. Function $\varphi$ is used to quantify how disperse are the values in each row (and column), *i.e.*, to measure the divergence from a spike distribution, observed if the corresponding clusters are perfectly matched.

**Corollary 3.3.2.** *$\mathcal{D}$ is bounded if $\varphi$ is a positive superadditive function,* i.e., $\varphi(x) \geq 0 \wedge \varphi(x+y) \geq \varphi(x) + \varphi(y) \implies 0 \leq \mathcal{D}^{\eta}_{\varphi}(U||V) \leq \varphi(\sum_{v\in V}\sum_{u\in U}\eta_{uv})$*; see proof in Appendix B.1.2.*

Using this bound as a normalizing factor, we define:

**Definition 3.3.2.** Normalized Generalized Clustering Distance ($\mathcal{ND}$)

$$\mathcal{ND}^{\eta}_{\varphi}(U,V) = \frac{\mathcal{D}^{\eta}_{\varphi}(U,V)}{NF(U,V)}, \quad NF(U,V) = \varphi(\sum_{v\in V}\sum_{u\in U}\eta_{uv})$$

Here, we first show that both the Rand Index (*RI*) and the (normalized) Variation of Information ($VI$) generate from this normalized distance. Then, we introduce an adjusted form for $\mathcal{D}$, and show that similarly, the *ARI* and $NMI$ both derive from the adjusted form. More specifically,

**Identity 3.3.3.** *The Variation of Information (Equation 3.8) derives from $\mathcal{ND}$ if we set $\varphi(x) = x\log x$, and $\eta$ as the overlap size: $\eta_{uv} = |u \cap v|$ (proof in Appendix B.1.3),* i.e.,

$$\mathcal{ND}^{|\cap|}_{x\log x}(U,V) \equiv \frac{VI(U,V)}{\log n}$$

**Identity 3.3.4.** *The Rand Index (Equation 3.2) derives from $\mathcal{ND}$ if we set $\varphi(x) = \binom{x}{2}$, and $\eta$ as the overlap size (proof in Appendix B.1.4),* i.e.,

$$\mathcal{ND}^{|\cap|}_{\binom{x}{2}}(U,V) \equiv 1 - RI(U,V), \;\; also \quad \mathcal{ND}^{|\cap|}_{x^2}(U,V) \equiv 1 - RI'(U,V)$$

Similar to the Identity 3.3.4, in the rest of this chapter, we consider clustering agreement ($\mathcal{I}$) and normalized distance ($\mathcal{ND}$) interchangeably, *i.e.*, using $\mathcal{I} = 1 - \mathcal{ND}$. We further adjust the generalized distance to take its maximum, *i.e.*, one, if $U$ and $V$ are independent. Assume $P_{U,V}$ as the joint probability distribution with the marginals of $P_U$ and $P_V$, as:

$$P_{U,V}(u,v) = \frac{\eta_{uv}}{\sum_{uv}\eta_{uv}}, \quad P_U(u) = \sum_v P_{U,V}(u,v) = \frac{\eta_{.v}}{\sum_{uv}\eta_{uv}}, \quad P_V(v) = \frac{\eta_{u.}}{\sum_{uv}\eta_{uv}}$$

Then the independence condition for $U$ and $V$, *i.e.*, $P_{U,V}(u,v) = P_U(u)P_V(v)$, translates into $\eta_{uv} = (\eta_{u.}\eta_{.v})/\sum_{uv}\eta_{uv}$. On the other hand, from Definition 3.3.1, we have:

$$\mathcal{D}^{\eta}_{\varphi}(U,V) = \sum_{v\in V}\varphi(\eta_{.v}) + \sum_{u\in U}\varphi(\eta_{u.}) - 2\sum_{v\in V}\sum_{u\in U}\varphi(\eta_{uv})$$

Therefore, we define the adjusted distance as:

**Definition 3.3.3.** Adjusted Generalized Clustering Distance ($\mathcal{AD}$)

$$\mathcal{AD}^{\eta}_{\varphi} = \frac{\mathcal{D}^{\eta}_{\varphi}(U,V)}{NF(U,V)}, \quad NF = \sum_{v\in V}\varphi(\eta_{.v}) + \sum_{u\in U}\varphi(\eta_{u.}) - 2\sum_{u\in U}\sum_{v\in V}\varphi\left(\frac{\eta_{.v}\eta_{u.}}{\sum\limits_{u\in U}\sum\limits_{v\in V}\eta_{uv}}\right)$$

**Identity 3.3.5.** *The Normalized Mutual Information (Equation 3.9) derives from $\mathcal{AD}$, if we set $\varphi(x) = xlogx$, and $\eta$ as the overlap size: $\eta_{uv} = |u \cap v|$ (proof in Appendix B.1.5),* i.e.,

$$\mathcal{AD}^{|\cap|}_{xlogx}(U,V) \equiv 1 - NMI_{sum}(U,V)$$

**Identity 3.3.6.** *The Adjusted Rand Index of Equation 3.5 and Equation 3.6 derive from $\mathcal{AD}$, if we set $\varphi(x) = x(x-1)$ and $\varphi(x) = x^2$ respectively, where $\eta$ is the overlap size, (proof in Appendix B.1.5),* i.e.,

$$\mathcal{AD}^{|\cap|}_{x^2}(U,V) \equiv 1 - ARI'(U,V), \quad \mathcal{AD}^{|\cap|}_{\binom{x}{2}}(U,V) \cong 1 - ARI(U,V)$$

This line of generalization is similar to the works in Bergman Divergence and $f$-divergences. For example, the mutual information and variance are proved to be special cases of Bergman information [9]. The (reverse) KL divergence and Pearson $\chi^2$ are shown to be $f$-divergences when the generator is $x \log x$ and $(x-1)^2$ respectively [110]. Beside this analogy, our generalized measure is different from these divergences. One could consider our proposed measure as an (adjusted normalized) conditional Bergman entropy for clusterings. This relation is however non-trivial and is out of scope of this thesis.

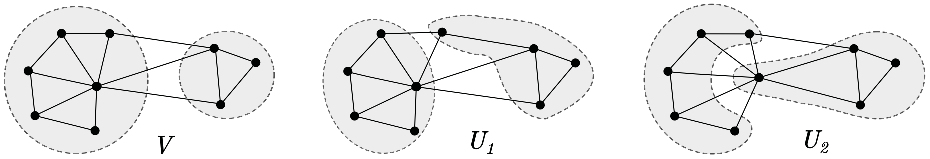


***Figure 3.2:*** *An example graph clustered in three different ways: by clustering $V$ (i.e. true clustering), and by $U_1$ and $U_2$ (i.e. two candidate clusterings). Considering only the number of nodes in the overlaps and ignoring the edges, $U_1$ and $U_2$ have the same contingency table with $V$, i.e. $|\cap|(U_1, V) = |\cap|(U_2, V) = \{\{5, 0\}, \{1, 3\}\}$. Therefore, they have the same agreement with $V$, regardless of the choice of the agreement measure: ARI, NMI, etc. However if considering the edges, $U_1$ is more similar to the true clustering $V$. This could be enforced using an alternative overlap function that incorporates edges, such as the degree weighted overlap function, by which we get: $\Sigma d(U_1, V) = \{\{18, 0\}, \{3, 9\}\}$ and $\Sigma d(U_2, V) = \{\{14, 0\}, \{7, 9\}\}$; or the edge based variation, which gives: $\xi(U_1, V) = \{\{7, 0\}, \{0, 3\}\}$ and $\xi(U_2, V) = \{\{4, 0\}, \{0, 3\}\}$.*

### 3.3.1 Extension for Inter-related Data

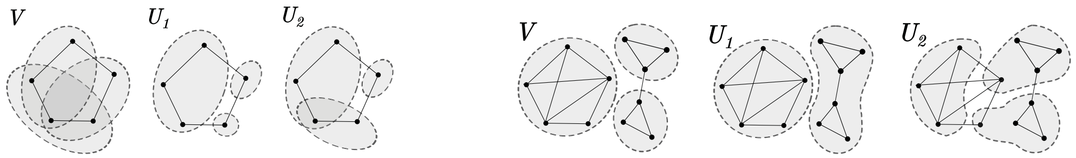
The common clustering agreement measures introduced in the previous section, only consider memberships of data-points in clusters, and *overlook the attributes of individual data-points or any relations between them.* This neglect is problematic, as also mentioned by a few previous works. For example, Zhou et al. [173] illustrate the issue of ignoring the distances between data-points, when comparing clusterings; and propose a measure which incorporates the distances between the representatives of clusters. This is in particular important when comparing *clusterings of nodes within information networks.* An information network encodes relationships between data-points, and a clustering on such network forms sub-graphs. Using the original clustering agreement measures, we only consider the nodes in measuring the clustering distance. One should however also consider edges when comparing two sub-graphs; see Figure 3.2 for a clarifying example. To incorporate the structure of the data in our generalized distance (Definition 3.3.1), we simply modify the overlap function $\eta$. The generator overlap function for the original measures (*RI*, *VI*, *ARI*, and *NMI*) is $|\cap| : \eta_{uv} = \sum_{i \in u \cap v} 1$; which counts the number of common nodes. Therefore, the first intuitive modification to incorporate the structure is to consider a degree weighted function as:

$$\Sigma d : \eta_{uv} = \sum_{i \in u \cap v} d_i \tag{3.11}$$

Using this $\eta$, well-connected nodes with higher degree weigh more in the distance. Another possibility is to alter $\eta$ to directly assess the structural similarity of these sub-graphs by counting their common edges, as:

$$\xi : \eta_{uv} = \sum_{i,j \in u \cap v} A_{ij} \tag{3.12}$$

One can consider many other alternatives for measuring the overlaps based on the application at hand. We revisit and delve deeper in this topic in Section 3.4, after providing an alternative formulation for the clustering distance or agreement measures.



***(a)*** *Omega example: the extended pair-counts matrices of $U_1$ and $U_2$ with $V$ are respectively {{3, 0, 0}, {1, 1, 1}, {2, 0, 1}}, and {{3, 0, 0}, {3, 2, 0}, {0, 2, 0}}. In the latter, the second row corresponds to the pairs of nodes which are together in one cluster in $V$; where 3, 2, and 0 of them are respectively clusters together in 0, 1, and 2 clusters of $U_2$.*

***(b)*** *Matching example: using the original formulation we have $NMI(U_1, V) = 0.78$ and $NMI(U_2, V) = 0.71$; whereas the overlapping version results in $NMI'(U_1, V) = 0.61$ and $NMI'(U_2, V) = 0.62$ with Lancichinetti et al. [78] extension and $NMI''(U_1, V) = 0.53$ and $NMI''(U_2, V) = 0.61$ with the extension proposed by McDaid et al. [95].*

***Figure 3.3:*** *Example for **the limitation of Omega index** on the left: the pair-counts table for $U_1$ and $U_2$ with $V$ have the same trace, and therefore $U_1$ and $U_2$ have the same agreement with $V$ according to the Omega index; whereas $U_2$ should have been ranked more similar. Example for **the problem of matching** on the right: using the set matching based measures, such as the overlapping version of the $NMI$, clustering $U_2$ is in higher agreement with $V$, while the non-overlapping version of $NMI$ suggests the opposite*[6].

### 3.3.2 Extension for Overlapping Clusters

There are several non-trivial extensions of the clustering agreement measures for the crisp overlapping clusters, *i.e.*, when data-points can fully belongs to multiple clusters. Notably, Collins and Dent [30] proposed the *Omega index* as a generalization of the (adjusted) rand index; which expands the 2×2 pair-counts table $\{\{M_{00}, M_{10}\}, \{M_{01}, M_{11}\}\}$; to $M_{ij}$ that counts the pairs of data-points which appeared together in $i$ clusters of $U$ and $j$ clusters of $V$. Similar to the *RI*, trace of this matrix, *i.e.*, $\sum_i M_{ii}$, gives the agreement index, which is further adjusted for chance using the marginals of $M$. The (Adjusted) Omega index reduces to the *(A)RI* if the clusterings are disjoint. However, it only considers the pairs that appeared in the *exact* same number of clusters together, and ignores the partially matched pairs. Figure 3.3a provides an example which illustrates the issue of this limitation. Another commonly used measure is *the overlapping extension of $NMI$* proposed by Lancichinetti et al. [78]. This extension does not reduce to the original $NMI$ if the clusterings are disjoint. Moreover, it assumes a matching between clusters in $U$ and $V$, and only compares the best matched clusters. Therefore, it suffers from the "problem of matching" [96], which is an inherent problem for any index defined based on the best matching of clusters; Figure 3.3b gives a visualized example. Other examples of matching based agreements are the Balanced Error Rate with alignment, average $F1$ score, and Recall measures used in [93, 95, 167]. There is also a line of work on extending the agreement indexes for fuzzy clusters with soft memberships [8, 18, 20, 66, 122]. The fuzzy measures are not applicable to cases where a data-point could fully belong to more than one cluster, *i.e.*, crisp overlappings (*e.g.*, $V$ in Figure 3.4); which are common in clusters in networks, a.k.a. communities. However, the bonding concept presented by Brouwer [18] is similar to the main idea behind our extension for overlapping cases, which we introduce in the next section.

[6] Here we used a disjoint example to be able to compare the results quantitatively with the original $NMI$; the same problem however exists for any matching based measure, regardless of the overlapping or disjoint.

The extension of the proposed $\mathcal{D}$ formula (Definitions 3.3.1, 3.3.2, and 3.3.3) for overlapping clusters is not straightforward. The $(\mathcal{A}/\mathcal{N})\mathcal{D}$ formula is indeed bounded for overlapping clusters, and reduces to the original formulation if we have disjoint covering clusters. However, the current formulation is not appropriate for comparing overlapping clusters, since it treats overlaps as variations and penalizes them. Consider an extreme example when we are comparing two identical clusterings, and therefore we should have $(\mathcal{A}/\mathcal{N})\mathcal{D} = 0$ (*i.e.*, the perfect agreement); this is true if there is no overlapping nodes, however as the number of overlapping nodes increases, $(\mathcal{A}/\mathcal{N})\mathcal{D}$ also increases (*i.e.*, the agreement decreases). This is an inherent problem in any agreement measure formulated only based on the contingency/confusion table, since overlaps in the data are confused with the overlaps between the matched clusters. We overcome this problem by proposing an alternative formulation for the clustering agreement measures, presented in the next section. More generally, the difficulty of computing the agreement of different clusterings, and in particular their extension for general cases such as overlapping clusters, comes from the fact that there is no matching between the clusters from the two clusterings. Therefore, one should consider all the permutations (*e.g.*, using the contingency table), or only consider the best matching, which is cursed with the "problem of matching" as discussed earlier. Alternatively, we propose an algebraic formulation which takes the permutation out of the equation.

## 3.4 Algebraic Formulation for Clustering Agreements

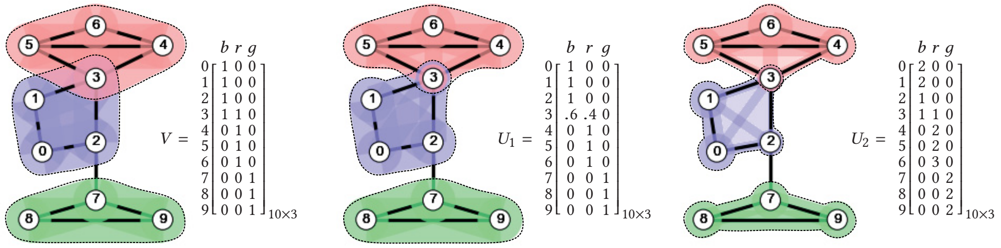
We first show that the proposed generalized formulae in Section 3.2 can be reformulated in terms of matrices. These formulae are defined based on the contingency table of $U$ and $V$, which we obtain from the $\eta$ overlap function. We can denote this contingency table with a $k \times r$ matrix, *i.e.*, $N_{k\times r}$. Then, we can rewrite $\mathcal{D}$ (Definition 3.3.1) and $\mathcal{N}\mathcal{D}$ (Definition 3.3.2) as follows:

$$\mathcal{D}_\varphi = \left[\mathbf{1}\varphi(N\mathbf{1}^T) - \mathbf{1}\varphi(N)\mathbf{1}^T\right] + \left[\varphi(\mathbf{1}N)\mathbf{1}^T - \mathbf{1}\varphi(N)\mathbf{1}^T\right], \quad \mathcal{N}\mathcal{D}_\varphi = \frac{\mathcal{D}_\varphi}{\varphi(\mathbf{1}N\mathbf{1}^T)} \tag{3.13}$$

where $\mathbf{1}$ is a vector of ones with appropriate shape so that the matrix-vector product is valid, *i.e.*, $\mathbf{1}N = [n_{.1}, n_{.2}, \dots n_{.r}]$, and $N\mathbf{1}^T = [n_{1.}, n_{2.}, \dots n_{k.}]^T$; and $\varphi$ is applied element-wise to the given matrix. In the same manner, we can reformulate $\mathcal{A}\mathcal{D}$ (Definition 3.3.3 on page 39) as:

$$\mathcal{A}\mathcal{D}_\varphi = \frac{\mathcal{D}_\varphi}{\frac{1}{2}[\mathbf{1}\varphi(N\mathbf{1}^T) + \varphi(\mathbf{1}N)\mathbf{1}^T] - E}, \quad E = \mathbf{1}\varphi(\frac{(N\mathbf{1}^T) \times (\mathbf{1}N)}{\mathbf{1}N\mathbf{1}^T})\mathbf{1}^T \tag{3.14}$$

Naturally, the identities proved in Section 3.2 still hold; for example, when $\eta$ gives the overlap sizes, the normalized Variation of Information derives from $\mathcal{D}$ with $\varphi(x) = x\log x$ (Identity 3.3.3); and $1 - \mathcal{N}\mathcal{D}_{\varphi(x)=\binom{x}{2}}$ is equivalent to the rand index (Identity 3.3.4). These formulations based on the contingency matrix, as also discussed in Section 3.3.2, are only appropriate for disjoint clusters. Therefore in the following, we propose another reformulation, which is not defined based on the



$$V = \begin{array}{c|ccc} & b & r & g \\ \hline 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 1 & 1 & 0 \\ 4 & 0 & 1 & 0 \\ 5 & 0 & 1 & 0 \\ 6 & 0 & 1 & 0 \\ 7 & 0 & 0 & 1 \\ 8 & 0 & 0 & 1 \\ 9 & 0 & 0 & 1 \end{array}_{10\times 3} \quad U_1 = \begin{array}{c|ccc} & b & r & g \\ \hline 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & .6 & .4 & 0 \\ 4 & 0 & 1 & 0 \\ 5 & 0 & 1 & 0 \\ 6 & 0 & 1 & 0 \\ 7 & 0 & 0 & 1 \\ 8 & 0 & 0 & 1 \\ 9 & 0 & 0 & 1 \end{array}_{10\times 3} \quad U_2 = \begin{array}{c|ccc} & b & r & g \\ \hline 0 & 2 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 1 & 1 & 0 \\ 4 & 0 & 2 & 0 \\ 5 & 0 & 2 & 0 \\ 6 & 0 & 3 & 0 \\ 7 & 0 & 0 & 2 \\ 8 & 0 & 0 & 2 \\ 9 & 0 & 0 & 2 \end{array}_{10\times 3}$$

***Figure 3.4:*** *Example of general matrix representation for a clustering: $V$ and $U_1$ are the classic overlapping clusters with crisp, and soft memberships respectively. Node 3 fully belongs to both blue and red clusters in $V$, wherein $U_1$, it belongs 60% to the blue cluster and 40% to the red cluster. This representation is general in a sense that it could encode membership of nodes to clusters in any form, with no assumptions on the matrix.*

contingency matrix, and is valid for both disjoint and overlapping cases. In more detail, let $U_{n\times k}$ denote a general representation for a clustering of a dataset with $n$ data-points; *i.e.*, $u_{ik}$ represents the memberships of node $i$ in the $k^{th}$ cluster of $U$. Different constraints on this representation derive different cases of clustering[7]; see Figure 3.4 for examples. Then, the contingency matrix of $U$ and $V$ obtained by $\eta = |\cap|$ (*e.g.*, in Identity 3.3.3), which indicates the size of overlaps between all pairs of clusters in $U_{d\times k}$ and $V_{d\times r}$, can also be derived from $N = (U^T V)_{k\times r} = (V^T U)^T_{k\times r}$. On the other hand, there is an analogy between co-membership and overlap, *i.e.*, $(UU^T)_{ij}$ denotes in how many clusters node $i$ and $j$ appeared together, and $(U^T U)_{ij}$ denotes how many nodes clusters $i$ and $j$ have in common. Inspired by this analogy, we propose to measure the distance between clusterings directly by comparing their co-membership matrices, *i.e.*, $(UU^T)_{n\times n}$ $v.s.$ $(VV^T)_{n\times n}$, instead of their contingency/overlap matrix, *i.e.*, $(U^T V)_{k\times r}$. More specifically, we consider the clustering co-membership difference as follows, and then show that both the rand index (*RI*) and the adjusted rand index (*ARI*) derive from different normalization of this difference.

**Definition 3.4.1.** Clustering Co-Membership Difference Matrix is $\Delta(U,V) = (UU^T - VV^T)_{n\times n}$.

To calculate the distance between $U$ and $V$, we need to quantify $\Delta$ using a matrix function: $\mathbb{R}^{n\times n} \to \mathbb{R}$, *e.g.*, a matrix norm. In particular, the *RI* and *ARI* are different normalized form of $\Delta$ when we use the Frobenius matrix norm, *i.e.*,

**Identity 3.4.1.** *The Rand Index of Equation 3.2 derives from $\Delta$,* i.e.,

$$\frac{\|\Delta(U,V)\|_F^2}{\mathfrak{m} \times n(n-1)} \equiv 1 - RI(U,V)\ , \quad also \quad \frac{\|\Delta(U,V)\|_F^2}{\mathfrak{m} \times n^2} \equiv 1 - RI'(U,V) \tag{3.15}$$

*where $\|.\|_F^2$ sums the squared values of the given matrix, a.k.a. squared Frobenius norm; and $\mathfrak{m} = [max(max(UU^T), max(VV^T))]^2$, which is equal to one for disjoint clusters.*

[7] For crisp clusters (a.k.a. strict membership), $u_{ik}$ is restricted to 0, 1 (1 if node $i$ belongs to cluster $k$ and 0 otherwise); whereas for probabilistic clusters (or soft membership), $u_{ik}$ could be any real number in $[0,1]$. Fuzzy clusters usually assume an additional constraint that the total membership of a data-point is equal to one, *i.e.*, $u_{i.} = \sum_k u_{ik} = 1$. Which should also be true for disjoint clusters, since each data-point can only belong to one cluster.

The normalization factor for the Rand Index in the Identity 3.4.1, assumes an unlikely worse case scenario when all pairs are in disagreements. The *ARI* normalization in Identity 3.4.2, on the other hand, adopts the expected difference when $UU^T$ and $VV^T$ are independent.

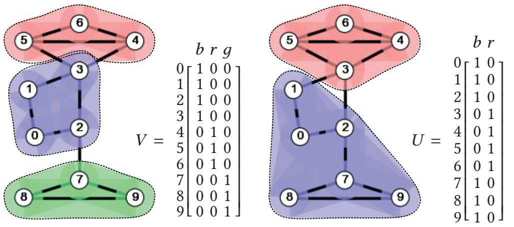
**Identity 3.4.2.** *The Adjusted Rand Index of Equation 3.6 (on page 35) derives from* $\Delta$, i.e.,

$$\frac{\|\Delta(U,V)\|_F^2}{\|UU^T\|_F^2 + \|VV^T\|_F^2 - \frac{2}{n^2}|UU^T||VV^T|} \equiv 1 - ARI'(U,V) \tag{3.16}$$

*where* $|.|$ *is the sum of all elements in the matrix.*

Appendix B.1.6 provides details on the derivation of these normalizing factors and proofs of these two Identities. The *ARI* of Equation 3.5 derives from the same formula, if we set the diagonal elements of the co-membership matrices to zero, *i.e.*, $(UU^T)' = UU^T - \mathbf{I}_n$. Since the original *ARI* formula counts only the co-memberships of different nodes, *i.e.*, $(i,j)$ where $i \neq j$; whereas, *ARI′* also considers the co-memberships for each single node with itself in different clusters, which is more suitable for overlapping cases.

The $\Delta$-based formulations for *RI* (Identity 3.4.1) and *ARI* (Identity 3.4.2), denoted respectively by $RI_\delta$ and $ARI_\delta$ hereafter, not only are identical to the original formulations if the clusterings are disjoint (see Figure 3.5 for an example), but are also valid for overlapping cases (see Figure 3.6 for examples). Unlike the prior contingency based formulations (*i.e.*, Definition 3.3.1, 3.3.2, and 3.3.3), $RI_\delta$ and $ARI_\delta$ do not need to consider all permutations of the matched clusters using their pair-wise overlaps, and hence will not confuse the natural overlaps in the data with the overlaps of matched clusters used to compute the agreements. For the extreme example discussed earlier in Section 3.3.2, $RI_\delta$ and $ARI_\delta$ always return 1 if the clusterings are identical, regardless of the amount of the overlapping nodes.



$$V = \begin{array}{c|ccc} & b & r & g \\ \hline 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 4 & 0 & 1 & 0 \\ 5 & 0 & 1 & 0 \\ 6 & 0 & 1 & 0 \\ 7 & 0 & 0 & 1 \\ 8 & 0 & 0 & 1 \\ 9 & 0 & 0 & 1 \end{array} \qquad U = \begin{array}{c|cc} & b & r \\ \hline 0 & 1 & 0 \\ 1 & 1 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \\ 4 & 0 & 1 \\ 5 & 0 & 1 \\ 6 & 0 & 1 \\ 7 & 1 & 0 \\ 8 & 1 & 0 \\ 9 & 1 & 0 \end{array}$$

$$\Rightarrow N = U^TV = \begin{bmatrix} 3 & 0 & 3 \\ 1 & 3 & 0 \end{bmatrix} \quad , \ \mathbf{1}\varphi(N)\mathbf{1}^T = 9$$

$$N\mathbf{1}^T = [6, 4] \ , \ \mathbf{1}\varphi(N\mathbf{1}^T) = 21$$

$$\mathbf{1}N = [4, 3, 3] \ , \ \varphi(\mathbf{1}N)\mathbf{1}^T = 12$$

$$\Rightarrow \mathcal{D} = 0.667, \ \mathcal{AD} = 0.312$$

$$\begin{array}{ccccc} VV^T & - & UU^T & = & \Delta \\ \begin{bmatrix} 1&1&1&1&0&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&0&0&1&1&1 \end{bmatrix} & - & \begin{bmatrix} 1&1&1&0&0&0&0&1&1&1 \\ 1&1&1&0&0&0&0&1&1&1 \\ 1&1&1&0&0&0&0&1&1&1 \\ 0&0&0&1&1&1&1&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0 \\ 1&1&1&0&0&0&0&1&1&1 \\ 1&1&1&0&0&0&0&1&1&1 \\ 1&1&1&0&0&0&0&1&1&1 \end{bmatrix} & = & \begin{bmatrix} 0&0&0&1&0&0&0&1&1&1 \\ 0&0&0&1&0&0&0&1&1&1 \\ 0&0&0&1&0&0&0&1&1&1 \\ 1&1&1&0&1&1&1&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 1&1&1&0&0&0&0&0&0&0 \\ 1&1&1&0&0&0&0&0&0&0 \\ 1&1&1&0&0&0&0&0&0&0 \end{bmatrix} \end{array}$$

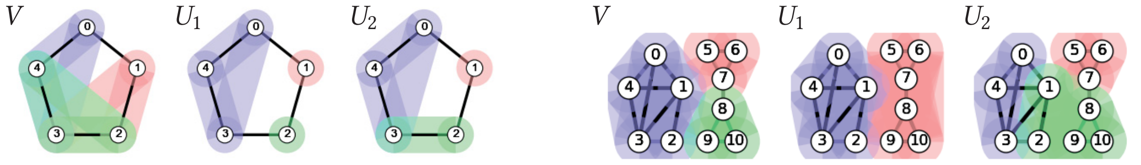
$$\Rightarrow \|\Delta\|_F^2 = 30$$

$$NF_{RI} = 90 \Rightarrow \mathcal{D} = 0.667$$

$$\|VV^{T'}\|_F^2 = |VV^{T'}| = 24 \ , \ \|UU^{T'}\|_F^2 = |UU^{T'}| = 42$$

$$NF_{ARI} = 43.5 \Rightarrow \mathcal{AD} = 0.312$$

***Figure 3.5:*** *Example for contingency v.s. co-membership based formulation. The (A)RI is first derived from the contingency table N, using* $\mathcal{D}$ *formula where* $\varphi(x) = x(x-1)/2$*. Then the same results are derived from the comparison of co-membership matrices* $UU^T$ *and* $VV^T$*, using the alternative formulation of* $\mathcal{D}$*, where* $A'_{n\times n} = A - \mathbf{I}_n$.



| | $\omega$ | $A\omega$ | $RI_\delta$ | $ARI_\delta$ | $RI'_\delta$ | $ARI'_\delta$ |
|---|---|---|---|---|---|---|
| $(V, U_1)$ | 0.5 | 0.22 | 0.80 | 0.25 | 0.90 | 0.32 |
| $(V, U_2)$ | 0.5 | 0.19 | 0.88 | 0.49 | 0.94 | 0.58 |

***(a)*** *Revisit to Figure 3.3a. Reported in the table are values for Omega index ($\omega$), and its adjusted version ($A\omega$), followed by our exact and approximate (marked by$'$) $\delta$-based $(A)RI$, derived from the clustering co-memberships distance $\Delta$.*

| | $NMI$ | $NMI'$ | $NMI''$ | $RI_\delta$ | $ARI_\delta$ | $RI'_\delta$ | $ARI'_\delta$ |
|---|---|---|---|---|---|---|---|
| $(V, U_1)$ | 0.78 | 0.61 | 0.53 | 0.84 | 0.66 | 0.85 | 0.70 |
| $(V, U_2)$ | 0.71 | 0.62 | 0.61 | 0.78 | 0.47 | 0.80 | 0.57 |

***(b)*** *Revisit to Figure 3.3b. Here our exact and approximate co-membership based formulations are in agreement with the original non-overlapping $NMI$, and give a higher similarity score to $U_1$. Whereas the two overlapping $NMI$ extensions state the opposite.*

***Figure 3.6:*** *Revisit to the examples of Figure 3.3. On the left we see that Omega index ($\omega$) is unable to differentiate between $U_1$ and $U_2$, whereas its adjusted version even gives higher score to $U_1$, which is the opposite of what we expect. The fact that $U_2$ is more similar to $V$ is captured by our $\delta$-based $(A)RI$. On the right we see an example of disagreement between the original $NMI$ and its two set-matching based extensions for overlapping cases. Here since the problem is disjoint, $(A)RI_\delta$ gives same results as the original $(A)RI$.*

Figure 3.6 shows the other two test case examples from Section 3.3.2, and compares the results from $RI_\delta$ and $ARI_\delta$ with the other alternative overlapping measures, *i.e.*, the *Omega* index and the two overlapping versions of $NMI$; where unlike these alternatives, the new $\delta$-based measures rank the agreements correctly.

It is worth mentioning that, the Omega Index($\omega$) [30] can also be derived from comparing the co-membership matrices. In more detail, if we define $\Omega = [UU^T == VV^T]$, *i.e.*, $\Omega_{ij} = 1$ if $(UU^T)_{ij} == (VV^T)_{ij}$ and zero otherwise; and assuming $f_A(i)$ denotes the frequency of value $i$ in matrix $A$, then $\omega$ and its adjusted version ($A\omega$) can be calculated as:

$$\omega = |\Omega| - tr(\Omega), \quad A\omega = \frac{\omega - E[\omega]}{1 - E[\omega]}, \quad where \quad E[\omega] = \sum_{i=0}^{min(r,k)} f_{UU^T}(i) f_{VV^T}(i) \tag{3.17}$$

Similarly, we can compare the co-membership matrices of $UU^T$ and $VV^T$ in other way, *e.g.*, using matrix divergences [40, 74]; or considering other normalized forms of $\Delta$. In our experiments, we examine these two variations:

$$\mathcal{D}_{norm} = \frac{\|UU^T - VV^T\|_F^2}{\|UU^T\|_F^2 + \|VV^T\|_F^2} \tag{3.18}$$

$$I_{\sqrt{tr}} = \frac{tr(UU^T VV^T)}{\sqrt{tr((UU^T)^2) tr((VV^T)^2)}} = \frac{|UU^T \circ VV^T|}{\|UU^T\|_F^2 \|VV^T\|_F^2} \tag{3.19}$$

It is also worth pointing out that in some applications, such as ensemble or multi-view clustering, we may not need the normalization and a measure of distance may suffice. In which case we can directly work with $\Delta$ in Definition 3.4.1.

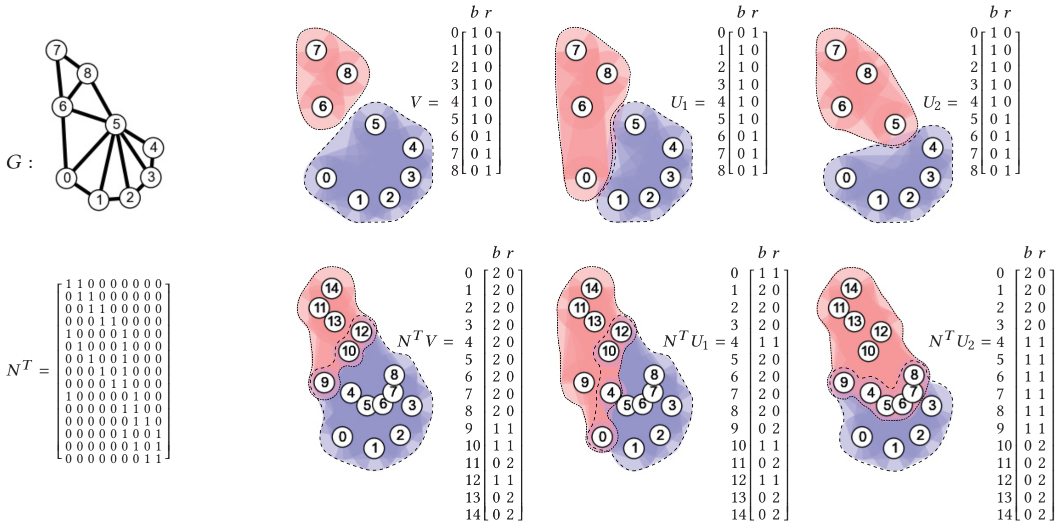


***Figure 3.7:*** *Revisiting the example of Figure 3.2. Top) In the original data and considering only nodes, $U_1$ and $U_2$ have the same agreement with $V$, since both $U_1$ and $U_2$ have one node clustered differently than $V$. Bottom) Transformed data structure using the corresponding clusterings clearly identifies that $U_1$ is closer to $V$ when compared to $U_2$,* i.e., *the difference between $U_1$ and $V$ is less. Note that the transformed data is similar to the line graph (edges as nodes) of the original data.*

We conclude this section by presenting the extension of these algebric reformulations for *network clustering*. Let $N$ denote the structure of the graph $G$ as an incidence matrix, *i.e.*, $N_{ik} = \sqrt{A_{ij}}$ if node $i$ is incident with edge $k = (i, j)$, and zero otherwise. Assuming a clustering as a transformation which assigns each data-point to one of its $k$ clusters, *i.e.*, $U : n \mapsto k$ , we can incorporate the structure by measuring the distance between the transformed data by $U$ and $V$ as:

$$\mathcal{D}_{\perp}(U, V|G) = \mathcal{D}(N^T U, N^T V) \tag{3.20}$$

Figure 3.7 provides an intuitive example for this transformation. This transformation generates overlaps, hence only the new reformulations, which are valid for overlapping clusters, are applicable as $\mathcal{D}$; *e.g.*, the $ARI_\delta$. One should also note that, this is very similar to counting the edges using overlap function $\xi$ introduced earlier in Section 3.3.1. Alternatively, we can assume each edge as a cluster of two nodes, and measure the distance of a clustering from the underlying structure of the graph. Consequently, the structure dependent distance of $U$ and $V$ can be defined as a combination of $\mathcal{D}(U, N)$, $\mathcal{D}(V, N)$ and $\mathcal{D}(U, V)$, for example:

$$\mathcal{D}_{+}(U, V|G) = \alpha\mathcal{D}(U, V) + (1 - \alpha)|\mathcal{D}(U, N) - \mathcal{D}(V, N)|, \quad \alpha = 0.5 \tag{3.21}$$

Table 3.1, Table 3.2 and Table 3.3 compare the structure dependent and independent measures for our earlier test case examples in Figure 3.2, Figure 3.3a, and Figure 3.3b, respectively. The first two rows of these tables show the structure independent measures, the next four rows are the

| $I:$ | $RI_\delta$ | $ARI_\delta$ | $RI'_\delta$ | $ARI'_\delta$ | $I_{norm}$ | $I_{\sqrt{tr}}$ |
|---|---|---|---|---|---|---|
| $(V,U_1)$ | 0.778 | 0.556 | 0.802 | 0.604 | 0.695 | 0.815 |
| $(V,U_2)$ | 0.778 | 0.556 | 0.802 | 0.604 | 0.695 | 0.815 |
| $C_\perp(V,U_1\|G)$ | 0.926 | 0.744 | 0.928 | 0.752 | 0.799 | 0.923 |
| $C_\perp(V,U_2\|G)$ | 0.857 | 0.417 | 0.859 | 0.435 | 0.708 | 0.844 |
| $C_+(V,U_1\|G)$ | 0.889 | 0.773 | 0.901 | 0.797 | 0.843 | 0.904 |
| $C_+(V,U_2\|G)$ | 0.833 | 0.660 | 0.900 | 0.776 | 0.832 | 0.885 |
| $(N,V)$ | 0.750 | 0.500 | 0.979 | 0.327 | 0.512 | 0.662 |
| $(N,U_1)$ | 0.750 | 0.491 | 0.979 | 0.337 | 0.503 | 0.668 |
| $(N,U_2)$ | 0.639 | 0.264 | 0.977 | 0.275 | 0.481 | 0.616 |

***Table 3.1:*** *Results of different agreement measures for the test case of Figure 3.2 (and Figure 3.7). For example looking at $ARI'_\delta$, from the structure independent version we have $ARI'_\delta(V,U_1) = ARI'_\delta(V,U_2) = 0.604$; whereas when considering the structure, both $C_\perp ARI'_\delta$ and $C_+ ARI'_\delta$ rank $U_1$ in higher agreement with $V$ compared to $U_2$,* i.e., *$C_\perp ARI'_\delta(V,U_1|G) = 0.752 > C_\perp ARI'_\delta(V,U_2|G) = 0.435$ and $C_+ ARI'_\delta(V,U_1|G) = 0.797 > C_+ ARI'_\delta(V,U_2|G) = 0.776$.*

structure based versions, and the last three rows show the agreement of each clustering directly with the structure. In particular in Table 3.1, we see that unlike the original structure independent measures which result in the same agreement for $U_1$ and $U_2$, all the structure based extensions correctly give higher agreement score to $U_1$ compared to $U_2$. We can also see that $U_1$, when compared to $U_2$, has in fact more agreement with the structure of the underlying graph. Table 3.2 and Table 3.3 extend the results presented in Figure 3.6 for the two overlapping test case examples.

| $I:$ | $RI_\delta$ | $ARI_\delta$ | $RI'_\delta$ | $ARI'_\delta$ | $I_{norm}$ | $I_{\sqrt{tr}}$ |
|---|---|---|---|---|---|---|
| $(V,U_1)$ | 0.800 | 0.245 | 0.902 | 0.318 | 0.532 | 0.764 |
| $(V,U_2)$ | 0.875 | 0.490 | 0.942 | 0.577 | 0.663 | 0.894 |
| $C_\perp(V,U_1\|G)$ | 0.856 | 0.186 | 0.868 | 0.211 | 0.536 | 0.860 |
| $C_\perp(V,U_2\|G)$ | 0.913 | 0.427 | 0.924 | 0.483 | 0.672 | 0.961 |
| $C_+(V,U_1\|G)$ | 0.775 | 0.556 | 0.919 | 0.617 | 0.720 | 0.859 |
| $C_+(V,U_2\|G)$ | 0.863 | 0.712 | 0.954 | 0.765 | 0.824 | 0.945 |
| $(N,V)$ | 0.850 | 0.333 | 0.933 | 0.528 | 0.682 | 0.816 |
| $(N,U_1)$ | 0.600 | 0.200 | 0.870 | 0.444 | 0.590 | 0.771 |
| $(N,U_2)$ | 0.700 | 0.400 | 0.900 | 0.576 | 0.666 | 0.822 |

***Table 3.2:*** *Results of different agreements for the omega example of Figure 3.3a.*

| $I:$ | $RI_\delta$ | $ARI_\delta$ | $RI'_\delta$ | $ARI'_\delta$ | $I_{norm}$ | $I_{\sqrt{tr}}$ |
|---|---|---|---|---|---|---|
| $(V,U_1)$ | 0.836 | 0.660 | 0.851 | 0.703 | 0.705 | 0.840 |
| $(V,U_2)$ | 0.782 | 0.471 | 0.802 | 0.567 | 0.626 | 0.721 |
| $C_\perp(V,U_1\|G)$ | 0.900 | 0.790 | 0.906 | 0.806 | 0.768 | 0.902 |
| $C_\perp(V,U_2\|G)$ | 0.857 | 0.564 | 0.862 | 0.607 | 0.667 | 0.798 |
| $C_+(V,U_1\|G)$ | 0.855 | 0.708 | 0.922 | 0.793 | 0.839 | 0.866 |
| $C_+(V,U_2\|G)$ | 0.818 | 0.556 | 0.897 | 0.716 | 0.782 | 0.804 |
| $(N,V)$ | 0.945 | 0.865 | 0.977 | 0.620 | 0.615 | 0.814 |
| $(N,U_1)$ | 0.818 | 0.621 | 0.970 | 0.502 | 0.589 | 0.707 |
| $(N,U_2)$ | 0.800 | 0.506 | 0.968 | 0.485 | 0.552 | 0.702 |

***Table 3.3:*** *Results of different agreements for the matching example of Figure 3.3b.*

Here we see that the results of the structure dependent measures are consistent with the structure independent measures; however, the structure dependent agreements become stronger than the independent versions in Table 3.3, while getting weaker in Table 3.2. This is due to the fact that the clusterings in Figure 3.6b better correspond with the structure of the underling graph, when compared to the clusterings of Figure 3.6a.

To summarize, here we presented an algebraic reformulation for the *ARI*, denoted by $ARI_\delta$, based on the difference between the co-membership matrices of the two clusterings. $ARI_\delta$ is identical to the original *ARI* measure if clusters are disjoint; while unlike the original measure, it naturally extends to overlapping cases. We showed that this overlapping extension does not have the shortcomings of other current alternatives, mainly the overlapping extensions of *NMI* and the Omega index. However, one should note that the formulation of $ARI_\delta$ requires matrix representations, hence it is harder to implement and computationally more expensive; particularly for its structure dependent variation in Equation 3.20, which also requires matrix multiplications. Fortunately, we can derive $ARI_\delta$ from an alternative formula which is not based on a matrix representation of the data, and hence is more efficient and practical. We present this new formulation in the next section. The new formulation is more generalized, similar to the generalization presented earlier in Section 3.3. However unlike the previous generalization which does not extend to overlapping cases, as discussed in Section 3.3.2, the new generalization accurately measures agreement of overlapping clusters; hence it can be used to construct new overlapping agreement indexes. For instance, in the next section we also introduce a novel overlapping extension of *NMI* derived from this generalization which, is not directly available from Δ. This extension reduces to the original *NMI* if clusterings are disjoint, which is unlike the other common extensions [78, 95].

## 3.5 Clustering Agreement Index (CAI) for Overlapping Clusters

Consider clustering $U$ which clusters dataset $D$ with $n$ datapoints. For each data point $i \in D$, and cluster $u \in U$, let $u_{\leftarrow i}$ denote the membership strength of data-point $i$ in cluster $u$. The restrictive assumption on this definition is $\sum_u u_{\leftarrow i} = 1, \ \forall \ i \in D$, for disjoint and fuzzy cases, whereas in the former we also have $u_{\leftarrow i} \in \{0,1\}$, and in the latter $u_{\leftarrow i} \in [0,1]$. Here, we do not put any assumption on the form of the clusterings, $u_{\leftarrow i}$, and define a general agreement measure for two given clusterings of a same data-set. More formally, we assume $i \in u$ iff $u_{\leftarrow i} > 0$ and obtain the size of each cluster $u \in U$ as:

$$o_u = \sum_{i \in u} u_{\leftarrow i} \tag{3.22}$$

Similarly, we consider pairwise overlap between each pair of clusters $u$ and $v$, where $i \in u \cap v$ iff $u_{\leftarrow i} > 0 \wedge v_{\leftarrow i} > 0$, and compute the size of this overlap as:

$$o_{uv} = \sum_{i \in u \cap v} u_{\leftarrow i} \times v_{\leftarrow i} \tag{3.23}$$

If we consider two clusterings $U$ and $V$ of the (same) dataset $D$, then $\{\{o_{uv}\ \forall u \in U\}\ \forall v \in V\}$ constitutes the confusion matrix of $U$ and $V$. In the same manner, we can represent the intrinsic overlaps of the clustering themselves as $\{\{o_{uu'}\ \forall u \in U\}\ \forall u' \in U\}$ and $\{\{o_{vv'}\ \forall v \in V\}\ \forall v' \in V\}$. We quantify and sum up these overlaps using a generic function, $\varphi : \mathbb{R}^{\geq 0} \mapsto \mathbb{R}$; and also consider an expectation form, $\mathcal{E}$, to make the parallel with the previous formulations. More formally:

$$\begin{aligned} O_{UU} &= \sum_{u \in U} \sum_{u' \in U} \varphi(o_{uu'}) \ , \quad O_{VV} = \sum_{v \in V} \sum_{v' \in V} \varphi(o_{vv'}) \\ O_{UV} &= \sum_{u \in U} \sum_{v \in V} \varphi(o_{uv}) \ , \quad \mathcal{E}_{UV} = \sum_{u \in U} \sum_{v \in V} \varphi(\frac{o_u o_v}{n}) \end{aligned}$$

Based on these we define the Clustering Agreement Index (CAI) as:

$$CAI(U,V) = \frac{O_{UV} - \mathcal{E}_{UV}}{\frac{1}{2}(O_{UU} + O_{VV}) - \mathcal{E}_{UV}} \tag{3.24}$$

We can derive different agreement measures using different $\varphi(.)$ functions. In particular, we introduce two agreement indexes derived with $\varphi(x) = x^2$ and $\varphi(x) = x \log(x)$; which are respectively called *CRI* and *CMI*. More specifically, *CRI* and *CMI* are derived/defined as:

$$CRI(U,V) = \frac{\sum\limits_{u \in U} \sum\limits_{v \in V} o_{uv}^2 - \sum\limits_{u \in U} \sum\limits_{v \in V} (\frac{o_u o_v}{n})^2}{\frac{1}{2}\left[\sum\limits_{u \in U} \sum\limits_{u' \in U} o_{uu'}^2 + \sum\limits_{v \in V} \sum\limits_{v' \in V} o_{vv'}^2\right] - \sum\limits_{u \in U} \sum\limits_{v \in V} (\frac{o_u o_v}{n})^2} \tag{3.25}$$

$$CMI(U,V) = \frac{\sum\limits_{u \in U} \sum\limits_{v \in V} o_{uv} \log(o_{uv}) - \sum\limits_{u \in U} \sum\limits_{v \in V} \frac{o_u o_v}{n} \log(\frac{o_u o_v}{n})}{\frac{1}{2}[\sum\limits_{u,u' \in U} o_{uu'} \log(o_{uu'}) + \sum\limits_{v,v' \in V} o_{vv'} \log(o_{vv'})] - \sum\limits_{u \in U} \sum\limits_{v \in V} \frac{o_u o_v}{n} \log(\frac{o_u o_v}{n})} \tag{3.26}$$

The *CRI* and *CMI* indexes reduce respectively to the original *ARI* and *NMI* indexes when clusterings are disjoint, however, they are also applicable to the overlapping clusters, *i.e.*,

**Identity 3.5.1.** *CRI of Equation 3.25 reduces to the ARI of Equation 3.6 for disjoint clusters; refer to Appendix B.1.7 for the proof.*

**Identity 3.5.2.** *CMI of Equation 3.26 reduces to the $NMI_{sum}$ of Equation 3.9 for disjoint clusters; refer to Appendix B.1.8 for the proof.*

Moreover, *CRI* provides an alternative formulation for the $\Delta$-based formulation of *ARI*, for when clusterings are disjoint or overlapping , *i.e.*,

**Identity 3.5.3.** *CRI of Equation 3.25 is equivalent to the $ARI_\delta$ of Equation 3.16; refer to Appendix B.1.9 for the proof.*

Although equivalent, the *CRI* formulation is more efficient than the $ARI_\delta$, and is in fact on par with the disjoint variation in terms of the efficiency. In more detail, let $k$ and $r$, denote the number of clusters in the two given clusterings, where $k \geq r$. Also let $m$ denote the cardinality of the largest cluster in the two clusterings. The complexity of *ARI* (Equation 3.6) and *NMI* (Equation 3.9), which are only defined for disjoint clusters, is $O(krm)$. Since they are both formulated based on the $k \times r$ confusion matrix, whose entries are the pairwise cluster overlaps measured by set intersection, *i.e.*, $O(m)$. The CAI (Equation 3.24), is also formulated based on $k \times r$ pairwise cluster overlaps, which are defined in Equation 3.23, and can be computed in $O(m)$. Moreover, CAI also measures the pairwise cluster overlaps in each of the two clusterings themselves, which costs $O(k^2m)$ and $O(r^2m)$. Hence the total complexity of CAI formula, is $O((kr + k^2 + r^2)m)$, which is in the order of $O(k^2m)$. This is about the same as the cost for computing agreement of disjoint clusters; and is much more efficient compared to the Δ-based formulation of ARI (Equation 3.16), which is in the order of $O(n^3)$, where $n$ denotes the number of data points. Therefore CAI is more favourable in real world applications where $n$ is large.

On the other hand, when compared to the previous generalization in Section 3.3 ($\mathcal{D}$), CAI distinguishes the overlaps within the data itself, and factors them from the confusion matrix; hence measures the agreement of the overlapping clusters correctly. For instance, following the discussion in Section 3.3.2, unlike the indexes derived from $\mathcal{D}$ in which the agreement becomes a function of the overlap within the data even if clusters are identical, CAI always returns 1 for identical clusterings, *i.e.*, $CAI(U,U) = 1$, since we have,

$$CAI(U,U) = \frac{\mathcal{O}_{UU} - \mathcal{E}_{UU}}{\frac{1}{2}(\mathcal{O}_{UU} + \mathcal{O}_{UU}) - \mathcal{E}_{UU}} = 1$$

We can further show that *CAI* is symmetric, *i.e.*, $CAI(U,V) = CAI(V,U)$; since $o_{uv} = o_{vu}$, from which simply follows that $\mathcal{O}_{UV} = \mathcal{O}_{VU}$ and $\mathcal{E}_{UV} = \mathcal{E}_{VU}$.

One of the advantages of CAI is that it does not enforce any assumptions on the form of memberships of data-points in clusters, which makes it flexible. This flexibility is in particular important when adopting the clustering agreement indexes to compare communities, *i.e.*, clusters of nodes/vertexes on networks. Hence we can use the intuitive ways discussed in Section 3.3.1 to incorporate the structure of the data in *CAI* when measuring the agreement between clusterings of networks; *i.e.*, to simply modify the memberships of nodes in clusters (*e.g.*, weighted by degree) and/or the overlap of two clusters (*e.g.*, sum of edges instead on nodes), to also consider the relationships between data points. One should note that these modifications are only possible with a flexible measure such as *CAI* which has no restricting assumptions on the marginals of the contingency table, or the form of data-points' memberships in clusters.

## 3.6 Experimental Comparison of Agreement Indexes

Here, we examine the clustering agreement measures, introduced above, in the context of community mining evaluation; which is their most common application. More specifically, we select a set of common community mining methods, which discover clusters in a given network based on different methodologies. We then rank their performance according to different clustering agreement measures; which compare the results of these methods with the ground-truth clustering. However, the purpose here is not to compare the general performance of community mining methods, but rather to show different comparisons/rankings we obtain using different agreement measures.

In more detail, the selected methods are: Louvain [16], WalkTrap [120], PottsModel [138], FastModularity [104], and InfoMap [140] for disjoint clusterings of Section 3.6.1 and 3.6.2; and COPRA [55], MOSES [94], OSLOM [81], and BIGCLAM [167] for overlapping clusterings in Section 3.6.3. The authors' original implementations are used for the methods, with no parameter tuning (defaults are used); and the reported results are averaged over ten runs.

The datasets are generated using LFR [79] benchmarks, which are commonly used in the evaluation and comparison of community mining algorithms. LFR benchmarks generate networks with built-in (disjoint or overlapping) community structure, with controlled degree of difficulty; in particular, how mixed/well-separated are the communities, and the fraction of nodes which are overlapping. Here, the results are reported for the basic LFR parameters, chosen similar to the experiments by Lancichinetti and Fortunato [76], *i.e.*, networks with 1000 nodes, average degree of 20, max degree of 50, and power law degree exponent of -2; where the size of communities follows a power law distribution with exponent of -1, and ranges between 20 to 100 nodes. However, we observed similar patterns from other parameter settings. We describe these benchmarks in details in Chapter 4, where we also introduce an alternative generator called FARZ. The goal of the following experiments is to show how the choice of clustering agreement measure affects the common evaluation practice in the community detection literature, therefore we stick to the LFR benchmarks which are the current gold-standard. We leave the comparison of different benchmark generators to Chapter 4. .

### 3.6.1 Classic Measures

Figure 3.8 shows the rankings of the selected algorithms with respect to six common agreement measures. In more detail, each subplot provides a comparison of the community detection methods according to the corresponding clustering agreement index. In each subplot, the average agreement of the algorithms' results with the ground-truth clustering, is plotted as a function of the hardness of the problem. *First*, we can see that the rankings are overall consistent, which is expected since these indices are measuring the agreement with similar principle, as shown with our generalizations. *Second*, from the plot for $NMI$ we can observe its bias in favor of the large number of clusters [156]; *i.e.*, for large mixing parameters, when PottsModel algorithms detects signifi-

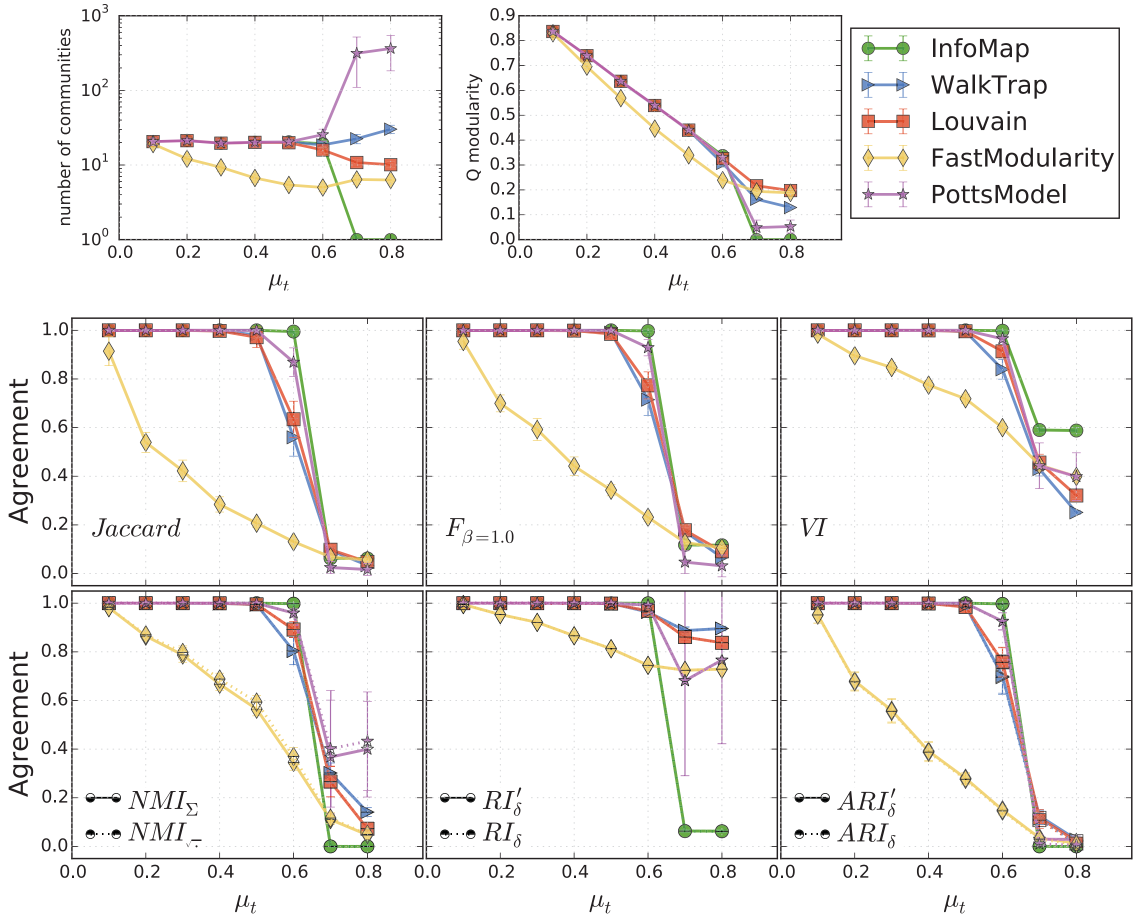


***Figure 3.8:*** *The agreement of results from different community detection algorithms with the ground-truth in unweighted LFR benchmarks, plotted as a function of the mixing parameter for topology $\mu_t$ (which determines the average fraction of edges that go outside the communities per each node,* i.e., *how well separated are the communities). The two first subplots report the number of communities found by each algorithm and the quality of the result according to the modularity $Q$ . In the last three plots, similar measures are overlaid to highlight the fact that they are highly similar,* e.g., *we can see that the plots for ARI and approximate ARI, ARI′, exactly overlap.*

cantly more communities, $NMI_{\sum}$ and $NMI_{\sqrt{\cdot}}$ rank the PottsModel significantly higher; which is not true according to all the other measures. The opposite bias is also observed in the plots of $VI$, where the algorithm that finds significantly less communities is ranked significantly higher, *i.e.*, Infomap. Based on these observation, we advise against using these two measures, particularly if the number of discovered clusters/communities might be very different from the ground-truth. *Third*, from the plot for *ARI* we can see that there is no clear difference between the rankings obtained by *ARI* of Equation 3.5 and Equation 3.6, which are plotted as $ARI_{\delta}$ and $ARI'_{\delta}$ respectively[8]. The

[8] The $\delta$ subscript indicates that the *ARI* is computed based on our $\delta$-based formulation, which is equivalent to the original *ARI* in this experiment, since communities are covering all nodes and non-overlapping (Identity 3.4.2).

latter is less commonly used, whilst its extended form presented in Section 3.4 is more appropriate for overlapping cases; hence *ARI′* is in general more favourable.

### 3.6.2 Structure Dependent Measures

Figure 3.9 compares the selected methods over *weighted* LFR benchmarks. These methods all result in clusterings which correspond well with the underlying structure, therefore the effect of ignoring the structure in comparing the clusterings is less apparent[9].

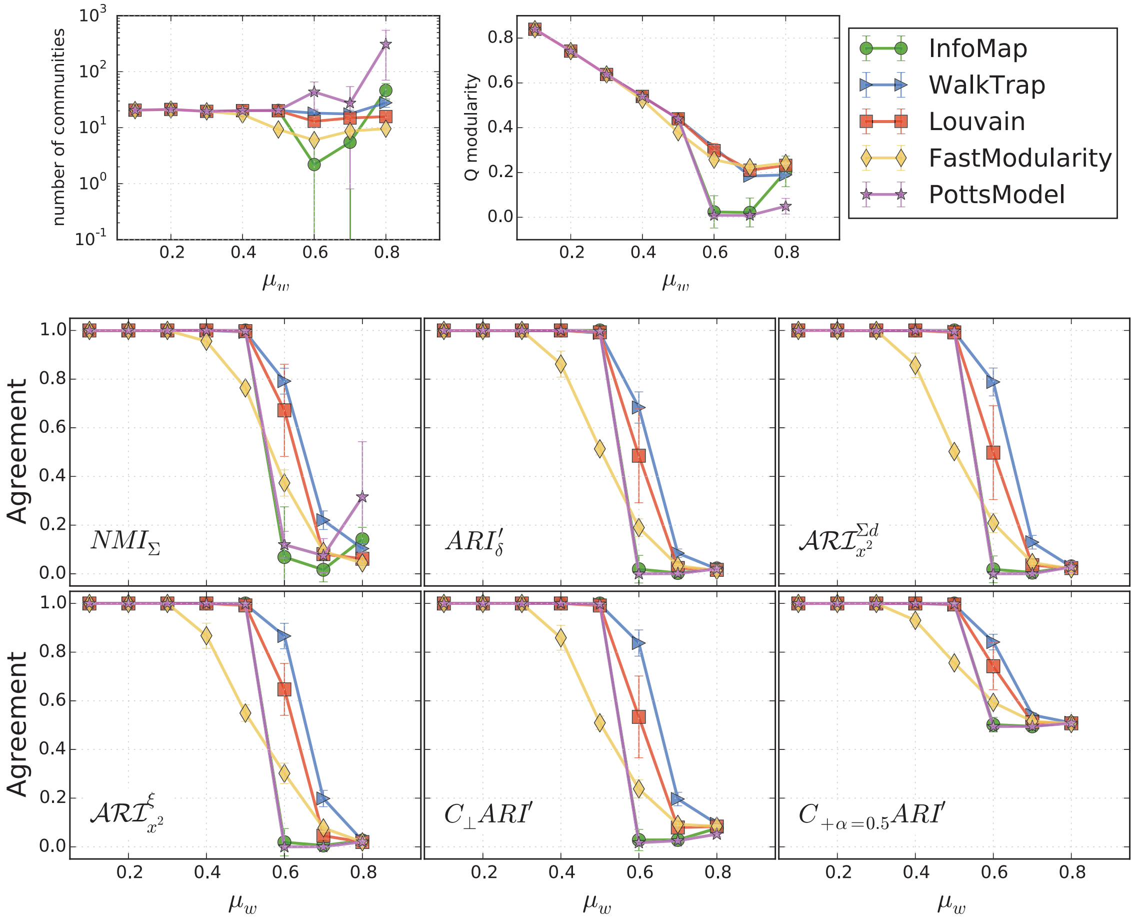


***Figure 3.9:*** *Comparison of the agreement indexes on weighted LFR benchmark, when the mixing parameter for weights varies and the mixing parameter for topology is fixed to* 0.5.

The rankings are overall consistent, however, we can still observe the difference between the structure dependent and independent agreement measures, which becomes clear only with the

[9] One could design experiments based on the test cases discussed in Section 3.4, which would result in significant difference between the structure dependent and independent measures. Here, however, our goal is to show the effect of different agreement indexes on the comparison of community detection algorithms, where a common practice is to use one of structure independent indexes.

presence of weights in this experiment. In more detail, we can see that the Walktrap method is performing better according to all the measures, however its superiority is more significant according to the structure dependent measures: *i.e.*, $\mathcal{ARI}^{\xi}_{x^2}$ , and $\mathcal{ARI}^{\Sigma d}_{x^2}$ introduced[10]in Section 3.3.1, and $C_{\perp}ARI'$ introduced in Section 3.4. On the other hand, $C_{+}ARI'$ does not decrease to zero similar to the other measures; since both the results and the ground-truth are with the same distance from the structure, *i.e.*, $|\mathcal{D}(U,N) - \mathcal{D}(V,N)| \simeq 0$, hence $\mathcal{D}_{+}(U,V|G) = \frac{1}{2}\mathcal{D}(U,V)$. Being less consistent with the other measures, the three former structure dependent forms are more favourable.

### 3.6.3 Overlapping Measures

Figure 3.10 shows the comparison of the selected methods based on the different overlapping agreement indexes, which are: the overlapping extensions of $NMI$, *i.e.*, $NMI'$ by Lancichinetti et al. [78] and $NMI''$ by McDaid et al. [95]; the omega index ($\omega$), and its adjusted version ($A\omega$); and our $\delta$-based formulations for the $RI$ and $ARI$, *i.e.*, $RI'_{\delta}$, and $ARI'_{\delta}$.

Obtained rankings are generally consistent, similar to the previous experiments. *First,* we can see that the unadjusted measures, *i.e.*, $\omega$ and $RI'_{\delta}$, can not properly differentiate between the different algorithms. *Secondly,* we observe the "problem of matching" with the overlapping extensions of $NMI$, described earlier in Section 3.3.2. In more detail, MOSES algorithm results in finer grained and thus more communities, which are more likely to not get matched/compared with the communities in the ground-truth, when applying a set-matching based agreement measure. Therefore it gets unfairly penalized, and is ranked significantly lower than the OSLOM. Their difference, however, is less significant according to both adjusted omega, $A\omega$, and our overlapping extension of $ARI$, $ARI'_{\delta}$; whereas the quality of the MOSES results are higher than OSLOM according to the modularity Q of Newman [104]. *Lastly,* in this setting the difference between $A\omega$ and $ARI'_{\delta}$ is not as clear as it could be, since each node can only belong to at most two communities; whereas the difference becomes clear if a node can belong to many communities, see Section 3.4.

Figure 3.11 compares the CRI (Equation 3.25) and CMI (Equation 3.26) derived from $CAI$ generalization (Equation 3.24) against other overlapping alternatives. Here, we also observe that rankings of the algorithms are consistent according to the different agreement measures. Which is expected, since these indexes are very similar or in case of $ARI_{\delta}$ and $CRI$, identical. We can however observe the differences between set matching based extensions of $NMI$, *i.e.*, the first two subplots in the top row, with the $NMI$ extension presented here, *i.e.*, $CMI$ in the bottom right subplot. For example we can see that according to both $CMI$ and $ARI$, the performances of OSLOM and MOSES algorithms are close, however the previous overlapping extensions of $NMI$, *i.e.*, $NMI'$ and $NMI''$, rank OSLOM significantly higher. This is probably because OSLOM finds relatively less communities, and hence its communities are better matched with the ones in the ground-

[10] $\mathcal{ARI}_{x^2}$ derives from $\mathcal{AD}$ of Definition 3.3.3 with $\varphi = x^2$, and is same as the $ARI'$ if $\eta = \cap$ (Identity 3.3.6); here, however, the structure dependent variations of $\eta$ are used, *i.e.*, $\eta = \xi$ (edge counting) and $\eta = \Sigma d$ (degree weighted).

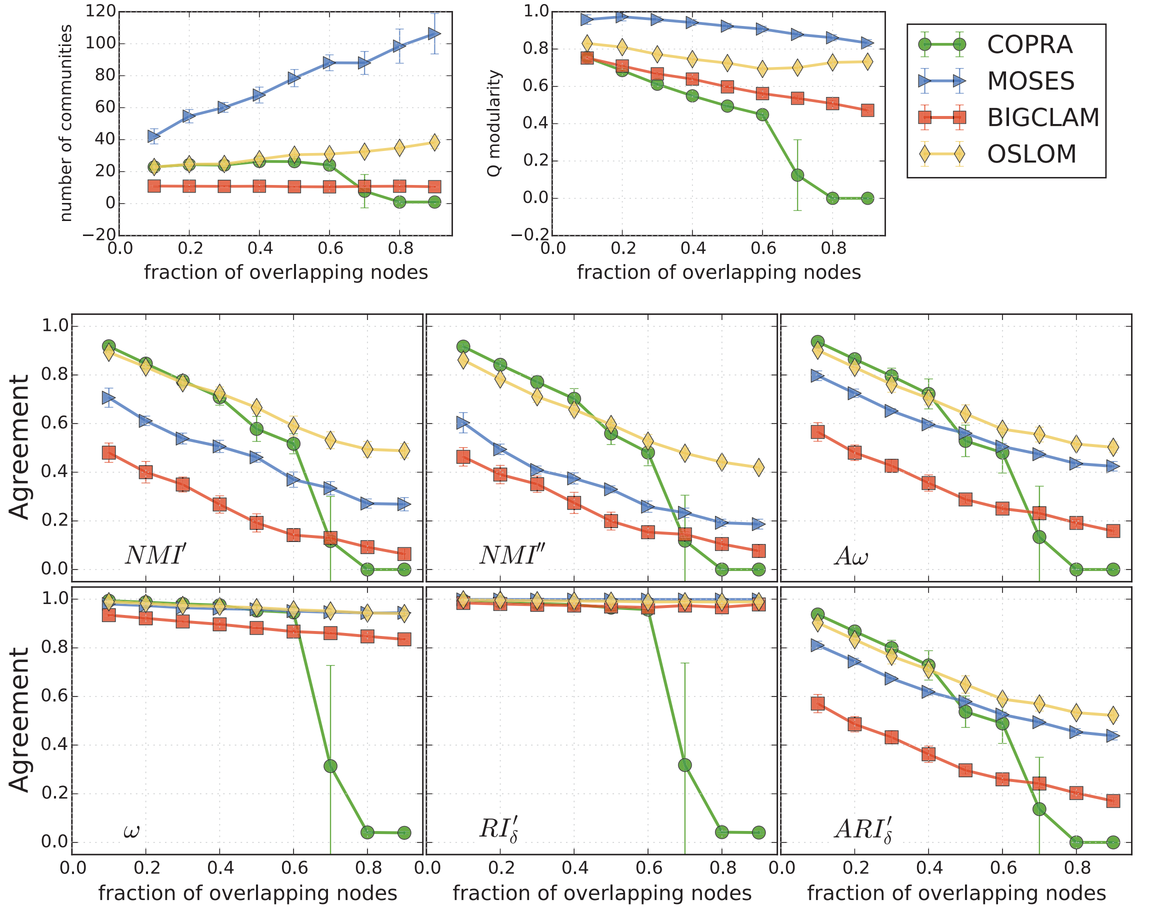


***Figure 3.10:*** *Comparison of agreement indexes on unweighted overlapping LFR benchmark, where the fraction of overlapping nodes varies, the mixing parameter for topology is fixed to* 0.1*, and the maximum number of communities a node can belong to is limited to* 2*; similar to experiments in [81].*

truth, whereas MOSES's communities are finer grained and not matched with any community in the ground-truth, when using a set-matching based agreement index. We can observe a similar pattern in the comparison of BIGCLAM and COPRA; *i.e.*, BIGCLAM finds too few communities and its performance plot is consequently shifted much lower by the $NMI'$ and $NMI''$, when compared to $CMI$ and other agreement measures. Therefore, in general CMI seems to be a more accurate overlapping extension for the $NMI$ when compared to the $NMI'$ and $NMI''$.

We further compare the time complexity of these indexes, reported in Figure 3.12, where the average run time for computing different indexes is plotted as a function of number of the nodes in the network. We can see that the proposed CMI and CRI overlapping derivations of CAI are much more efficient compared to the other measures, particularly when compared to the $ARI_\delta$.

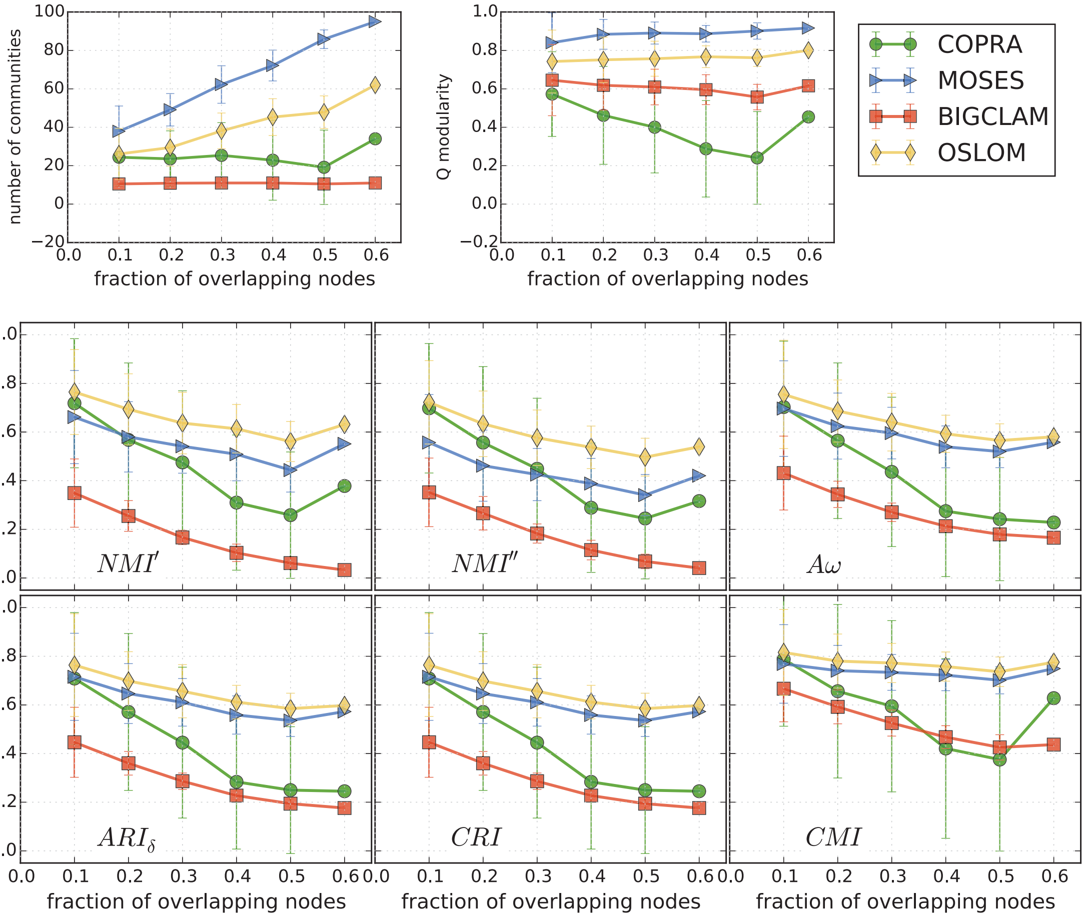


***Figure 3.11:*** *Comparison of overlapping agreement indexes with the CRI and CMI derivations obtained from CAI ; on unweighted overlapping LFR benchmark similar to the settings in Figure 3.10.*

## 3.7 Conclusions and Recommendations

In this chapter, we presented a generalized clustering distance, from which we can derive the two commonly used clustering agreement measures, *i.e.*, *NMI* and *ARI*. Not only this generalization sheds light on the relation between these two measures, but we also recommend using the derived formulae from this distance over the original formulations; since, *first*, they are identical when the original measures are defined; *second*, they require less assumptions on the clusterings and hence apply to more general cases, *e.g.*, when there are un-clustered data-points; *third*, they can be easily altered, for example to generate specific measures for clusters in networks. The latter example is in particular important, since all of the current agreement measures overlook the relationships between the datapoints, and hence are not appropriate for comparing clusters over networks, a.k.a. communities. Using our generalization, we introduced two extensions of the *ARI*, which incorporate the structure when comparing communities, *i.e.*, $\mathcal{ARI}^{\Sigma d}_{x^2}$ (degree weighted

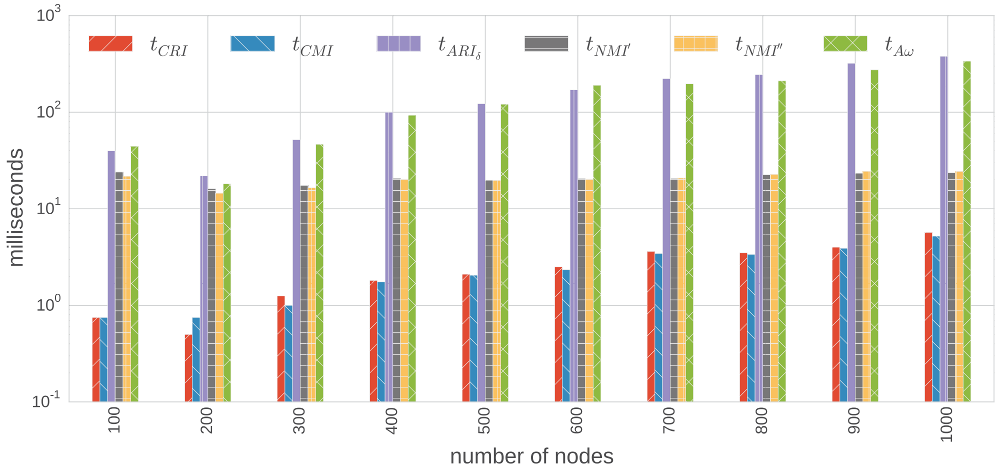


***Figure 3.12:*** *Experimental time comparison of CRI (Equation 3.25) and CMI (Equation 3.26) derived from our generalization (Equation 3.24), with $ARI_\delta$ (the $\Delta$-based formulations of Equation 3.16).*

overlap function) and $\mathcal{ARI}^{\xi}_{x^2}$ (edge counting overlap function). We recommend using these two extensions when comparing disjoint communities.

The generalized clustering distance, similar to other contingency based measure, does not readily extend to overlapping cases. Therefore, we presented an algebraic reformulation for the *ARI*, based on the difference of co-membership matrices of the two clusterings, denoted by $ARI'_\delta$. We recommend using $ARI'_\delta$, in particular when clusters are overlapping; since, *first*, it is identical to the original measure if clusters are disjoint; *second*, it naturally extends to overlapping cases; *third*, it is more valid compared to the current alternative overlapping measures, *i.e.*, it does not have the shortcomings of the overlapping extensions of *NMI* or the Omega index. However, one should note that this formulation requires matrix representations, hence is harder to implement and computationally more expensive. Even when using an efficient implementations using sparse matrices. To also incorporate the structure within this algebraic reformulation, we proposed $C_\perp ARI'$, and $C_+ARI'$. The former measures the distance between the transformed structure by each clustering; whereas the latter linearly combines the distances of each clustering to the structure, assuming the structure itself as another clustering, *i.e.*, when each edge is considered as a single cluster. We recommend using $C_\perp ARI'$ when comparing overlapping communities, since it is more consistent with the other measures. However, this transformation requires matrix multiplications and hence is computationally expensive; hence it is not as scalable as the other measures.

We further presented a generalization of the clustering agreement indexes, called *CAI* , which naturally extends to cases with overlapping clusters. The extensions derived from the *CAI* generalization are in particular important in the context of clustering networked data, a.k.a. community detection, where clusters are known to be highly overlapping. *CAI* is scalable, and therefore, unlike the previous measures, is applicable to the typical network datasets. We showed that the *ARI* derivation of the *CAI* formula, called *CRI*, is equivalent to the $ARI_\delta$. Whereas, the newly proposed formulation is more general and could be used to derive new agreement indexes, such as the novel overlapping extension of $NMI$, which is one of the most commonly used measure for disjoint clusters. Unlike previous overlapping extensions of $NMI$, the new derivation, called $CMI$, reduces to the original measure in case of disjoint clusterings. Therefore we recommend using the the derivation of measures for our newly proposed generalization in all cases of disjoint, fuzzy, or crisp overlapping clusters, and in particular to incorporate relations into the agreement measure when working with networked data.

# Chapter 4

# Modelling Modular Networks

The analysis presented in the previous chapters depends on the synthetic benchmark generators to create sample datasets with built-in ground-truth communities. In general, such network generator models are used extensively in the validation of community detection algorithms, where results of the algorithms are compared against the known structure in the synthetic networks. This chapter focuses on network models which synthesize networks with explicit modular structure. Here, we study the popular network generators, and disscuss how to improve upon the shortcomings of these models, in order to generate more lifelike networks. We further introduce a realistic and flexible benchmark generator geared toward validating and comparing community detection methods. The newly proposed model generates more truthful networks, *i.e.*, the characteristics of the synthesized networks and communities are more similar to what is observed in real world networks. Moreover, this model incorporates intuitive parameters, which have meaningful interpretation. Tuning these parameters provides means to generate a variety of realistic networks and presents different settings for comparing community detection algorithms. Parts of this chapter are published in [82], and the new model is submitted for publication.

## 4.1 Introduction

Community mining methods are often evaluated and compared based on their performance on benchmark datasets for which the true communities are known [38, 59, 76]. Since there exist only few and typically small real world networks with known community structure, these benchmarks are often synthesized using a network model, which has a built-in modular structure [51, 79]. This kind of evaluation is built upon the *assumption* that the performance of an algorithm on the benchmark datasets is a good predictor for its performance when applied to a real world network with unknown modular structure. For this assumption to hold, these benchmarks should be similar to the target real world networks, *i.e.*, comply with their observed characteristics. However, the current synthetic generators fail to exhibit some of the basic characteristics of real networks such as assortativity and transitivity [100, 143, 175]. Here, we review these network models focusing on the generators with an explicit modular structure, such as GN benchmarks by Girvan and Newman [51], and LFR benchmarks by Lancichinetti, Fortunato, and Radicchi [79].

It is also worth to mention common alternatives to the synthetic benchmarks, which are real-world networks with explicit or predefined nodal attributes that are considered as the ground-truth communities; examples are: user memberships in a social network, venues in a scholarly collaboration network, or product categories in an online co-purchasing network [70, 71, 146, 166]. In general, there exists an interplay between the attributes of nodes and the structure of the networks [33, 75], and in some contexts these attributes act as the primary organizing principle of the underlying communities [152]. However, this notion of ground-truth communities is weak [67], and these attributes should be considered correlated with the underlying community structure; we discuss this further in Chapter 5.

In this chapter, we attempt to provide better benchmarks for the community detection task, where we first examine the current generators, discuss their shortcomings and limitations, and propose alterations to improve them. Then, we present a simple alternative benchmark generator, called FARZ[1], which follows the evolution patterns and characteristics of real networks, and hence is more suitable for validation of community detection algorithms. In FARZ, communities are defined as the natural structure underlying the networks. This is unlike its common contender, LFR, where a community structure is overlaid on an existing network by imposing rewirings of multiple connections. Moreover, FARZ incorporates relevant intuitive parameters which could be used to generate a wide range of experimental settings, and hence enables a more thorough, domain dependent, comparison of community detection algorithms.

## 4.2 Overview of Network Models

There are many generative models proposed for real world networks [22, 52, 101, 109, 154]. Here, we survey the common models with an emphasis on those that incorporate a modular structure.

### 4.2.1 Classic Network Models

The classical ***Erdős and Rényi (ER)*** model [45] generates random graphs of a given size, where edges are formed independently and with uniform probability. These graphs comply with the small-world property observed in real world graphs, *i.e.*, they have a relatively small diameter. However, they have binomial degree distribution, which converges to a Poisson degree distribution for large number of nodes. This is unlike real world networks which are known to exhibit heavy tail degree distributions, a.k.a. scale-free characteristic. Another important disagreement of the real networks and the networks synthesized by the ER model is that the ER networks fail to exhibit transitivity [108]. Transitivity is measured by the average clustering coefficients of nodes,

[1] Following the same naming convention as the previous models, the letters are the first name of the authors involved in the development of the model; where the ordering is chosen so that the word has a meaning, *i.e.*, based on different transliterations, FARZ means sorting, division, or assess in Arabic; and assumption, or agile in Persian/Farsi.

which represents the fraction of connected neighbours per node. The ***Watts and Strogatz (WS)*** model [162] is another notable generator for small world graphs. This model starts with a regular graph (ring lattice), then rewires links with a probability $\beta$. The WS model is therefore able to generate graphs with high transitivity. The degree distribution of the generated networks, however, does not follow a power law, and therefore this model is also not scale-free.

The ***Barabási and Albert (BA)*** model [10] is one of the most basic models that generates random scale-free networks. Starting with an initial network, nodes are added at each step, while the newly added node forms $m$ connections with the existing nodes according to the preferential attachment a.k.a. accumulative advantage, Yule process, Matthew effect, or rich get richer; which states that the probability of forming a connection to an existing node is proportional to the current degree of that node. The networks generated with this model are analytically shown to have a power law degree distribution, small average path length (small-world), assortative mixing (of degrees), and transitivity higher than random graphs [6]. Although the generated BA graphs comply with macroscopic properties observed in real networks, the evolution of networks in this model is not realistic as discussed by Leskovec et al. [86].

Leskovec et al. [84, 85] propose the ***Forest Fire (FF)*** model which has similar desirable properties, while it is also designed to follow the evolution trends observed empirically in the real social networks *i.e.*, the networks become denser over time, with the average degree increasing, and the diameter decreasing. The FF model grows one node at a time, where every new node, first connects to an existing node called ambassador, chosen uniformly at random. Then, the new node recursively forms a random number of connections with the neighbours of every node it connects to −outlinks to specific number of inlink and outlink neighbours, drawn from geometric distributions with means of $p/(1-p)$ and $rp/(1-rp)$ respectively, where $p/rp$ is called forward/backward burning probability. The rich get richer effect comes naturally as the nodes with more connection are more likely to get linked to the new node, which provides the heavy-tailed degree distribution. The densification power law comes from the fact that the newly entered node would probably have more links to neighbours close to its ambassador. And different parameter values of the model could generates sparse/dense graphs with shrinking or increasing diameter. The FF model is introduced as an improvement over another model proposed in the same article, *i.e.*, ***Community Guided Attachment (CGA)***, which generates networks starting from a backbone tree that represents the hierarchical community structure. The CGA model can not exhibit the shrinking diameter and needs an explicit community structure to begin with. The latter, although being listed as a shortcoming in the original article, is a necessity for community detection benchmarks.

Other notable synthetic generators are the mathematical tractable models, such as the Stochastic ***Kronecker Graph*** model [87], and its generalization, the ***Multifractal Network*** model [12, 118]. These models generate networks with realistic properties, *i.e.*, , heavy-tailed degree distributions and high clustering coefficient, that can be mathematically proved. The recursive generation process is based on a set of generating parameters, *i.e.*, , hierarchical categories assigned to nodes that determine their probability of forming an edge. These parameters can be further fitted to a given real network instance.

### 4.2.2 Network Models with Explicit Modular Structure

***Girvan and Newman (GN)*** model [51] is the first model presented to generate synthetic networks with planted modular structure, to be used as community detection benchmarks. It is built upon the classic *Erdős and Rényi (ER)* model [45], and incorporates a modular structure by considering different probabilities for edges formed within and between modules. More precisely, nodes in the same community link with probability of $p_{in}$, and nodes from different communities link with probability of $1 - p_{in}$. However, these networks, since generated with the ER model, are not scale-free. Moreover, the GN model creates networks which are divided into groups of equal sizes (usually 128 nodes divide into four groups); whereas the sizes of communities in real networks do not have any reason to be equal in size [108]. In fact the sizes of communities in many real world networks are shown to follow a power law distribution [29].

The ***Lancichinetti, Fortunato, and Radicchi (LFR)*** model [79] amends the GN model by considering power law distributions for the degrees of nodes and community sizes. In more detail, it first samples the degree sequence and community sizes from power law distributions. Then, it randomly assigns each node (sampled degree) to a community, and links the nodes to create a network. Finally, it rewires the links such that for each node, a fraction ($\mu$) of its links go outside its community, while the rest ($1 - \mu$) are inside its community. The LFR benchmark is built upon the ***Configuration*** (CF) model [103]; which generates random graphs from a given degree sequence, by fixing the degree of each node, and connecting the available edge stubs uniformly at random. The networks generated with CF model are known to exhibit low transitivity. Hence, LFR applies also a post-processing rewiring step to increase the transitivity. This could have been addressed by using a more realistic starting model instead, as also suggested in [114]. However, LFR applies an extensive rewiring process on the initial network to overlay the communities, which changes the network structure chaotically. Therefore, even starting with a realistic network model, the properties are not guaranteed to be preserved, and in fact in most cases they are not. LFR is lately extended for hierarchical and overlapping communities [77], where the generation process is modified so that it generates the within and between links separately, instead of realizing the whole network at once. In more detail, after sampling the degree sequence $\mathcal{D}$, the within and between degree sequences are derived as $(1 - \mu)\mathcal{D}$ and $\mu\mathcal{D}$ respectively. Then, the CF is used to generate a subgraph per community from the derived within degree sequence. There is still however the need for an extensive rewiring step for forming the external edges, since the derived between degree sequence is first used by the CF model to generate a set of edges; then those edges that fall within communities are rewired until none of them is a within link. Similar to the original model, this generation process also uses CF which is an unrealistic network model. However, since this generation process is tangled with the degree sequence, it is less trivial how to substitute the CF model in this modified extension. Furthermore, unlike the original model, it results in all nodes having the exact same fraction of within/between edges, which is artificial.

The ***block two-level Erdős-Rényi (BTER)*** model proposed in [145], directly incorporates communities in the generative model, whereas their networks are scale-free collections of ER subgraphs as communities. The BTER starts with a pre-processing where nodes are distributed into affinity blocks (communities) and each node is assigned a degree and a clustering coefficient (the latter determines the portion of inter(between) to intra(within) community links), which are input to the model. Then in the first phase, local links are formed within each block(community) according to a constant probability computed for that block, and in the second phase, global (between) links connect communities together from nodes that have connections less than their assigned degree. If the input degree distribution follows a power law, the resulted networks are shown to be scale-free. The degree and clustering coefficient are required by this model; which can be randomly generated to be matched against and generate sample networks for community mining benchmarks [72]. The main idea of FARZ, presented in the Section 4.4, is similar to the BTER model, *i.e.*, community structure is present from the start and affects how edges are formed. However, FARZ directly extends the network evolution models by incorporating the extra factor of communities. Moreover, FARZ is defined based on relevant and intuitive parameters that directly control different growth factors in networks, unlike having distributions of degree and clustering coefficient as input. This provides flexibility and expressiveness, and makes FARZ a perceptive and simple alternative benchmark generator for community evaluation.

### 4.2.3 Attributed Network Models with Explicit Modular Structure

Evidences of homophily in most real networks suggest that connections are formed with a bias in favor of similar characteristics/attributes of nodes (including their degrees). Some network generator models assume that links between nodes are formed solely based on their attributes [69, 163]; whereas others augment the classic network generation models by incorporating the effect of attributes. For example, ***Social-Attribute Network (SAN)*** [54] follows an attribute-augmented preferential attachment (the probability of a node $u$ linking to a node $v$ depends on the degree of node $v$ as well as the number of attributes $u$ and $v$ have in common) as well as an attribute-augmented triangle-closing (randomly connecting $u$ with its 2-hop social neighbours, where the hop could be through attribute nodes). This model however does not provide an explicit community structure. In fact, very few generator models allow one to build a network having both an explicit community structure and attributes associated with the nodes. Dang [37] proposes a simple generation model in which for each new node, its attributes and community membership are independently sampled from a multinomial and normal distribution, respectively. Then a specified number of edges are formed, where the probability of a node $u$ linking to an existing node $v$ depends on (the multiplication of) the degree of $v$, and the attribute similarity of $u$ and $v$, as well as the attribute similarity of the classes that $u$ and $v$ belong to. In a similar effort, we proposed a generator in [82], which augments and combines the BA and BTER models; whereas it follows a local preferential attachment, which states that a node is more likely to create connections with nodes that have high degrees which are also close-by. This model also incorporates attributes assuming that nodes that belong to the same community should be more similar in terms of their attributes; also when forming long range edges between communities, it considers the similarity of nodes in terms of their characteristics(attributes).

On the other hand, there exists a family of generative network models [119, 168], which model networks with attributes, and consider communities as latent parameters. These models are not dedicated to synthesizing networks, and the main objective is to discover the parameters of the model when fitted to a given real network instance, *i.e.*, to learn the latent parameters of the model for that instance. The discovered latent parameters are used to infer knowledge from the graph, *e.g.*, to determine its community structure. For example Yang et al. [168] propose a generative Bayesian model to learn the latent parameters of attribute models and communities assuming that the graph and the attributes of nodes are observed and independent given community structure. Although the model they proposed is generative, and could be used to sample networks similar to the given real world network, when synthesizing benchmarks for their evaluation, they use the FF model to generate the graph and then randomly generate attributes for the nodes. Similarly, [119] propose a generative Bayesian model for sampling clustered attributed networks, and infer clusters in such networks based on a variational approximation approach.

This chapter is focused on the basic case of generating realistic modular networks. When the desired structural properties are reached, then the proposed models could be augmented to also incorporate the attributes. In the following, we first discuss how to improve the LFR model, which has been used extensively in the evaluation of community mining algorithms [23, 41, 146]. Then we present the FARZ model, in Section 4.4, which provides a simple and effective alternative.

## 4.3 Generalized 3-Pass Model

We generalize and modify the original LFR benchmarks: *1)* to start with any network model, so that it could be plugged in with more realistic network models; and more importantly, *2)* to assign nodes to communities in a more efficient way, so that the resulted assignments require fewer rewirings, hence keeping the properties of the original network intact. The generalized benchmark generator has three phases: first, it realizes a network according to a network model $\mathcal{M}$, which has a parameter set $\theta^G$; second, it creates communities based on a given parameter set, $\theta^C$, and assigns the nodes to these communities; and third, it overlays the community structure on the network, to satisfy the constraints given in $\theta^C$. In the original LFR, we have $\mathcal{M} = CF$, with parameters $\theta^G =$

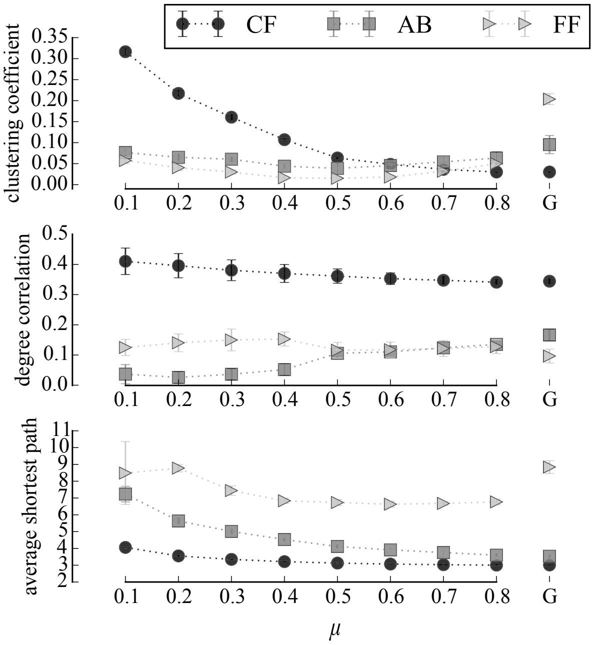


***Figure 4.1:*** *Benchmarks created by the generalized 3-pass model using different start network models: CF, AB, and FF. Properties of the synthesized networks are plotted as a function of the mixing parameter μ, in different subplots. The properties are also reported for the start network (marked by G),* i.e.*, before overlay and rewiring phases. Results are averaged over 10 simulations,* i.e.*, realizations of the networks.*

$\{N, k_{avg}, k_{max}, \gamma\}$, which are respectively: the number of nodes, the average degree, the maximum degree and the exponent of the power law degree distribution. These parameters are used to determine and sample a degree sequence, from which the graph is then synthesized using the Configuration (CF) model. We substitute the CF model with two models which are more realistic, *i.e.*, the *Barabási and Albert (BA)* model [10] and the *Forest Fire (FF) model* [84, 85].

Figure 4.1 experimentally compares the properties of the networks realized by the LFR variations when using these alternative network models. The properties compared are the average clustering coefficient of the nodes, the degree correlation coefficient (Pearson correlation for degrees of the connecting nodes), and the average shortest path distances between all the pairs of nodes. The parameters of the models are chosen so that the initial networks have similar degree distributions. In more detail, for CF, BA, and FF we respectively have: $\theta^G = \{N : 1000,\ k_{avg} : 15,\ k_{max} : 50,\ \gamma : 3\}$, $\theta^G = \{N : 1000,\ m : 2\}$, and $\theta^G = \{N : 1000,\ p : 0.1,\ rp : 0.0\}$.

In the top subplot of Figure 4.1, we can see that the clustering coefficient of the CF model is almost zero for the initial network (marked by $G$), and the rewiring actually brings some modularity to the network and increases the average clustering coefficient, but only for small mixing parameters, *i.e.*, when communities are well-separated and many links are rewired to lie inside communities. However, as $\mu$ increases, *i.e.*, when communities get more tangled together, the average clustering coefficient decreases in the generated network, and reaches zero for large values

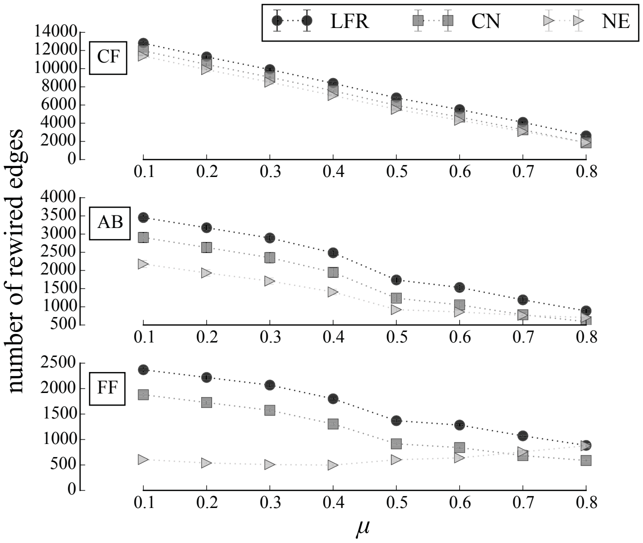


***Figure 4.2:*** *Comparing the number of edges rewired using each of the three node assignment variations: LFR (original), CN (common neighbour), and NE (neighbour expansion). The subplots correspond to the different initial network models,* i.e., *CF, AB, and FF.*

of $\mu$. This is also true if we start with a network with high clustering coefficient, such as $FF$, as the network structure is extensively changed after the rewiring phase to overlay the communities. We can see similar effects on the two subsequent subplots, where the rewiring changes the average degree correlation and shortest paths of the original synthesized network. Hence in the next step, we try to reduce the amount of rewirings necessary to overlay the communities.

In the original rewiring/overlay phase, nodes are assigned to communities uniformly at random. Here, we propose two modified variations that result in far less rewirings in the subsequent overlay procedure. More specifically, we examine two variations: *1)* Common neighbour ($CN$) assignment, *i.e.*, probability of joining a community is proportional to the neighbours a node has in that community; *2)* Neighbour expansion ($NE$) assignment, *i.e.*, after assigning a node to a community chosen uniformly at random, also assign all of its neighbours to that same community, and continue until the community is full according to its size which is predetermined based on $\theta^C$. These procedures are described in details in Appendix C.1.1. Figure 4.2 compares the percentage of edges rewired using these three node assignment approaches.

Similar to Figure 4.1, the subplots of Figure 4.2 are a function of the mixing parameter $\mu$ *i.e.*, the constraint used in the rewiring/overlay phase. The complete parameters used to overlay communities are $\theta^C = \{\mu, \beta : 2, c_{min} : 20, c_{max} : 50\}$. The three latter parameters determine the capacity of communities; which are respectively: exponent of the power law distribution for community sizes, the minimum size, and the maximum size for communities. These three parameters describe a truncated power law distribution from which the community sizes are then sampled. We can see in Figure 4.2 that the amount of rewirings significantly reduces only when the initial network

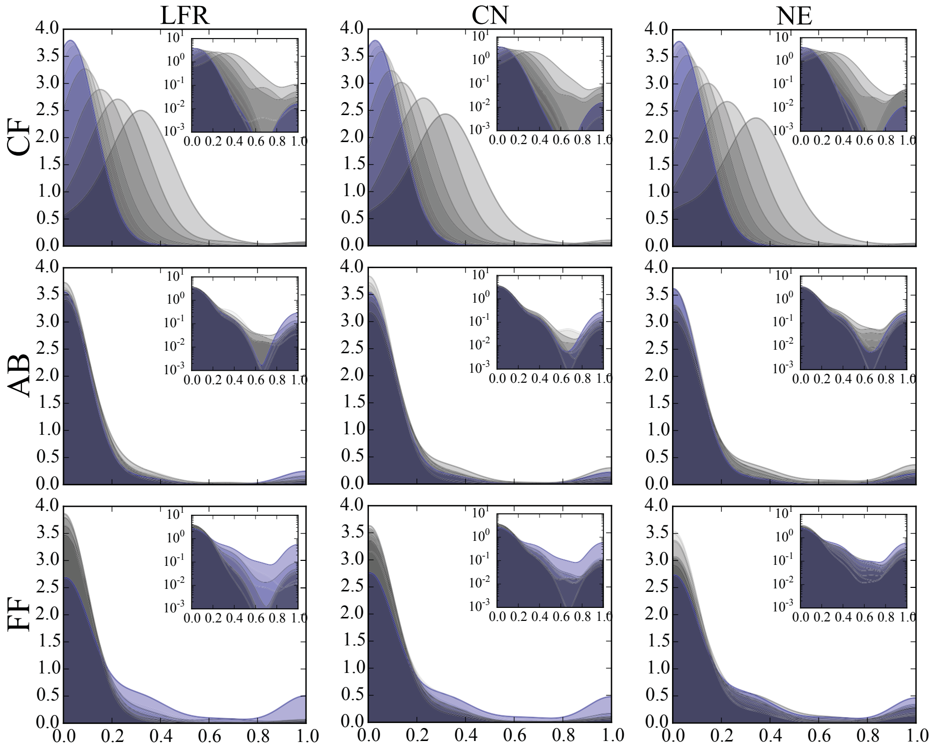
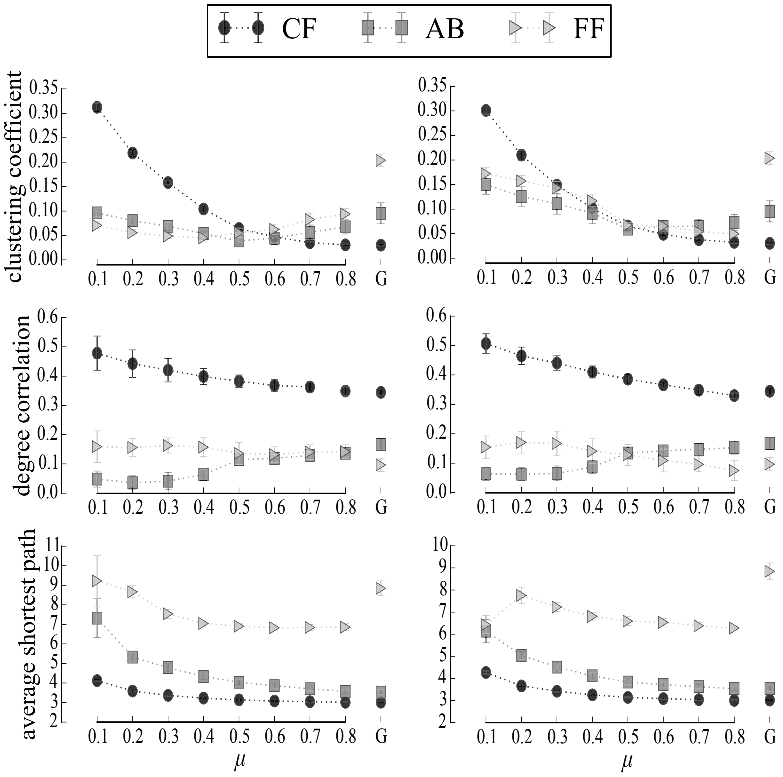


***Figure 4.3:*** *The effect of rewiring on the probability density function (pdf) of the clustering coefficient. The blue pdf shows the distribution of clustering coefficients of all nodes in the initial network. The grey pdfs correspond to different values of $\mu$. The insets represent the same graph on a log scale y-axis.*

model is *FF* and the assignment approach is *NE*. In other words, to improve over the original LFR, we require both a realistic initial model and an efficient node assignment technique.

Figure 4.3 illustrates the effect of changing the mixing parameter $\mu$ on the clustering coefficient of nodes, for the nine variations of LFR derived from our generalization, *i.e.*, when using different initial network models and different assignment approaches to overlay the communities. We can see that the distribution of clustering coefficient is preserved the best for the combination seen in the bottom right subplot, *i.e.*, when the network start model is *FF* and the assignment approach is *NE*; since in this case the network has a better clustering distribution from the start, and the assignment of nodes preserves those clustered nodes. Hence, this proposed variation generates networks with more realistic clustering coefficient distribution and is more favourable to the original LFR.

Figure 4.4 compares the averages of the clustering coefficient, degree correlation and shortest paths for the networks synthesized using the *CN* and *NE* assignment variations; which is on par with Figure 4.1 that shows the same properties for the random node assignment approach of the original LFR. We can see that the *CN* and *NE* variations better preserve the properties of the network when compared to the original LFR. For the clustering coefficient in particular, we



(a) CN

(b) NE

***Figure 4.4:*** *Comparing the properties of networks using the two assignment variations, CN on the left and NE on the right; which compares against the Figure 4.1 that uses the original assignment approach in LFR.*

see improvement for larger values of $\mu$ in $CN$; however, this does not hold as $\mu$ decreases and the rewiring becomes more intrusive. For the $NE$ assignment on the other hand, the clustering coefficient of the original network is preserved and increases as communities become denser by decreasing the mixing parameter $\mu$. The difference of these variations is not noteworthy in the cases of degree correlation and shortest paths.

Although exhibiting more realistic properties, the benchmarks generated by these variations, similar to the original LFR benchmarks, enforce communities later on the network; which is contrary to their definition as the natural structure underlying the networks. A more important issue with the LFR benchmarks is however their lack of expressiveness and flexibility. For example, it is not clear how to generate networks with negative degree correlation (dis-assortative networks) using the LFR, whereas many real world networks are known to be dis-assortative. In fact, although the LFR benchmarks accept many input parameters, these parameters are not relevant in most cases, and almost all the works that use LFR in their evaluation, rely on the few example settings first used by the authors when introducing these benchmarks [79].

## 4.4 FARZ Benchmark Model

Similar to the most classical network models, FARZ follows a growth pattern, *i.e.*, it gradually expands the network following different evolution patterns while incorporating an underlying community structure. The overall FARZ procedure is summarized in Algorithm 1. The input parameters of $n$, and $k$ respectively determine the total number of nodes, and the number of communities; whereas $m$ determines the number of edges added at each step, which controls the total number of edges ($nm$), overall density of the networks ($2m/n$), and the average degree ($2m$).

FARZ expands the network one node at a time. Each node $i$ added to the network is immediately assigned to $r$ communities, where $r = 1$ in case of non-overlapping communities. The probabilities of these assignments are proportional to the (current) sizes of the communities. This would apply a preferential attachment mechanism and ensure the heavy tail distribution for the community sizes. More formally, the probability of node $i$ joining community $u$ is determined as:

$$p(u) = \frac{|u| + \phi}{\sum\limits_{v}(|v| + \phi)} \tag{4.1}$$

where $\phi = 1$ ensures that empty communities also have a chance to recruit. It could also be changed to control the effect of preferential attachment and move toward having equal sized communities; since as $\phi$ increases, the distribution for sizes of communities becomes closer to uniform.

```
 1: $G \leftarrow Graph()$ { initialize empty graph}
 2: $C \leftarrow \{c_1 = \emptyset, c_2 = \emptyset \ldots c_k = \emptyset\}$ { initialize communities}
 3: for $i \in [1 \ldots n]$ do
 4:     $G.add\_node(i)$ { add node $i$ }
 5:     $assign(i, C)$ { assign $i$ to communities}
 6:     $connect(i, G, C)$ { add an edge from node $i$}
 7:     for $[2 \ldots m]$ do { add $m - 1$ edges}
 8:         $j \leftarrow select(G.nodes)$ { select node $j$ from $G$}
 9:         $connect(j, G, C)$ { add an edge from node $j$}
10: return G, C
```

**Algorithm 1:** FARZ Generator (n, m, k)

After node $i$ joins the selected community or communities, it gets connected to the network by forming an edge (line 6 of Algorithm 1). This is to ensure that node $i$ connects to at least one node in the network and there are no singletons in the synthesized network, *i.e.*, unconnected nodes with the degree of zero. Then, $m - 1$ nodes, from the existing nodes within the network, are randomly selected and get a chance to also form connections. These new connections may or may not involve the newly added node $i$. This is a better alternative to forming all $m$ edges from the newly added node, since it does not limit the minimum degree of nodes to $m$, whereas it actualizes the preferential attachment effect through rich get richer pattern. In other words, adding constant number of edges at each round results in an accumulative advantage for the nodes which are added earlier to the network, since they get more chances to be selected and form connections, which naturally enforces the heavy tail degree distribution observed in real networks.

In Algorithm 1, the function $assign()$, in line 5, and $select()$, in line 8, are straightforward. The latter selects a node uniformly at random; whereas the former randomly chooses community assignments based on the probabilities in Equation 4.1. Algorithm 2 describes the function $connect()$, called in line 6 and 9 of Algorithm 1. This function enforces the community structure and controls the edge formations.

```
1: if random < $\beta$ then
2:     $c \leftarrow select(\{c,\ \forall c \in C \land i \in c\})$ { select a community from memberships of node i}
3: else
4:     $c \leftarrow select(\{c,\ \forall c \in C \land i \notin c\})$ { select a community that doesn't include node $i$}
5: $j \leftarrow choose(\{j,\ \forall j \in c \land j \neq i \land (i, j) \notin G.edges\})$ { choose a node from selected community}
6: $G.add\_edge(i, j)$
```

**Algorithm 2:** FARZ Connect (i, G, C)

When forming an edge, a node first selects a community, and then connects to a node within that community. More specifically, node $i$ forms its connection within the communities that it is a member of, with probability $\beta$, and connects to nodes from other communities with probability $1 - \beta$. The control parameter $\beta$ hence determines the strength of the overall community structure, and is analogous with the mixing parameter $\mu$ in the *LFR* model.

The function $choose()$, in line 5 of Algorithm 2, determines the probability of forming an edge from node $i$ to node $j$, which can be defined to depend on different driving factors. Here we consider two factors: the number of their common neighbours (Equation 4.2), and the similarity of their degrees (Equation 4.3), *i.e.*,

$$p_{ij} \propto \sum_{k=1}^{n} w_{ik} w_{jk} \tag{4.2}$$

$$p_{ij} \propto (d_i - d_j)^2 \tag{4.3}$$

where $w_{ij}$ represents the edge weight between node $i$ to node $j$, and $d_i = \sum_{k=1}^{n} w_{ik}$. Equation 4.2 enforces "triadic closure", which is known as a natural mechanism for edge formation in real networks [14], and results in the high clustering coefficient observed in real networks. Equation 4.3 implements the assortative mixing, *i.e.*, tendency of similar nodes to connect. Here we consider degree assortativity, measured by degree correlation. However, this can be extended for attributed networks where homophily is also a factor in the edge formation, *e.g.*, by measuring the cosine similarity of the attributes associated to the nodes. A function $\varphi(.,.)$ is used to combine the effect of Equation 4.2 and Equation 4.3, *i.e.*, $\varphi : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$. Here, we simply use $\varphi(x, y) = x^{\alpha} y^{-\gamma}$

to have both factors in effect; where $\alpha$ and $\gamma$ respectively control the effect of Equation 4.2 and Equation 4.3. The overall probabilities are hence computed as:

$$p_{ij} \propto (\sum_{k=1}^{n} w_{ik} w_{jk})^{\alpha} ((d_i - d_j)^2 + 1)^{-\gamma} + \epsilon \tag{4.4}$$

where $\epsilon$ is a small number that accounts for unlikely edges, and is particularly required at the initial stages. Different choices of $\varphi$ result in structurally different networks, however all these generated networks would have the heavy tail distributions for the degree of the nodes and community sizes, and a built-in community structure. The control parameter $\gamma$ indicates whether the degree correlation should be positive or negative in the generated network, *i.e.*, whether larger $\Delta d_{ij}$ decreases or increases $p_{ij}$, respectively. This makes FARZ able to generate both assortative and disassortative networks, which is a distinct advantage over the previous models.

Real networks are known to exhibit both negative and positive degree correlation [101]. In social networks, for instance, a positive degree correlation is often observed, which indicates that nodes with similar degrees tend to connect to each other; whereas some biological networks are known to be disassortative, *i.e.*, hubs with high degrees often connect to nodes with small degrees. For example, Figure 4.5 illustrates properties of four widely studied real networks; where all exhibit strong negative or positive degree correlation.

### 4.4.1 Comparing Properties of Networks

Figure 4.6 and 4.7 illustrate the basic properties for synthetic networks sampled from AB, FF, LFR, and FARZ models; these figures correspond to the properties reported for real networks in Figure 4.5. In Figure 4.6, we observe zero or small clustering coefficient for networks generated with the AB, and LFR models; which is due to the fact that these models do not factor in the transitivity. The FF model, however, directly evolves the network by connecting each node to the neighbours of its connections, *i.e.*, closing triangles, hence it achieves high clustering coefficient. The FF model, however, does not factor in assortativity, nor do the AB and CF models, and hence these models generate networks with zero degree correlation. These observations are inconsistent with the patterns observed for real world networks, as also shown in Figure 4.5. On the other hand, Figure 4.7 shows that the sample networks generated by FARZ comply well with the properties of real networks. They have small diameter, heavy tail degree distribution, and high clustering coefficient; moreover, they can exhibit positive or negative degree correlations based on the parameter $\gamma$ which directly controls the assortativity.

Figure 4.8 reports the average of properties for the synthesized FARZ networks, which is plotted as a function of $\beta$ that controls strength of the community structure. Here, we compare the four parameter settings of Figure 4.7, when $\beta$ varies, and the results are averaged over 10 realizations of the networks for each $\beta$. We plot the results for $\beta \in [0.5, 1]$, that is where a community structure

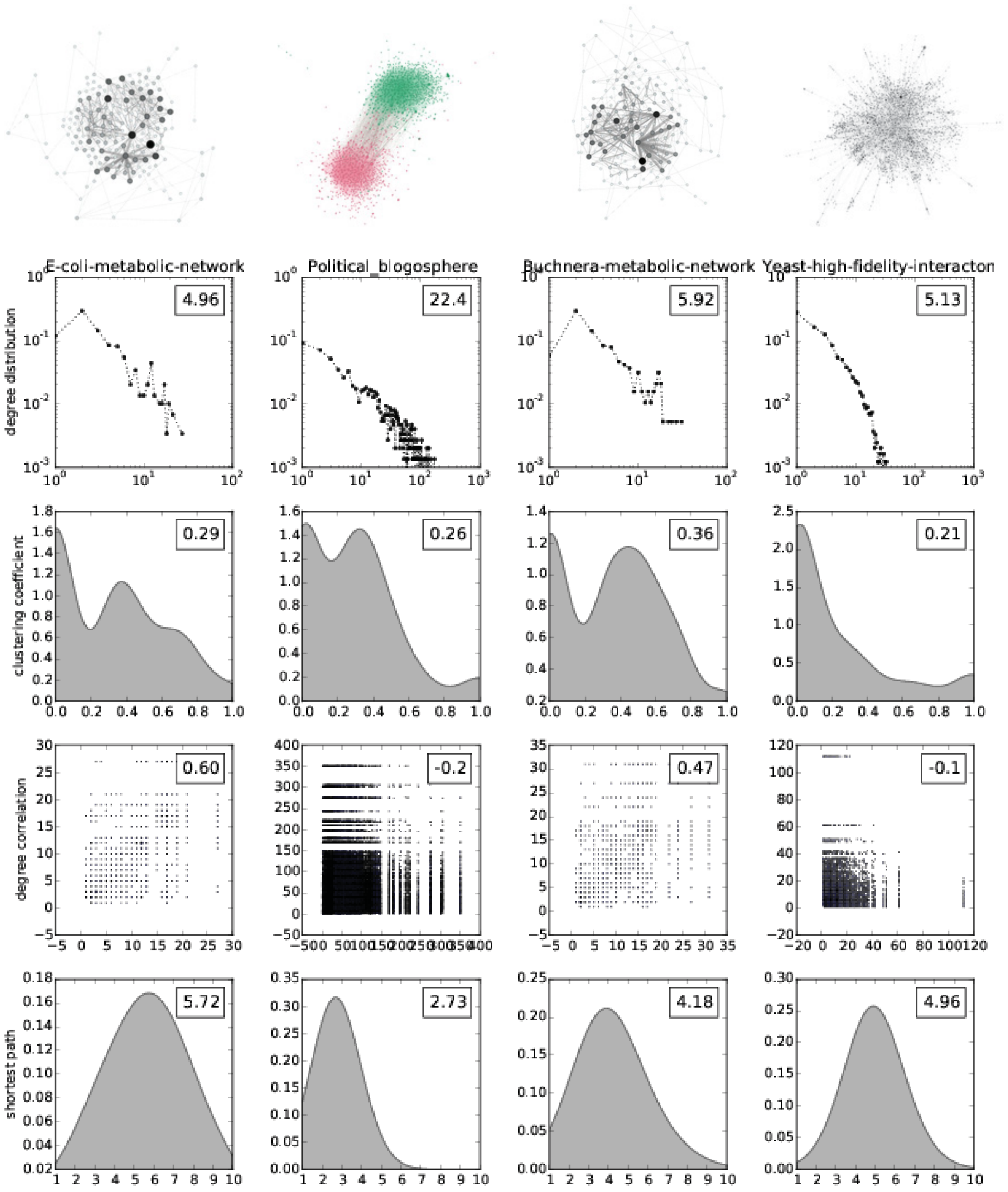


*__Figure 4.5:__ Basic properties of four example real world networks with positive and negative degree correlations. The insets respectively show the average degree, average clustering coefficient, degree correlation (Pearson correlation between the degrees of connected nodes), and the average shortest paths. The corresponding graphs are also visualized at the top; whereas if coloured, different colours represent the available community labels.*

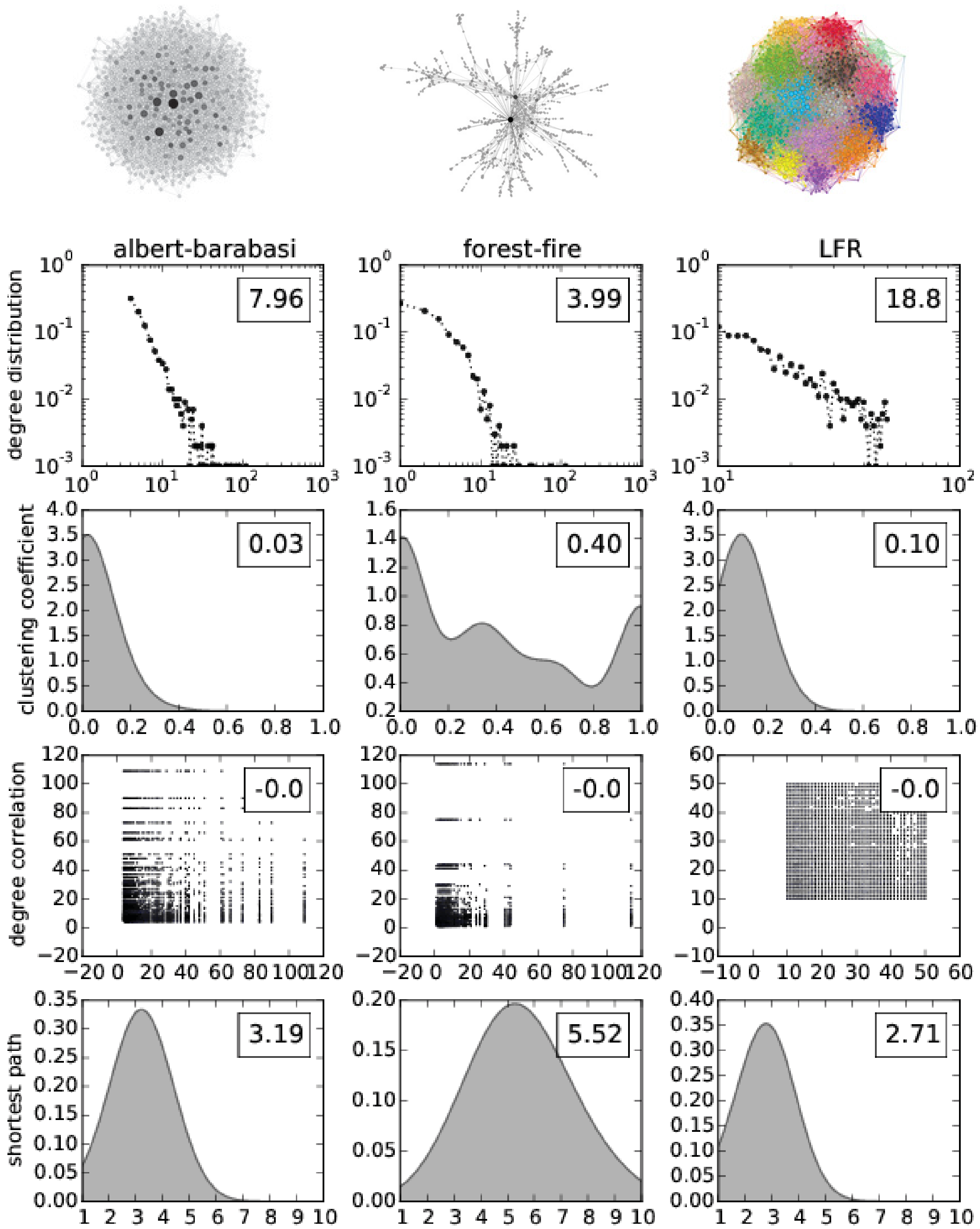


***Figure 4.6:*** *Basic properties of three synthetic networks, with 1000 nodes. The network is generated with* $m = 4$ *for AB model. Where the FF parameters are* $\{p : 0.4, rp : 0.2\}$*, and for the LFR model we used the commonly used setting of* $\{k : 20,\ k_{max} : 50,\ t_1 : 2,\ t_2 : 1,\ \mu : 0.4,\ c_{min} : 20,\ c_{max} : 100\}$*, and the original implementation provided by the authors.*

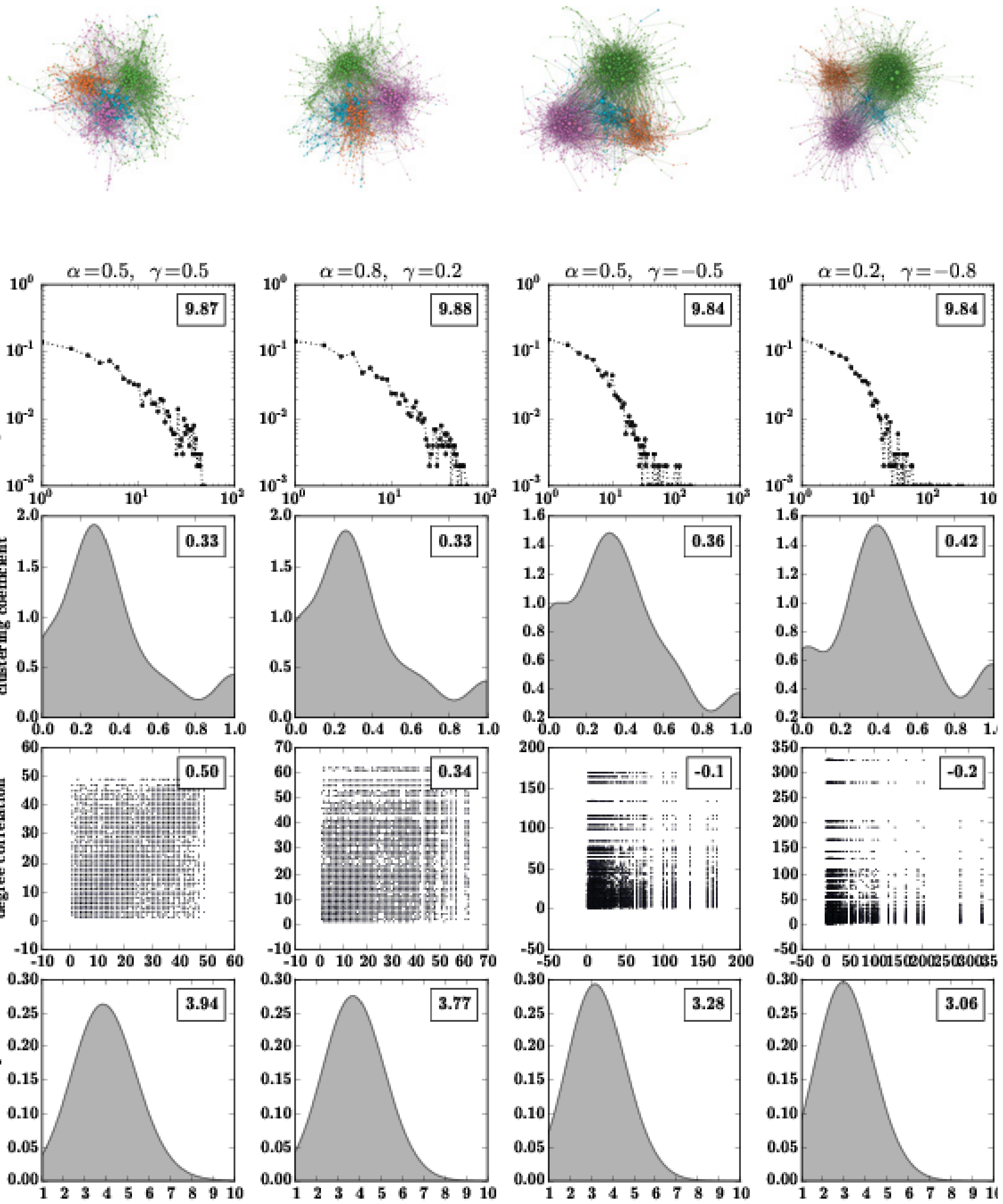


*__Figure 4.7:__ Basic Properties of four sample networks generated by FARZ for different combination values of $\alpha$ and $\gamma$. The complete parameters setting is $\{n$:1000, $k$:4, $m$:5, $\beta$:0.8, $\phi$:1, $r$:1, $\epsilon$ : $1e-07\}$.*

exists within the network, *i.e.*, the chances of edge formation is higher within the communities that outside of them. We can see in this plot that the FARZ benchmarks are consistent, as opposed to the LFR (as seen in Figure 4.1), *i.e.*, all the networks synthesized by FARZ exhibit degree correlation and clustering coefficient, regardless of the strength of the underlying community structure.

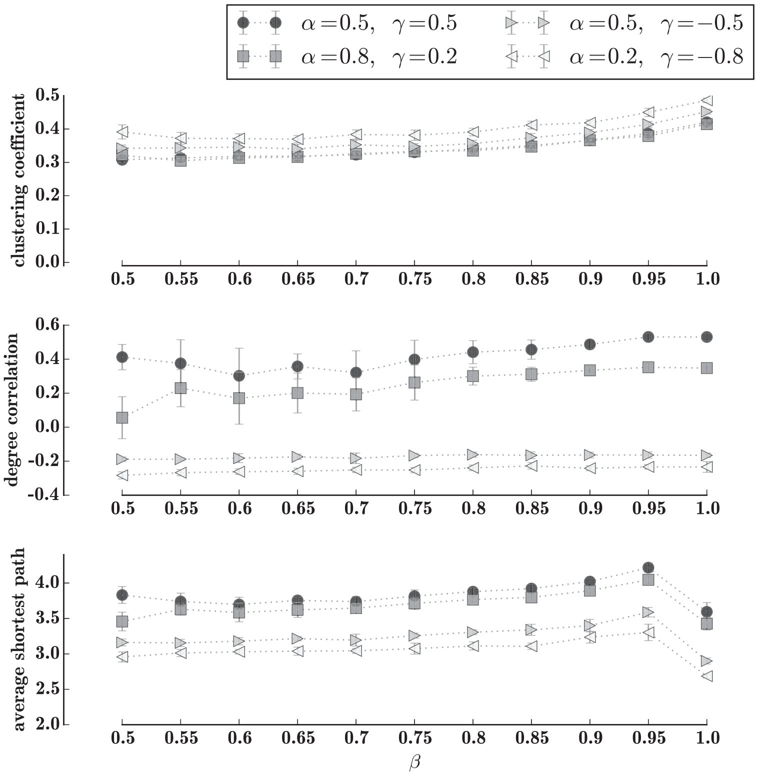
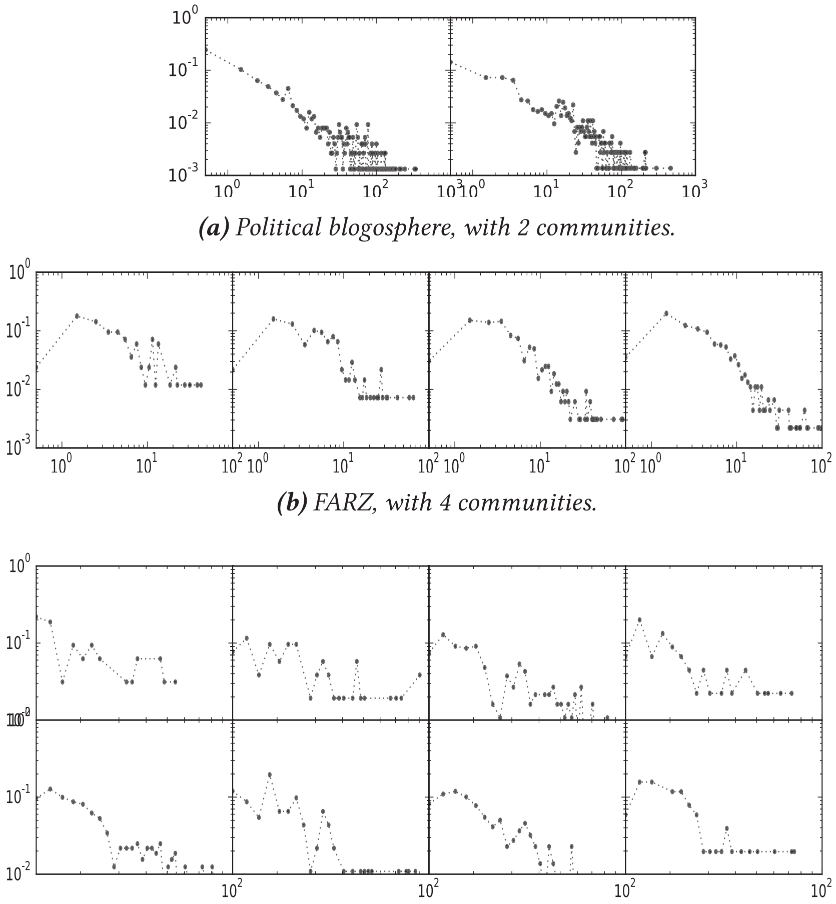


***Figure 4.8:*** *Basic properties of the synthesized FARZ networks plotted as a function of $\beta$,* i.e., *the probability of edges to form within the communities. This figure corresponds to Figure 4.1 and 4.4.*

### 4.4.2 Comparing Properties of Communities

All the comparison so far focuses on the general properties of the synthesized networks. We can further look at the properties in each community, and compare the patterns with what is observed in the real networks. More specifically, in Figure 4.6 and 4.7, we see that networks generated with both LFR and FARZ model have heavy tail degree distributions. In Figure 4.9, we compare the degree distributions inside each community created by these benchmark models. We can see that in the example real world network, for which the community labels are available (*i.e.*, Political_blogosphere in Figure 4.5), the degree distributions per community follows the same heavy tail distribution as the overall network (Figure 4.9a). The communities generated by FARZ benchmark, comply with this pattern and follow a heavy tail degree distribution as well (Figure 4.9b), *i.e.*, they comply with the observation in the real network example. However, we do not observe a clear heavy tail trend for the communities generated by the LFR benchmark (Figure 4.9c).

In Figure 4.9, the FARZ network (Figure 4.9b) corresponds to the third column in Figure 4.7,



***(a)** Political blogosphere, with 2 communities.*

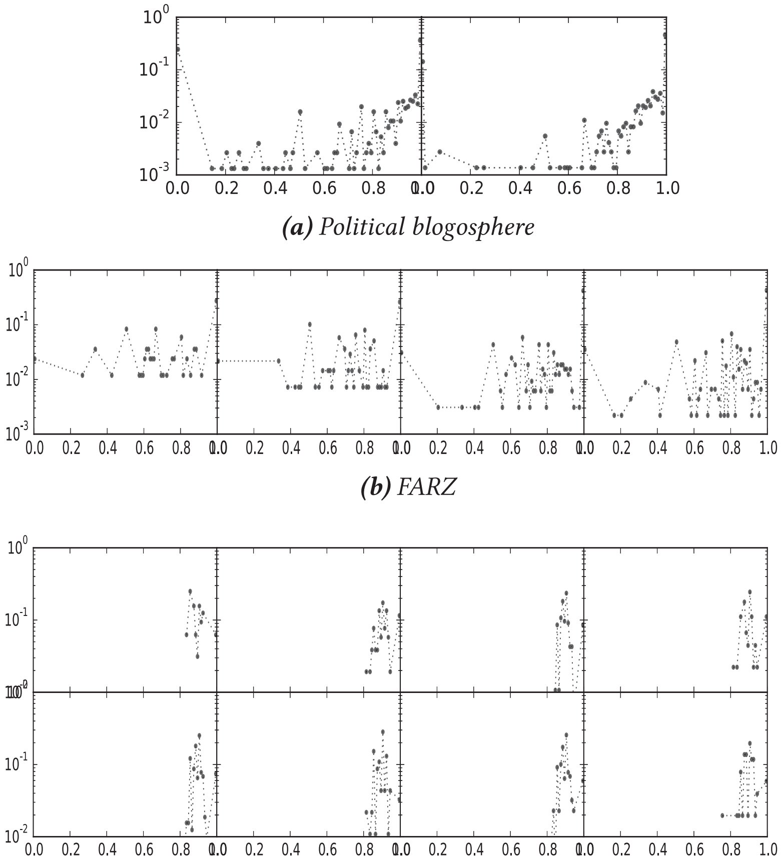
***(b)** FARZ, with 4 communities.*

***(c)** LFR, with 8 communities.*

***Figure 4.9:** Degree distributions per community for an example real network (top), and two synthetic networks generated by FARZ (middle) and LFR (bottom). Each subplot reports the degree distribution inside a community.*

*i.e.*, when $\alpha = 0.5$ and $\gamma = -0.5$. Other settings in Figure 4.7 exhibit the same heavy tail degree distribution pattern for inside the communities. The LFR network (Figure 4.9c) has similar parameters as the network reported in the third column of Figure 4.6, except the maximum community size is increased to 500 to get a smaller number of communities, to be able to better plot and compare the results. The plot for the exact network of Figure 4.6, which has 17 communities, shows similar patterns but requires more space for plotting, and is reported in Appendix C.2.2.

In Figure 4.10, we compare the ratio of within to total connections for the nodes in each community, *i.e.*, the degrees of the nodes within their community divided by their degree in the whole network; this corrsponds to $1 - \mu$ in the LFR. Here we can see that for the real world network example (Figure 4.10a), as well as the networks synthesized with FARZ, this ratio of within to total edges varies for the nodes inside each community between 0.0 and 1.0. However, this is not the case in the LFR example. LFR gets this ratio as an input parameter, *i.e.*, mixing parameter $\mu$. In



*(a)* *Political blogosphere*

*(b)* *FARZ*

*(c)* *LFR*

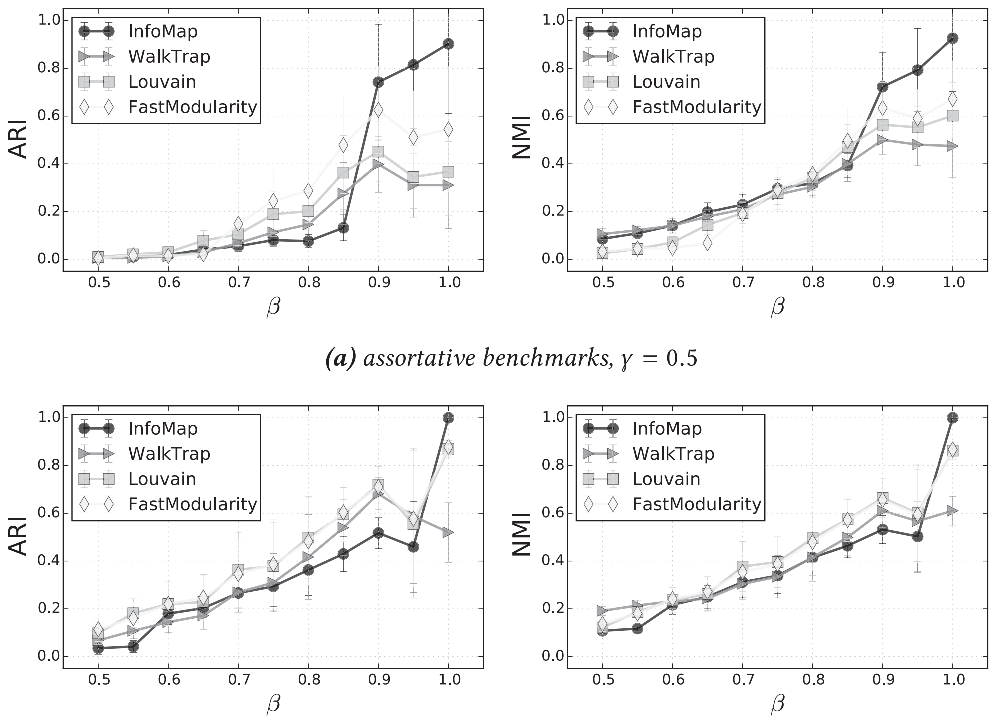
***Figure 4.10:*** *Distributions of within to total edges for the nodes in each community. Plots correspond to the networks of Figure 4.9.*

LFR, all the nodes within a community have the same degree of membership, which is artificial and unlike the observed pattern in real networks.

## 4.5 Application and Flexibility

In this section we show the application of FARZ in validating and comparing community mining algorithms. More specifically, we compare and rank selected community mining algorithms on the benchmarks generated by FARZ, where we tune its flexible parameters to rank the algorithms in different and meaningful experimental settings. In more detail, the selected algorithms are: Louvain [16], WalkTrap [120], FastModularity [104], and InfoMap [140] for disjoint clusterings; and COPRA [55], MOSES [94], OSLOM [81], and BIGCLAM [167] for overlapping clusterings in Section 3.6.3. The authors' original implementations are used for the methods, with no parameter tuning (defaults are used); and the reported results are averaged over ten runs.

The agreements (higher is better) are measured and reported with both *ARI* (Adjusted Rand



***(a)** assortative benchmarks,* $\gamma = 0.5$

***(b)** disassortative benchmarks,* $\gamma = -0.8$

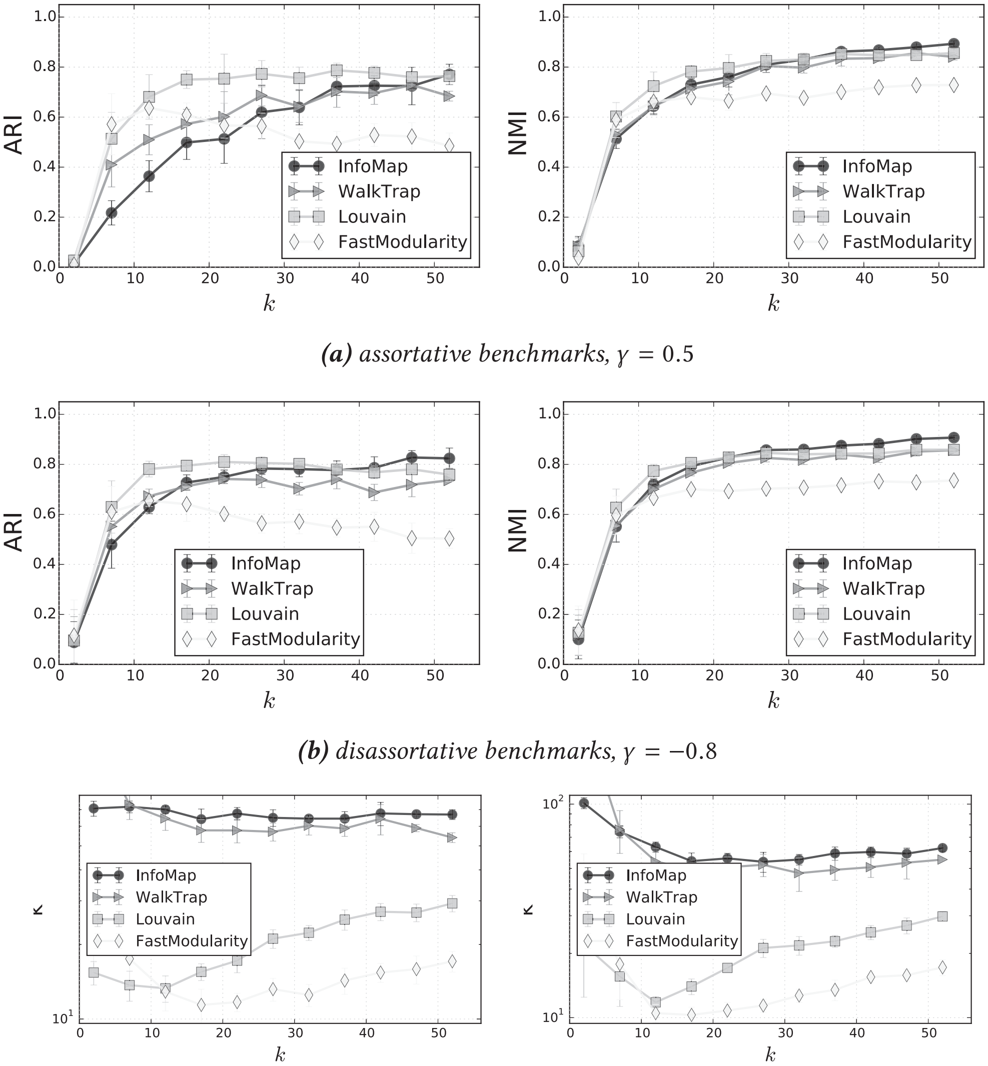
***Figure 4.11:** Performance of community mining algorithms on benchmarks with **degree assortativity v.s. degree disassortativity**; plotted as a function of the strength of the built-in community structure,* i.e.*, determined by $\beta$. Results are averaged over 10 runs. The parameter settings correspond to the first (4.11a) and last (4.11b) columns of Figure 4.7.*

Index) and *NMI* (Normalized Mutual Information) indexes; *NMI* is a widely used agreement index for comparing clusterings, which is known to be biased with the number of clusters, whereas *ARI* is less common, but more appropriate [123]. We described the clustering agreement measures in depth in Chapter 3.

### 4.5.1 Effect of the Degree Assortativity

Here, we compare performance of the community detection algorithms on the benchmarks with degree assortativity, *i.e.*, positive degree correlation (common in social networks); and degree disassortativity, *i.e.*, negative degree correlation (common in biological networks). Figure 4.11 shows the comparison of the four selected algorithms on the assortative and disassortative FARZ benchmarks; where overall the selected algorithms perform better on the disassortative benchmarks.

In the case of assortative networks (Figure 4.11a), FastModularity outperforms the other three methods when communities are not predominant, *i.e.*, for $\beta < 0.9$. From $\beta = 0.9$, Infomap becomes the best performing method, which is after a sharp transition from its poor performance for the less predominant communities. In the case of disassortative networks (Figure 4.11b), the performance of FastModularity is on a par with Louvain, which are superior to InfoMap until communities are well separated, *i.e.*, $\beta = 1$. These results are interesting since the InfoMap algorithm is known to be the best performing method from the selected set when evaluated on the LFR benchmarks [76, 123].



***(a)** assortative benchmarks, $\gamma = 0.5$*

***(b)** disassortative benchmarks, $\gamma = -0.8$*

***(c)** number of communities in the results, $k'$, v.s. the true number of communities, $k$ in the ground-truth*

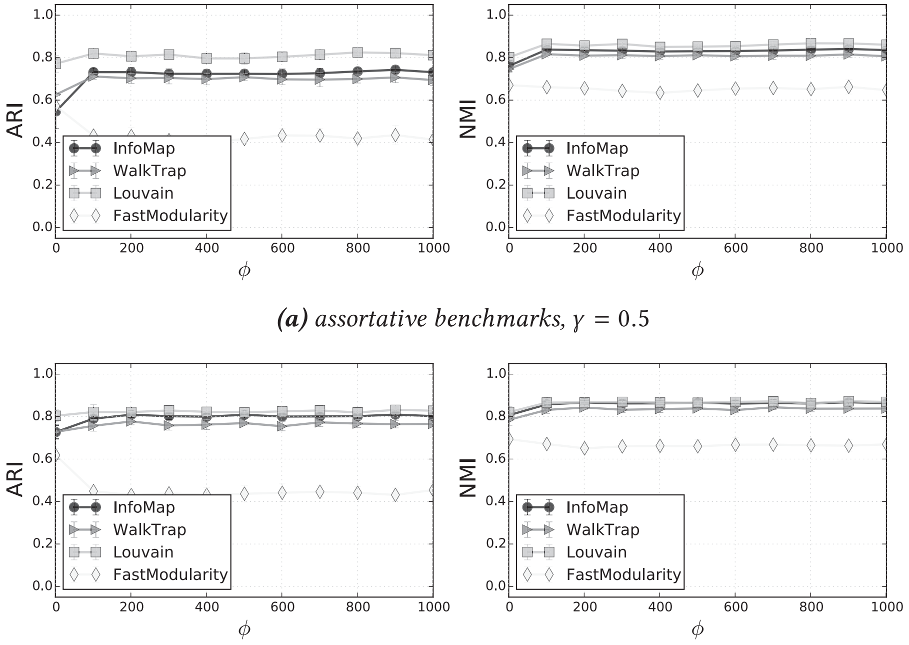
***Figure 4.12:** Performance of community mining algorithms on benchmarks with different **number of built-in communities**. Settings correspond to the Figure 4.11, and $\beta$ is fixed to 0.8.*

### 4.5.2 Effect of the Number of Communities

Here, we compare the algorithms on benchmarks with different numbers of built-in communities, by changing the parameter $k$. Figure 4.12 shows the results. We can see that all the algorithm have difficulty when the number of communities is small, *i.e.*, $k < 10$. Unlike other algorithms, the performance of FastModularity also drops as the number of communities increases. In the assortative networks in particular, the Louvain method is more consistent to the change of the number of communities. In Figure 4.12c, we can also observe that all these method fail to detect the true number of communities in the ground-truth, and the number of detected communities($k'$) seems to be independent of the true number of communities in the ground-truth($k$), particularly for InfoMap and WalkTrap and when $k$ is large.

### 4.5.3 Effect of Variation in Community Sizes

Here, we tune the parameter $\phi$ to change how well-balanced communities are in sizes, *i.e.*, move the distribution of community sizes from heavy tail to uniform. Figure 4.13 shows the comparison results. Similar to the effect of number of communities, FastModularity seems to be the least consistent method when the distribution of community sizes changes. While Louvain seems to be the superior method particularly in the assortative setting.



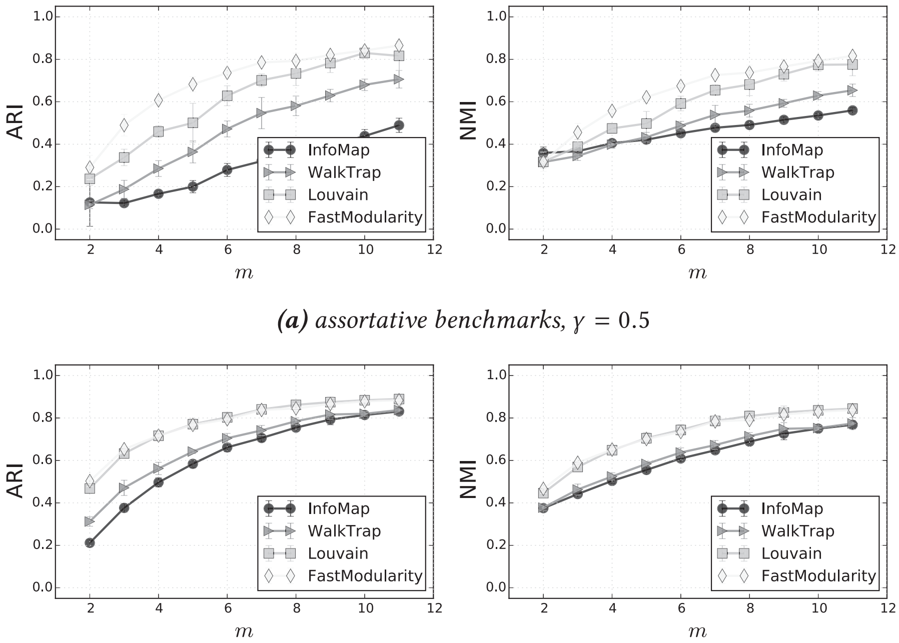
***(a)** assortative benchmarks, $\gamma = 0.5$*

***(b)** disassortative benchmarks, $\gamma = -0.8$*

***Figure 4.13:** Performance of community mining algorithms as a function of how equal are the **sizes of communities**. Settings correspond to the Figure 4.11, except $k$ that is increased to 20, to have more community sizes for the plot.*

### 4.5.4 Effect of the Density of Networks

Here, we tune the parameter $m$ to change how many connections nodes have on average, *i.e.*, move from sparse to less sparse networks and examine how the performance of different algorithms are affected by changing the density of benchmarks. In the results reported in Figure 4.14 we



***(a)** assortative benchmarks, $\gamma = 0.5$*

***(b)** disassortative benchmarks, $\gamma = -0.8$*

***Figure 4.14:** Performance of community mining algorithms on benchmarks with different **density**. Settings correspond to the Figure 4.11, and $\beta$ is fixed to 0.8.*

see that overall the algorithms perform better as networks become denser, *i.e.*, when the average degree of nodes increases, *i.e.*, when *m* increases. The performance boost is more significant for FastModularity, Louvain, and WalkTrap algorithms, and particularly in the assortative setting.

### 4.5.5 Effect of the Overlap

Figure 4.15 compares the performance of four overlapping community detection methods on the FARZ benchmarks with overlapping communities. We can see that all methods perform poorly, except COPRA, which is able to detect communities when the portion of overlapping nodes is small enough, i.e. $q < 0.2$. This is also interesting since these methods are shown to perform reasonably good on the overlapping extensions of LFR.

## 4.6 Conclusions

In this chapter, we introduced extensions to improve upon the shortcomings of the popular LFR network generator. We showed how these extensions refine the generated results towards more lifelike networks. We also introduced an alternative realistic and flexible benchmark generator for validating and comparing community detection methods, called FARZ. FARZ generates networks with built-in community structure, which can be compared, as a ground truth, against the

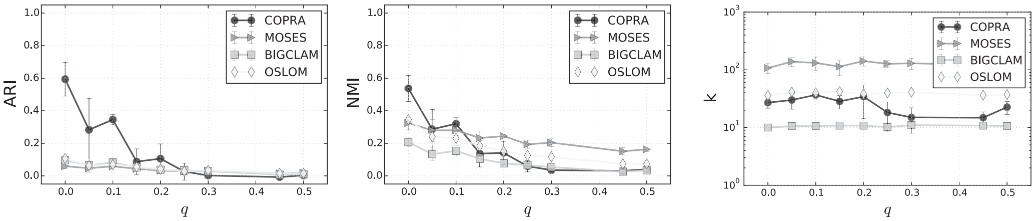


***Figure 4.15:*** *Performances as a function of the fraction of overlapping nodes, for the setting of* $\alpha = 0.5$, $\beta = 0.5$, $\gamma = 0.8$, *where the number of communities that each node can belong to is fixed to 3, and the portion of overlapping nodes (q) is varied from 0.0 (no overlap), to 0.5 (half of the nodes are overlapping).*

results of different community mining algorithms. The FARZ benchmark produces truthful networks, and the characteristics of the networks and communities synthesized by FARZ are similar to what is observed in real world networks. FARZ benchmark also incorporates intuitive parameters, which have meaningful interpretation and are easy to tune to control the experimental settings. More precisely, FARZ has three input parameters, $FARZ(n, m, k)$, which respectively determine the number of nodes, the (half of) average degree, and the number of communities. It also has four intuitive control parameters, $\beta$, $\alpha$, $\gamma$, and $\phi$; which respectively control the strength of the community structure, the clustering coefficient, the degree correlation, and the distribution of the community sizes. Tuning these parameters provides a means to generate a variety of realistic networks and presents different settings for comparing community detection algorithms, which can be used to determine under which settings each algorithm performs the best *e.g.*, assortative *v.s.* disassortative networks.

# Chapter 5

# Utilizing the Modular Structure of Networks

This chapter studies how the modular structure of networks could be utilized in different contexts. First, it overviews examples of applications which use module identification, a.k.a. community detection, in networks, including an e-learning setting where modules/communities effectively outline the collaboration groups of students, as well as the topics of their discussions, which are used to better monitor participation of students throughout a course; published in [128, 129, 132]. Second, it investigates the correlation between the attributes of the data-points and the relationships between these data-points, and presents a novel approach to derive alternative modular structures, whereas each alternative view is better aligned (is in better agreement) with a selected set of attribute(s); published in [125, 126].

## 5.1 Introduction and Example Applications

Analyzing the modular structure of networks, a.k.a. community detection, has a wide range of applications: from visualization and exploratory data mining to building prediction models [47, 121]. This analysis provides novel insights into the mesoscopic characteristics of networks, which are not available if we look at the network as a whole, *i.e.*, at a macroscopic level; or at the other extreme, focus on the properties of the individual nodes within the network at a microscopic level [121]. Consequently, community detection has been applied in networks from a very wide range of domains, including biology, marketing, epidemiology, sociology, criminology, zoology, etc.

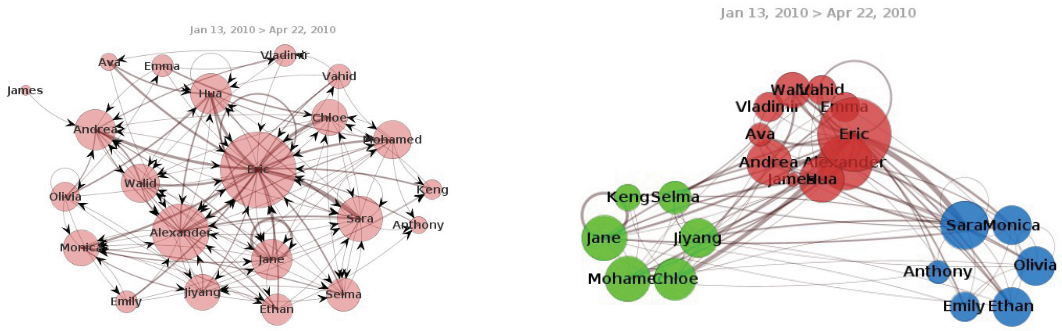
In biology, for example, Guimera and Amaral [56] extract the modules in metabolic networks of several species from different superkingdoms, using a simulated annealing based modularity optimization (see Section 2.2.1). They further assign a role to each metabolite based on how it is connected inside its own module (within-module degree) and also to the other modules (participation coefficient). They use the discovered modules and roles to build a cartographic representation of the metabolic network, which enables one to infer relevant biological information. Moreover, they study the evolutionary loss rate of different roles, and discover that ultra-peripheral metabolites (that have all their connections inside their modules) have the highest loss rate, whereas connector hubs (that connect to most of the other modules) are the most conserved across the species. In a follow up study, Zhao et al. [172] apply the same module identification technique on the metabolic network of Homo sapiens, and study the relationships between the modules themselves. They show that there exist core modules that perform the basic metabolism functions and behave cohesively in evolution, and periphery modules that only interact with few other modules and accomplish specialized functions, which also have a higher tendency to be gained/lost together through the evolution.

### 5.1.1 Role Mining in Social Networks

In an analogous problem, we have considered community-aware role mining in social networks [1, 2, 46], *i.e.*, defining and discovering social roles of individuals by taking into account the interactions between them, as well as their affiliations to the modules/communities. For instance in [2], we define four fundamental roles for an individual within a society: 1) leader (the most central node within its module), 2) outermost (the members with significantly low activity), 3) mediator (nodes which bridge multiple modules), and 4) outsider (individuals that are not affiliated to any one module). Our analysis of Enron email dataset shows how the changes in these structural roles, in combination with the changes in the modular structure of the network, provides additional clues into the dynamics of the network under study, *e.g.*, how a leader role change triggers modules to merge or split.

### 5.1.2 Analyzing Dynamics of Networks

Analyzing the dynamics of modules over time, can be useful in many applications such as targeted marketing and advertising. We have focused on different aspects of tracking the evolution of modules/communities in [148, 149]. In [148], we propose an approach to detect evolving communities over time, which mines communities incrementally by considering the previously discovered communities at each step. Based on the discovered meta communities which span over time, we then define and detect the critical events that characterize the evolution of communities, which are namely survive, dissolve, split, merge, and form. In [149], we further show how these evolution events of communities can be predicted based on the relevant structural and temporal properties. Moreover, we show how this analysis enables one to also identify the most prominent features in community transitions. For example, our analysis of the Enron dataset discovers that the average clustering coefficient and cohesion of modules/communities are prominent positive factors on their survival; while leaders discussing a stable topic, and the ratio of nodes leaving a community are important negative factors on the survival of that community. We further confirm that these prominent features depend on the underlying dynamic social network, *e.g.*, the ratio of nodes leaving a community is not a prominent feature when we apply the model on the DBLP dataset.



***(a)*** *Social network of students interacting in an online discussion forum. Nodes represent actors or students, while an edge from a student to the other summarizes messages sent in that direction and the thickness of that edge corresponds to the number of messages sent.*

***(b)*** *Communities detected in the social network of students interacted in the online discussion forum. Different colours represent the three different communities of students that communicated mostly within themselves throughout the course.*

***Figure 5.1:*** *Interpreting Students Interaction Network*

### 5.1.3 Educational Case Study

We track the evolution of interaction patterns and roles over time in an educational case study in [128, 129, 132]. These works focus on providing the course instructors with better means to assess the participation of students by analyzing the interactions of students in asynchronous discussion forums of online courses. Here, we give the instructor a quick view of what is discussed in these forums, what are the main topics discussed, how much each student has participated in these topics, and how the students collaborated on each discussion/topic. In more detail, from the discussion forums recorded in an e-learning environment, we extract both the interaction network of students (where edges correspond to exchange of written messages), and the co-occurrence network of terms used in their discussions (where terms/nodes are connected if they co-occur in the same sentence). Then we detect the communities in these two networks[1]. The discovered communities respectively correspond to the collaboration groups of students (Figure 5.1), and the topics of their discussions (Figure 5.3).

Furthermore, we monitor the changes in collaborative groups of students and detect events and patterns by a dynamic analysis of the communities. Such events can affect participation and engagement of students in the course, and detecting these events could be used to make proper recommendation to modify students behaviours affected by these changes. For instance a community split is detected in Figure 5.4a, and it can be predicted that the participation level of the detached students would drop, accordingly, hence they could be recommended and/or invited by one of the remaining collaboration groups to join and engage in their discussions. As another example, a

[1] Here, we used the FastModularity approach which is known to perform well in different domains (see Section 2.2.1); however any other community detection method could be used instead to detect the groupings or communities.

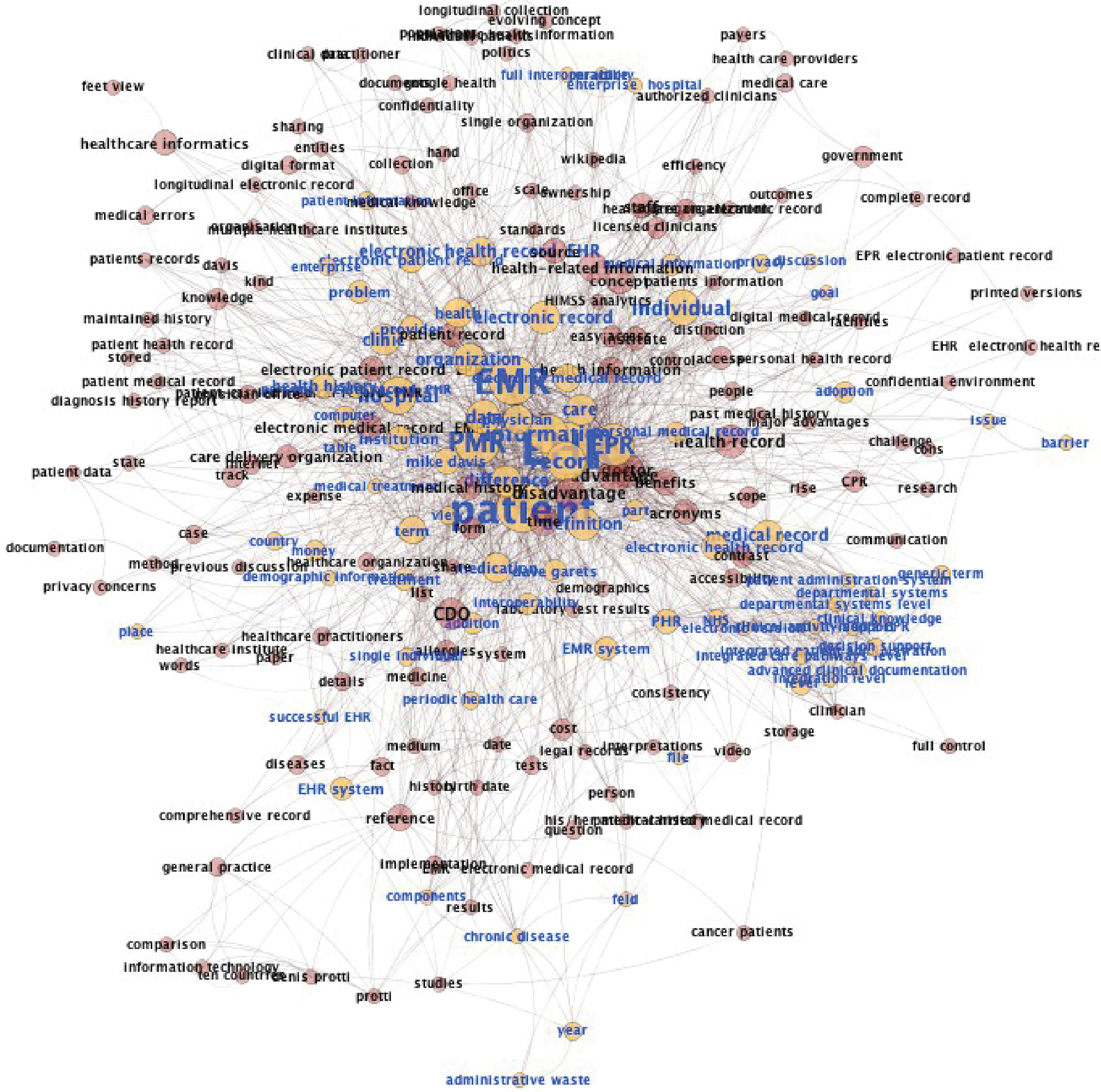


***Figure 5.2:*** *Examining words used by a particular student in a discussion thread in an e-learning course. This can be used to determine and compare interest of different students as well as their level of participation. Here, terms used by a selected student are highlighted.*

community growth can also be detected proceeding the community split of Figure 5.4a, illustrated in Figure 5.4b. Where the Red community recruits new members while at the same time the Green community dissolves into Purple. We can also see the effect of the leader move between communities that clearly has triggered most of these events. In Figure 5.4a, moving Eric from the Cyan community to Purple community caused the Cyan to split, while in Figure 5.4b, his next move

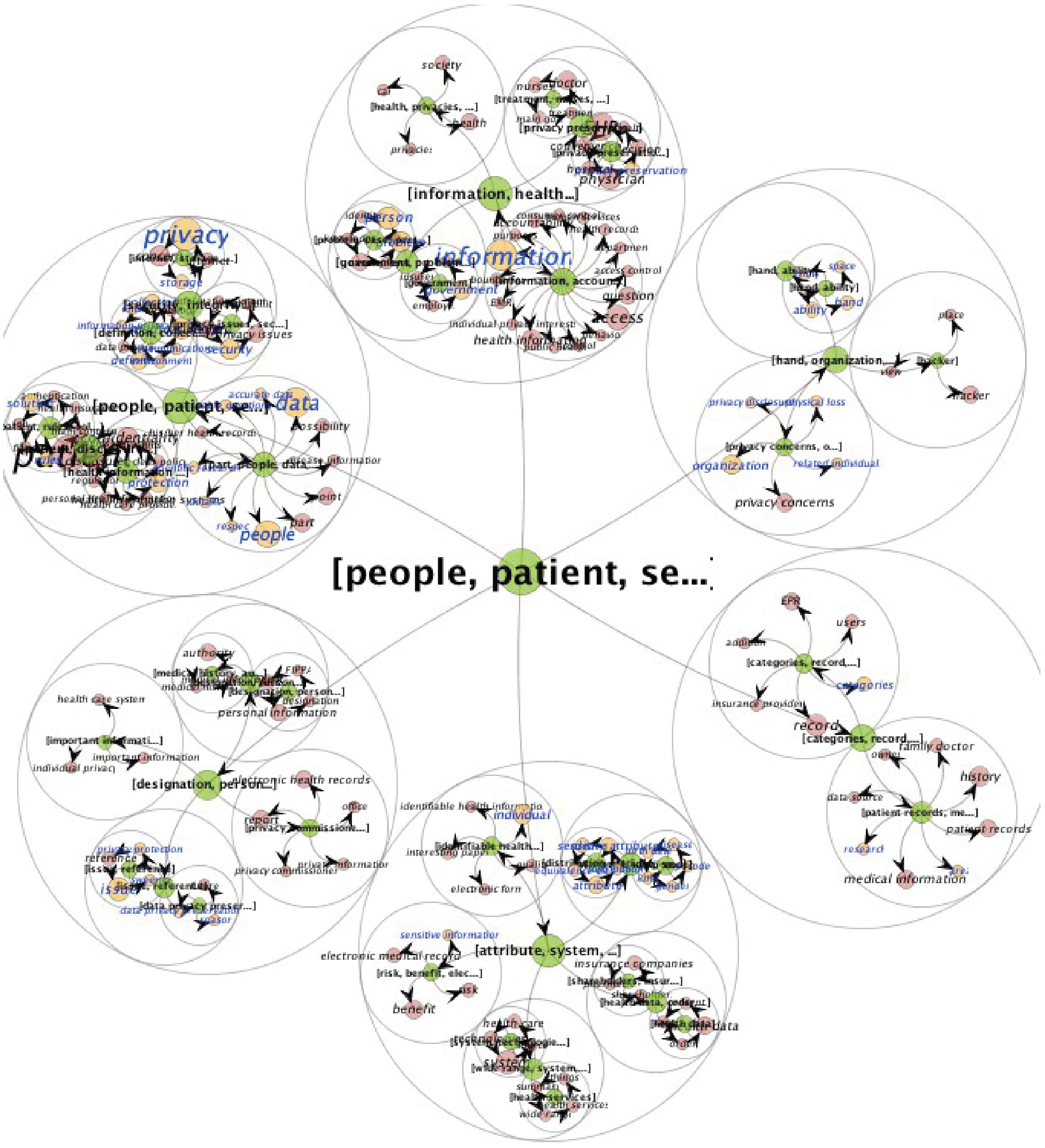
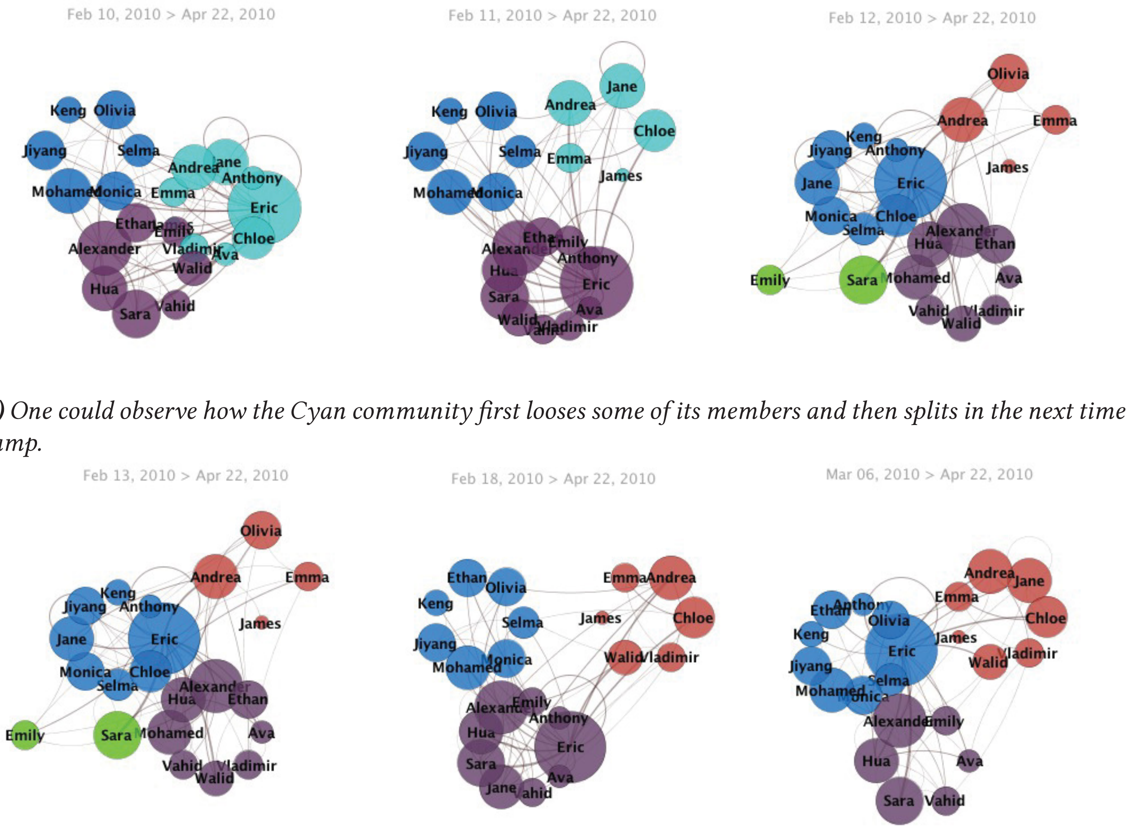


***Figure 5.3:*** *Hierarchical topic clustering of a discussion thread. In which one can also examine topics that a particular student was involved with, the content of these topics and the range of participation of the student.*

from Purple to Blue, helped the Red community to enlist some of the Purple members. Here, we used a simple matching between communities of different timestamps to find different instances of the same community through time, and the community events are detected manualy and in an exploratory manner. A more sophisticated framework for dynamic analysis of communities and automatic event detection is later described in [148, 149].


Feb 10, 2010 > Apr 22, 2010
Feb 11, 2010 > Apr 22, 2010
Feb 12, 2010 > Apr 22, 2010
Feb 13, 2010 > Apr 22, 2010
Feb 18, 2010 > Apr 22, 2010
Mar 06, 2010 > Apr 22, 2010
Keng
Olivia
Jiyang
Selma
Mohamed
Monica
Emma
Andrea
Jane
Anthony
Eric
Chloe
Ethan
Emily
Alexander
Vladimir
Ava
Walid
Hua
Sara
Vahid
James

*(a) One could observe how the Cyan community first looses some of its members and then splits in the next time stamp.*

*(b) the Green community follows leader into the Purple community; Purple members leave the community after the leader moves to the Blue community.*

***Figure 5.4:*** *Changes in the collaborative groups over time, and the effect of the leader move in these groups.*

In the following, we focus on the correlation between the characteristics of individual nodes and their community affiliations. We propose community guidance by attributes, which finds a modular structure that aligns well with a given attribute. Using this approach [125], different high quality community perspectives can be discovered where each best correspond with the selected set of attributes, and could then be utilized to predict the missing values of those attributes.

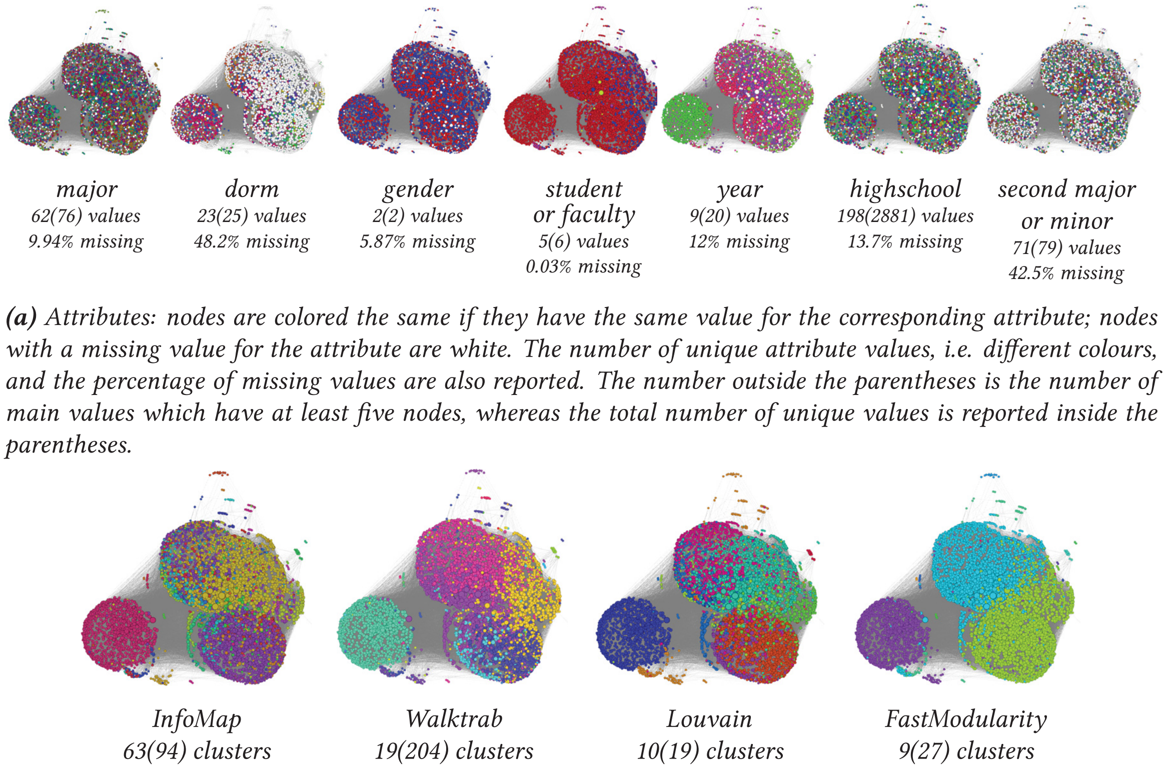
## 5.2 Modules and Attributes

Many real world applications include information on both attributes of individual nodes as well as relations between the nodes, while there exists an interplay between these attributes and relations [33, 75, 89]. More precisely, the relations between nodes motivates them to develop similar attributes (influence), whereas the similarities between them motivates them to form relations (selection), a property referred to as homophily. This homophily results in an observed correlation between the modular structure of networks and attributes of nodes, *i.e.*, self-identified/explicit user characteristics [152]; which has motivated defining ground-truth communities for real networks based on these explicit properties of nodes.

In more detail, alternative to generating benchmarks for the community detection task, large real world benchmarks are often used where the ground-truth communities are defined based on some explicit properties of the nodes such as user memberships in social network. Notably, Yang and Leskovec [166] adapt this approach to compare different community detection algorithms based on their performance on large real world benchmarks; where characteristics such as social groups are considered “reliable and robust notion of ground-truth communities”. For example, in a collaboration network of authors obtained from DBLP, venues are considered as the ground-truth communities, or in the Amazon product co-purchasing network, product categories are considered as the ground-truth. A similar analysis is performed by Yang et al. [170], including a comparison between the result on large real social networks and the LFR benchmarks, arguing that the former is better indicator of the performance of the algorithms. However, this ground-truth data is imperfect and incomplete and should be rather considered as metadata or labeled attributes correlated with the underlying communities, as also mentioned by Lee and Cunningham [83].

In the presence of attributes, a more plausible viewpoint is finding groups of nodes that are both internally well connected and having homogeneous attributes. This grouping is referred to as structural attribute clustering by Zhou et al. [174] or cohesive patterns mining by Moser et al. [99]. In [99], for example, a cohesive pattern is defined as a subset of nodes which satisfy both a density constraint and a subspace cohesion constraint; then the maximal cohesive patterns are discovered by pruning the search space based on these constraints. Similar to the community mining problem, several alternative approaches are proposed for integrating attributes and relationships in finding homogeneous modules in networks, examples are Cruz and Bothorel [34], Günnemann et al. [58], Hanisch et al. [61], Hu et al. [63], Mislove et al. [98], Yang et al. [169].

Here, we first investigate the correlations between attributes and community structure using our network specific agreement/external indexes proposed in Chapter 3. Then we present the concept of *community guidance by attributes*, where we adapt our previously proposed TopLeaders[127] community detection method, to find the right number of communities in the given network, based on the available attributes information.



***(a)** Attributes: nodes are colored the same if they have the same value for the corresponding attribute; nodes with a missing value for the attribute are white. The number of unique attribute values, i.e. different colours, and the percentage of missing values are also reported. The number outside the parentheses is the number of main values which have at least five nodes, whereas the total number of unique values is reported inside the parentheses.*

***(b)** Communities: nodes are colored the same if they belong to the same community in the results of corresponding community mining algorithms. The number of clusters, i.e. colours, with at least five members is reported, whereas the total number of clusters in the result is given inside the parentheses.*

***Figure 5.5:** Visualization of correlations between attributes and communities for the American75 dataset from* Facebook 100 dataset*[153]. This network has 6386 nodes and 217662 edges (friendships which are unweighted, undirected). Visualization is done with Gephi, and an automatic layout is used which positions nodes only based on their connections.*

### 5.2.1 Correlation of Communities and Attributes

Traud et al. [152] show that a set of node attributes can act as the primary organizing principle of the communities; e.g. House affiliation in their study of Facebook friendship network of five US universities. In computing the correlation between attributes and relations, Traud et al. [152] use the basic clustering agreement indices for communities comparison. They observe that the correlation significantly depends on this agreement index and differs significantly even between those indices that have been known to be linear transformation of each other. Here we perform similar experiments, but in the context of evaluating community mining algorithms. In more details, we compare the agreements of the results from four different community mining algorithms, with each attribute in the dataset; see Figure 5.5 for a visualized example. First, the community mining algorithms are applied on the dataset, which are InfoMap [140], WalkTrap [120], Louvain [16], and FastModularity [104]. Then the correlations between the resulted communities from these

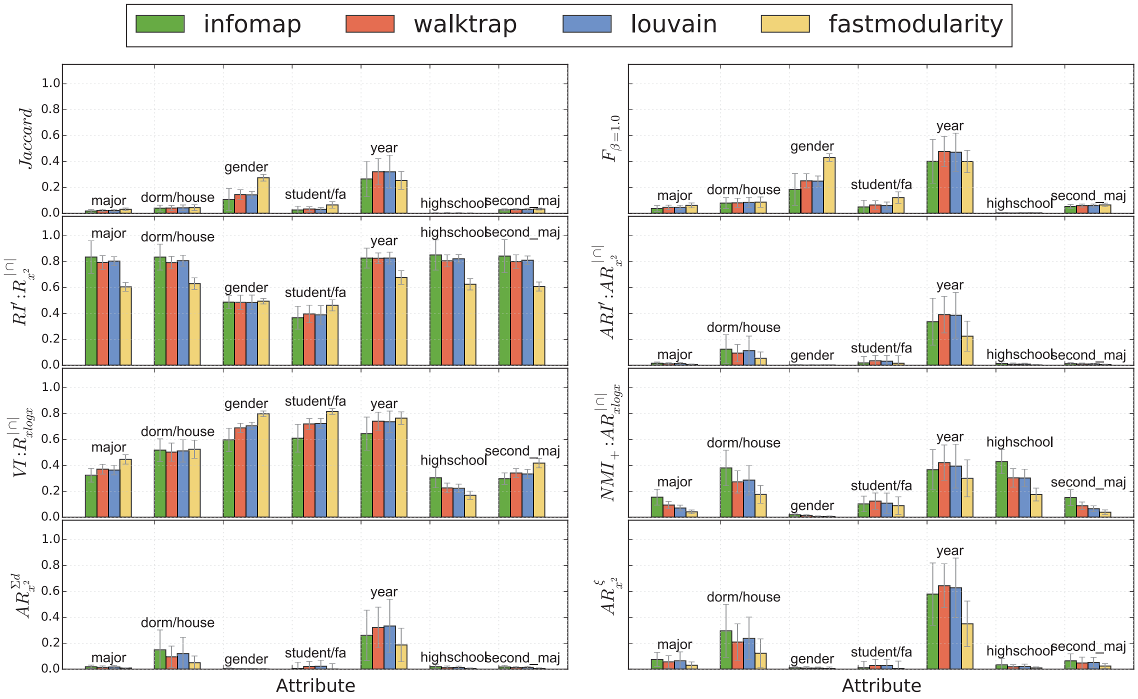


***Figure 5.6:*** *The agreement of different community detection algorithms with each attribute, averaged over 100 friendship networks of US colleges in* Facebook 100 dataset*. Each subplot shows the agreements when measured using the corresponding index, for instance the first subplot shows the agreements when using Jaccard index.*

algorithms and the attributes are measured using clustering agreement indices. More specifically, we measure the agreement assuming the unique attribute values are grouped together and formed a clustering. For example, for the attribute ‘year’, all nodes that have the value ‘2008’ are in the same group or cluster. Figure 5.6 shows the agreements of the community mining algorithms with each attribute averaged over all the networks in the Facebook 100 dataset[153]; which shows a snapshot of facebook friendships at 100 US universities. The agreements, between two groupings/clusterings of the dataset, are measured with eight different agreement indices: Jaccard Index, F-measure, Variation of Information(VI), Normalized Mutual Information (NMI), Rand Index (RI), Adjusted Rand Index (ARI), and two structure based extensions of ARI tailored for comparing network clusters: with overlap function as the sum of weighted degrees ($\mathcal{ARI}^{\Sigma d}_{x^2}$), and the number of common edges ($\mathcal{ARI}^{\xi}_{x^2}$); defined in Chapter 3.

Unlike the previous study, we observe very similar rankings with different agreement indexes. The most agreements are observed with the attribute ‘year’, followed not so closely by ‘dormitory’. We can however see that the ranking across different attributes is not the same, whereas Walktrap is the winner according to the ‘year’, and Infomap performs the best if we consider the agreement with the ‘dormitory’. Therefore, although we observe a correlation between the attributes and the communities, it is not wise to compare the general performance of community mining algorithms based on their agreements with a selected attribute as the ground-truth. Instead one should treat attributes as another source of information correlated with the structure of the network. In the next section, for example, we use the attribute information to fine tune the parameters of a community mining algorithm, so that it results in a community structure which complies most with our selected attribute. Before that, we present a discussion on the effect of missing values on the agreement indices.

### 5.2.2 Missing Values and Agreement Indices

The definitions of original agreement indices assume the two clusterings are covering the same set of datapoints. Therefore to use these indices, nodes with missing values should be either removed, grouped all as a single cluster, or each treated as a singleton cluster. The implementations we use here are based on our generalized formula proposed in Chapter 3. Unlike the original definitions, these formulae do not require the assumption that the clusterings cover the whole dataset. Hence they can be directly applied to the cases where we have un-clustered datapoints, which will be ignored. For the sake of comparison, in Figure 5.7 three bars are plotted per <attribute, community mining> pair, corresponding to how the missing values can be handled: (i) when nodes with missing values are removed from both groupings before computing the agreement, (ii) when all the nodes with missing attribute value are grouped into a single cluster, and (iii) when computing the agreements with lifting the covering assumption, using the formulations of Definition 3.3.1. This comparison is in particular important here, since we have many nodes with *missing values* for some of the attributes, such as ‘dormitory’ or ‘second major’; and the way missing values are handled significantly affects the agreements measured, as seen in the Figure 5.7. In the following experiments, we use the Definition 3.3.1, which does not treat the unclustered nodes, *i.e.*, missing values, as disagreement.

## 5.3 Guiding Community Detection by Attributes

Zhou et al. [174] propose clustering an attribute augmented network. The augmented network includes attribute nodes for each <attribute, value> and edges are added between original graph nodes to their corresponding <attribute, value> nodes, this graph representation has also been used in link recommendation, *e.g.*, see Gong et al. [53]. The authors show that a straightforward distance function based on a linear combination of the structural and attribute similarities, fails to outperform a similar method that only considers structural or attribute similarities. In Mislove et al. [98], communities are found using a link based approach but are initialized using a clustering based on their attribute similarities. As another example in Cruz and Bothorel [34], communities found by links are further divided into smaller sub-groups according to the attributes. In more detail, the overlap of each community is computed with each cluster in the clustering of the same data according to the attributes(rephrase). Then larger than average overlaps are cut from the main community to form smaller, more cohesive communities. All these works we have discussed

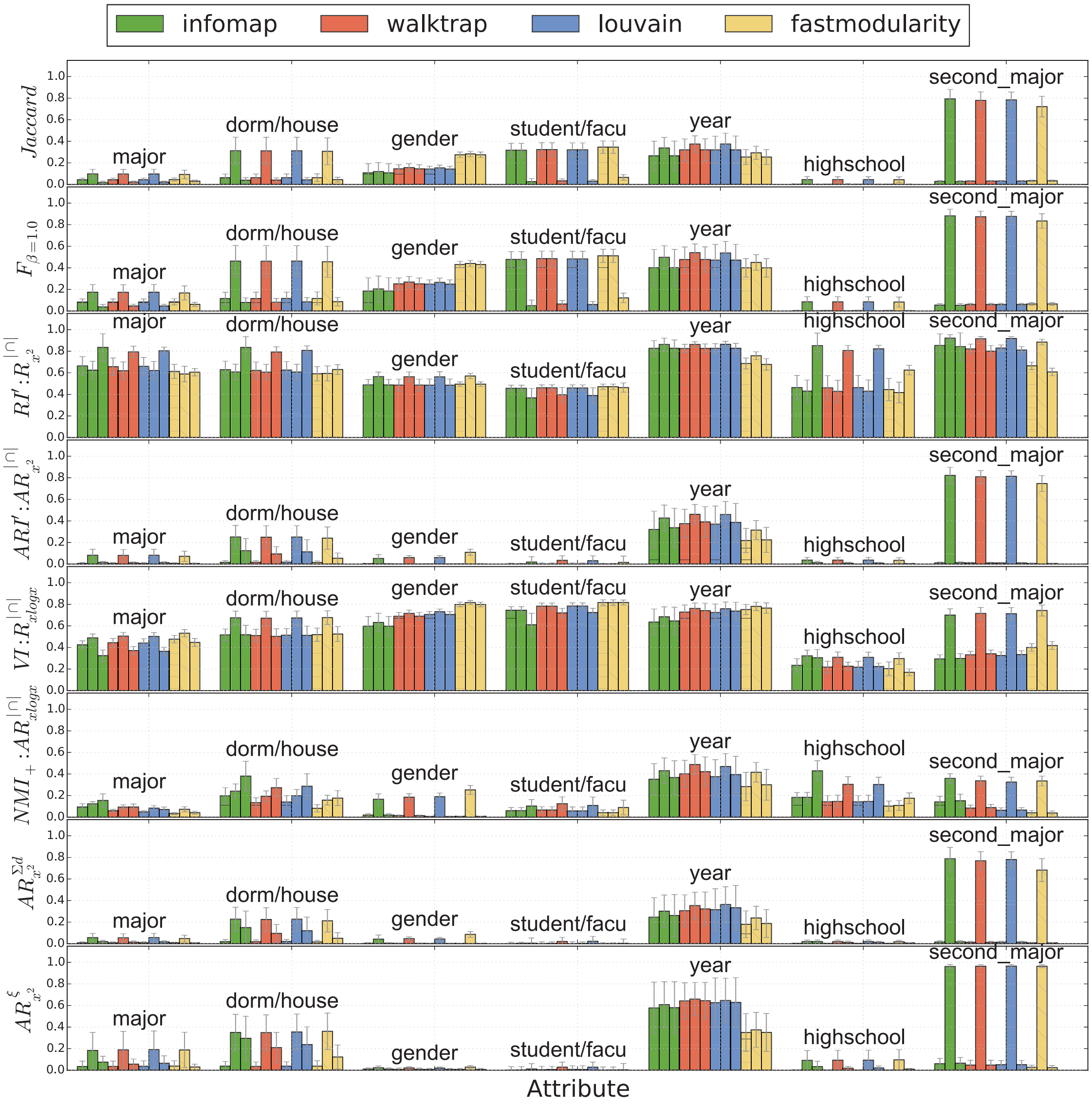


***Figure 5.7:*** *The effect of missing values: bars with horizontal, diagonal, and solid fill correspond respectively to removing missing values, adding missing values as a single cluster, or lifting the covering assumption.*

so far further motivate combining attribute and link data, rather than validating one based on the other. Here, we propose the concept of *community guidance by attributes*, where a given attribute (based on the application) is used to direct a community mining algorithm. More specifically, we guide our TopLeaders [124, 127] algorithm to find the right number of communities, based on the agreements of its result with the given attribute. The concept is however general and can be applied to fine tune parameters of any community mining algorithm. Which is true for algorithms which are capable of providing different community structure perspectives, based on different values for the algorithm parameters.

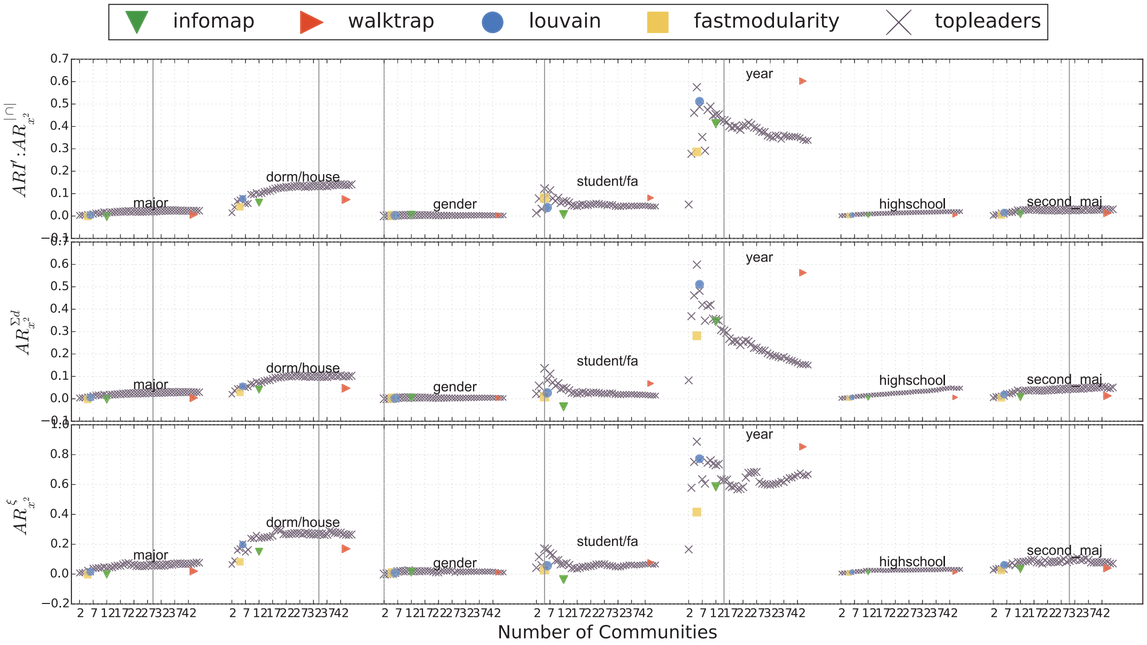


***Figure 5.8:*** *Agreement of attributes with the results of algorithms plotted as a function of number of communities.*

The number of communities, $k$ for short, is the main parameter for the TopLeaders algorithm, similar to the k-means algorithm for data clustering. Figure 5.8 illustrates an example on the Amherst41 dataset, where the agreements of each attribute with the results of Topleaders are plotted as a function of $k$. For some of the attributes, such as 'student/faculty', we observe a clear peak around the true number of classes. We also plotted where other algorithms land. However, there has not been any parameter tuning for those algorithm, and hence they are indicated with a single point. The vertical lines show the true number of classes for the corresponding attribute, *i.e.*, the distinct values; except for the attribute 'highschool', for which the true $k$ is 1075 and lies outside of the plot's scale.

Consequently, between the communities detected by the TopLeaders for different values of $k$, which only uses the links to discover communities, we select the one that has the most agreement with the given attribute. We used an exhaustive search to find the optimal $k$ for each attribute, in the range of $[2, \sqrt{n}]$, where $n$ is the total number of datapoints. Figure 5.9 shows the agreements obtained through this approach, compared to the four commonly used community detection algorithms. We can see in Figure 5.9 that the communities found by this approach have comparable, and in some cases better agreements with the attributes, compared to the methods which do not consider that extra information. This is more significant according to the structure based agreement measures, especially $\mathcal{ARI}^{\xi}_{x^2}$, which considers common edges as the cluster overlaps; and also for less trivial attributes which have a low agreement with the trivial communities, *e.g.*, 'stu-

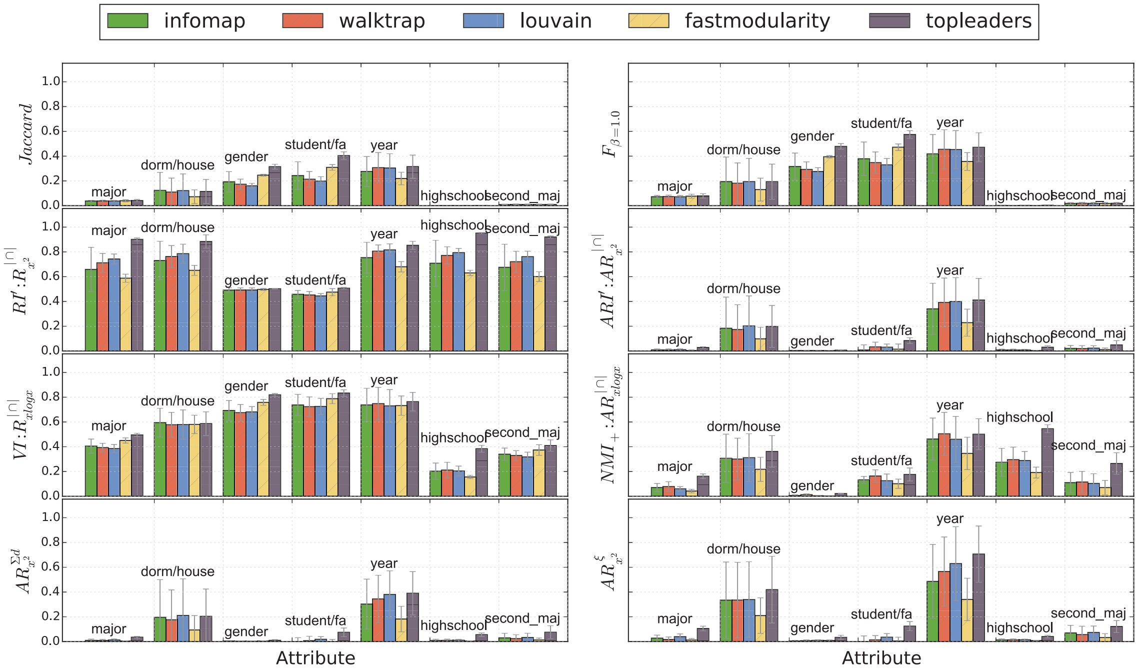


***Figure 5.9:*** *TopLeaders performance when the number of communities are chosen according to the agreement of its results with the given attribute. This result is averaged over a subset of 5 datasets from the 100 Facebook networks, which are: Amherst41, Bowdoin47, Caltech36, Hamilton46, and Haverford76.*

dent/faculty', 'second_major', or 'highschool'. One should however note that this is not a comparison for the performance of these algorithms, since TopLeaders used the agreements with the attribute to find the $k$, which is not available to the other methods.

## 5.4 Conclusions

In this chapter, we discussed utilization of the modular structure of networks in different contexts, including role mining in social and biological networks, as well as analyzing the dynamics of networks and tracking their evolution patterns. In particular, we focused on an application focused case example, which discovers modules to recover collaboration groups of students in an e-learning setting, and also to outline the topics of their discussions.

Further, we investigated the evaluation of communities on real-world networks with attributes, where there exist a correlation between the characteristics of individual nodes and their connections. We then proposed the concept of community guidance by attributes, where a community mining algorithm is guided to find a community structure which corresponds most to a given attribute. This is in particular useful in real world applications, since we often have access to both link and attribute information, and an idea of how communities will be used. For example, communities in protein-protein interaction networks are shown to be correlated with the functional categories of their members, which are used to predict the previously uncharacterized protein complexes [147]; in such case, one might be interested to select the community structure that corresponds most with the available functional categories.

# Chapter 6

# Conclusion

This thesis studied the modular structure of real-world complex networks. To summarize:

- Chapter 2 introduced network adaptations of the well-established clustering validity criteria, that quantify the goodness of a single clustering. The adapted network criteria provide a more extensive set of validity measures for the evaluation of community mining algorithms, *i.e.*, algorithms that detect the modular structure of networks.

- Chapter 3 presented generalizations of the well-known clustering agreement measures, that compare two clusterings of the same network. From these generalizations, extensions are derived for comparing overlapping and network clusters. These extensions are tailored for measuring the (dis)agreement of clusterings/communities, that represent the modular structure in networks.

- Chapter 4 examined the generative network models, generalized a common model used for synthesizing modular networks, and introduced an intuitive and flexible alternative model which more closely complies with the characteristics observed for real-world networks. The proposed alternative generator has a high degree of expressiveness and is particularly useful for generating benchmark datasets with built-in modular structure, that are used in the evaluation of community detection algorithms.

- Chapter 5, investigated how the modular structure of networks can be utilized in different contexts. On one hand, it focused on an e-learning application and illustrated how the network modules/communities can effectively outline the collaboration groups of students, as well as the topics of their discussions; and how this could be used to monitor participation trends of students throughout an online course. On the other hand, it showed the interplay between the attributes of nodes and their memberships in communities, and presented how this interplay can be leveraged for derivation of alternative modular structures that better align with a given subset of attribute(s).

## 6.1 Recommendations for Evaluation of Communities

An important research direction is to evaluate and compare the results of different community mining algorithms. An intuitive practice is to validate the results partly by a human expert [91]. However, the community mining problem is NP-hard; the human expert validation is limited, and is based on narrow intuition rather than on an exhaustive examination of the relations in the given network, specially for large real networks.

There is a congruence relation between defining communities and evaluating community mining results. In fact, the well-known modularity Q by Newman and Girvan [108] which is commonly applied as an objective function for community detection, was originally proposed for quantifying the goodness of the community structure, and is still commonly used for evaluating the algorithms [28, 139]. Considering that the only commonly used criterion is the modularity Q , in Chapter 2, we presented an extensive set of general objectives for evaluation of network clustering algorithms, mostly adapted from clustering background such as Variance Ratio Criterion, Silhouette Width Criterion, Dunn index, *etc.* One should however note that this type of evaluation is based on an assumption about what are the good communities, and hence is not appropriate for validating results of algorithms that are built upon different assumptions. In fact, choosing an evaluation criterion encompasses the same non-triviality as of the community mining task itself. Our experiments also revealed that the rankings of the adopted criteria depend on the experiment settings, and there is no winner criteria that could be used in all settings.

A common alternative evaluation practice is validating the algorithms on benchmark datasets by measuring the agreement between their resulted communities and the ground-truth structure available in these benchmarks. This agreement is measured using a clustering agreement index, which measures the similarity between two given clusterings, usually based on the pairwise overlaps of their clusters. The traditional clustering agreement measures only consider memberships of data-points in clusters, and overlook any relations between the data-points, which makes them inefficient in comparing network clusters. Hence we recommend using the structure based measures introduced in Chapter 3 when comparing community structures of networks.

Since there are few and typically small real world benchmarks with known communities available, this evaluation is usually performed on synthetic benchmarks. We are assuming that the performance of an algorithm on the benchmarks datasets, is a predictor of its performance on real networks. For this assumption to hold, we need realistic benchmark generators, with tunable parameters for different domains; since it has been shown that the characteristics of clusters in networks are remarkably similar between networks from the same domain [80, 113]. However, the current common generators used for synthesizing benchmarks are domain-independent and also overlook basic characteristics of the real networks, as discussed in Chapter 4. We have seen in Chapter 2 and Chapter 4, that the ranking of algorithms significantly changes based on the benchmark datasets used, and hence it is important to compare basic characteristics of benchmarks used for the evaluation with the target real-world network, to ensure the applicability of the community detection method. It is also important to improve the realistic degree of synthetic benchmark generators. With this regard, we have presented FARZ benchmarks which follow the characteristics of real networks more closely and hence are more recommended for evaluation of community mining algorithms.

Alternative to generating benchmarks for the community detection task, large real world benchmarks are often used where the ground-truth communities are defined based on some explicit properties of the nodes such as user memberships in social network. In Chapter 5, we discuss that this ground-truth data is incomplete and should be rather considered as metadata or labeled attributes correlated with the underlying communities.

## 6.2 Future Work

In our experiments of Chapter 2, several of the adapted criteria exhibit high performances on ranking different network clusterings of a given dataset, which makes them useful alternatives for the modularity Q ; particularly, the Z-Index criterion. Hence, it is interesting to apply the best performing criteria as objectives for finding communities and develop new community detetion algorithms. Another line of work following this chapter is to provide extensions of the criteria for more general cases of community mining, mainly overlapping communities, dynamic communities and also local communities.

In Chapter 3, we presented new agreement measures for comparing two clusterings in networks, mainly focusing on the cluster validation perspective. Alternative context in which these measure could be applied can be explored as future research, particularly applying them in ensemble clustering and multi-view clustering.

In Chapter 4, we presented the FARZ model for synthesizing realistic complex networks with built-in community structure. This model could be extended in several ways, particularly by incorporating attributes for nodes, as well as the temporal information required for evaluating dynamic community detection methods.

In Chapter 5, we proposed a semi-supervised network clustering algorithm, a.k.a. community detection method, which utilizes available attribute information to refine candid community structures. Since the resulted communities are tuned toward the selected attribute(s), a natural extension of this work is to use this discovered structure to infer the missing attribute values, and compare the performance of this module-based prediction model with the current methods which mainly rely on the fine grained structure of the networks. A second extension of this work is using a more efficient optimization approach for maximizing the agreement of the modular structure with an attribute based clustering, instead of the exhaustive search used.

# References

[1] A. Abnar, M. Takaffoli, R. Rabbany, and O. R. Zaïane. Ssrm: Structural social role mining for dynamic social networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 289–296, Aug 2014. 84

[2] A. Abnar, M. Takaffoli, R. Rabbany, and O. R. Zaïane. Ssrm: structural social role mining for dynamic social networks. *Social Network Analysis and Mining*, 5(1):1–18, 2015. 84

[3] C. C. Aggarwal and C. K. Reddy. *Data Clustering: Algorithms and Applications*. CRC Press, 2014. 31

[4] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010. 11

[5] A. N. Albatineh, M. Niewiadomska-Bugaj, and D. Mihalko. On similarity indices and correction for chance agreement. *Journal of Classification*, 23:301–313, 2006. 10.1007/s00357-006-0017-z. 22, 34, 35

[6] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002. 10, 61

[7] L. Ana and A. K. Jain. Robust data clustering. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–128. IEEE, 2003. 31

[8] D. T. Anderson, J. C. Bezdek, M. Popescu, and J. M. Keller. Comparing Fuzzy, Probabilistic, and Possibilistic Partitions. *IEEE Transactions on Fuzzy Systems*, 18(5):906–918, Oct. 2010. 41

[9] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman Divergences. *The Journal of Machine Learning Research*, 6:1705–1749, Dec. 2005. 39

[10] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. 61, 65

[11] F. Bauer. Normalized graph Laplacians for directed graphs. *ArXiv e-prints*, July 2011. 145

[12] A. R. Benson, C. Riquelme, and S. Schmit. Learning multifractal structure in large networks. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1326–1335, 2014. 61

[13] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981. 29

[14] G. Bianconi, R. K. Darst, J. Iacovacci, and S. Fortunato. Triadic closure as a basic generating mechanism of communities in complex networks. *Physical Review E*, 90(4), 2014. 70

[15] C. Biemann. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80. Association for Computational Linguistics, 2006. 10, 11

[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(10), 2008. 2, 8, 9, 51, 77, 90

[17] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20(2):172–188, 2008. 8, 12

[18] R. K. Brouwer. Extending the rand, adjusted rand and jaccard indices to fuzzy partitions. *Journal of Intelligent Information Systems*, 32(3):213–235, Apr. 2008. 41

[19] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3:1–27, 1974. 18

[20] R. Campello. Generalized external indexes for comparing data partitions with overlapping categories. *Pattern Recognition Letters*, 31(9):966–975, 2010. 29, 41

[21] R. Campello and E. R. Hruschka. A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems*, 157(21):2858–2875, 2006. 29

[22] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys*, 38, 2006. 60

[23] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, and S. Bhowmick. On the permanence of vertices in network communities. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1396–1405, 2014. 64

[24] J. Chen, O. Zaiane, and R. Goebel. An unsupervised approach to cluster web search results based on word sense communities. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 725–729, Dec 2008.

[25] J. Chen, O. R. Zaïane, and R. Goebel. Detecting communities in large networks by iterative local expansion. In *International Conference on Computational Aspects of Social Networks*, pages 105–112, 2009. 11

[26] J. Chen, O. R. Zaïane, and R. Goebel. Detecting communities in social networks using max-min modularity. In *SIAM International Conference on Data Mining*, pages 978–989, 2009. 8

[27] F. Chung. Laplacians and the cheeger inequality for directed graphs. *Annals of Combinatorics*, 9(1): 1–19, 2005. 145

[28] A. Clauset. Finding local community structure in networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 72(2), 2005. 2, 3, 13, 98

[29] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6), 2004. 9, 62

[30] L. M. Collins and C. W. Dent. Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. *Multivariate Behavioral Research*, 23(2):231–242, 1988. 29, 41, 45

[31] M. Cortina-Borja. Handbook of parametric and nonparametric statistical procedures. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 175(3):829–829, 2012. 36

[32] M. Coscia, F. Giannotti, and D. Pedreschi. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, 4(5):512–546, 2011. 11

[33] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 160–168, 2008. 5, 60, 89

[34] J. Cruz and C. Bothorel. Information integration for detecting communities in attributed graphs. In *Computational Aspects of Social Networks (CASoN), 2013 Fifth International Conference on*, pages 62–67, Aug 2013. 89, 92

[35] Y. Cui, X. Fern, and J. Dy. Non-redundant multi-view clustering via orthogonalization. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 133–142, Oct 2007. 31

[36] E. C. Dalrymple-Alford. Measurement of clustering in free recall. *Psychological Bulletin*, 74:32–34, 1970. 19

[37] T. Dang. *Analysis of communities in social networks.* PhD thesis, Ph. D. thesis, Université Paris 13, 2012. 63

[38] L. Danon, A. DÃŋaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005. 3, 22, 32, 59

[39] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. 13, 18

[40] I. S. Dhillon and J. A. Tropp. Matrix nearness problems with bregman divergences. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1120–1146, 2007. 45

[41] L. Duan, W. N. Street, Y. Liu, and H. Lu. Community detection in graphs through correlation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1376–1385, 2014. 64

[42] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical review E*, 72(2), 2005. 10

[43] D. Dumitrescu, B. Lazzerini, and L. C. Jain. Fuzzy sets and their application to clustering and training. *CRC Press*, 2000. 29

[44] J. C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974. 13, 18

[45] P. Erdős and A. Rényi. On the evolution of random graphs. In *Publication of the Mathematical Institue of the Hungarian Academy of Sciences*, pages 17–61, 1960. 1, 10, 60, 62

[46] J. Fagnan, R. Rabbany, M. Takaffoli, E. Verbeek, and O. R. Zaïane. Community dynamics: Event and role analysis in social network analysis. In *Advanced Data Mining and Applications*, pages 85–97. Springer, 2014. 84

[47] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010. 2, 11, 12, 15, 83

[48] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007. 9

[49] S. Fortunato and C. Castellano. Community structure in graphs. In *Computational Complexity*, pages 490–512. Springer, 2012. 11

[50] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983. 34

[51] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002. 8, 12, 21, 59, 60, 62

[52] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, Feb. 2010. 60

[53] N. Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, D. Song, et al. Jointly predicting links and inferring attributes using a social-attribute network (san). *arXiv preprint arXiv:1112.3265*, 2011. 92

[54] N. Z. Gong, W. Xu, L. Huang, P. Mittal, E. Stefanov, V. Sekar, and D. Song. Evolution of social-attribute networks: Measurements, modeling, and implications using google. In *Proceedings of the ACM Conference on Internet Measurement Conference*, pages 131–144, 2012. 63

[55] S. Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10), 2010. 11, 32, 51, 77

[56] R. Guimera and L. A. N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005. 3, 83

[57] R. Guimera, M. Sales-Pardo, and L. A. N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2), 2004. 10

[58] S. Günnemann, B. Boden, I. Färber, and T. Seidl. Efficient mining of combined subspace and subgraph clusters in graphs with feature vectors. In *Advances in Knowledge Discovery and Data Mining*, pages 261–275. Springer, 2013. 89

[59] M. Gustafsson, M. Hörnquist, and A. Lombardi. Comparison and validation of community structures in complex networks. *Physica A: Statistical Mechanics and its Applications*, 367:559–576, 2006. 12, 32, 59

[60] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17:107–145, 2001. 8, 29

[61] D. Hanisch, A. Zien, R. Zimmer, and T. Lengauer. Co-clustering of biological networks and gene expression data. *Bioinformatics*, 18(suppl 1):S145–S154, 2002. 89

[62] F. HpÌšpner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy cluster analysis: methods for classification, data analysis and image recognition.* J. Wiley, 1999. 29

[63] B. Hu, Z. Song, and M. Ester. User features and social networks for topic modeling in online social media. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 202–209, 2012. 89

[64] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985. 22, 31, 33, 34, 35

[65] L. J. Hubert and J. R. Levin. A general statistical framework for assessing categorical clustering in free recall. *Psychological Bulletin*, 83:1072–1080, 1976. 20

[66] E. Hullermeier, M. Rifqi, S. Henzgen, and R. Senge. Comparing Fuzzy Partitions: A Generalization of the Rand Index and Related Measures. *IEEE Transactions on Fuzzy Systems*, 20(3):546–556, June 2012. 41

[67] L. G. S. Jeub, P. Balachandran, M. A. Porter, P. J. Mucha, and M. W. Mahoney. Think locally, act locally: Detection of small, medium-sized, and large communities in large networks. *Physical Review E*, 91, Jan 2015. 60

[68] E. C. Kenley and Y.-R. Cho. Entropy-based graph clustering: Application to biological and social networks. In *IEEE International Conference on Data Mining*, 2011. 11

[69] M. Kim and J. Leskovec. Multiplicative attribute graph model of real-world networks. In *Algorithms and Models for the Web-Graph*, pages 62–73. Springer, 2010. 63

[70] K. Kloster and D. F. Gleich. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1386–1395, 2014. 60

[71] I. M. Kloumann and J. M. Kleinberg. Community membership identification from small seed sets. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1366–1375, 2014. 60

[72] T. G. Kolda, A. Pinar, T. Plantenga, and C. Seshadhri. A scalable generative graph model with community structure. *SIAM Journal on Scientific Computing*, 36(5):C424–C452, 2014. 63

[73] V. Krebs. Books about us politics. http://www.orgnet.com/, 2004. 22

[74] B. Kulis, M. A. Sustik, and I. S. Dhillon. Low-rank kernel learning with bregman matrix divergences. *The Journal of Machine Learning Research*, 10:341–376, 2009. 45

[75] T. La Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. In *Proceedings of the 19th International Conference on World Wide Web*, pages 601–610, 2010. 5, 60, 89

[76] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5), 2009. 3, 12, 32, 51, 59, 79, 154

[77] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 80(1), 2009. 62

[78] A. Lancichinetti, S. Fortunato, and J. Kertesz. Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics*, 11(3):20, 2008. 32, 41, 48, 54, 148

[79] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4), 2008. 12, 22, 51, 59, 60, 62, 68, 148

[80] A. Lancichinetti, M. Kivelä, J. Saramäki, and S. Fortunato. Characterizing the community structure of complex networks. *PloS one*, 5(8), 2010. 98

[81] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PloS one*, 6(4):e18961, 2011. 11, 51, 55, 77

[82] C. Largeron, P.-N. Mougel, R. Rabbany, and O. R. Zaïane. Generating attributed networks with communities. *PloS one*, 10(4):e0122777, 2015. 59, 64

[83] C. Lee and P. Cunningham. Benchmarking community detection methods on social media data. *arXiv preprint arXiv:1302.0739*, 2013. 5, 89

[84] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 177–187, 2005. 1, 61, 65

[85] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1), 2007. 61, 65

[86] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 462–470, 2008. 61

[87] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *The Journal of Machine Learning Research*, 11:985–1042, 2010. 61

[88] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *International Conference on World Wide Web*, pages 631–640, 2010. 2, 11, 13, 32

[89] K. Lewis, M. Gonzalez, and J. Kaufman. Social selection and peer influence in an online social network. *Proceedings of the National Academy of Sciences*, 109(1):68–72, 2012. 89

[90] R. J. Light and B. H. Margolin. An analysis of variance for categorical data. *Journal of the American Statistical Association*, 66(335):534–544, 1971. 135

[91] F. Luo, J. Z. Wang, and E. Promislow. Exploring local community structures in large networks. *Web Intelligence and Agent Systems*, 6:387–400, 2008. 8, 11, 12, 98

[92] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008. 34

[93] J. Mcauley and J. Leskovec. Discovering social circles in ego networks. *ACM Trans. Knowl. Discov. Data*, 8(1):4:1–4:28, Feb. 2014. 32, 41

[94] A. McDaid and N. Hurley. Detecting highly overlapping communities with model-based overlapping seed expansion. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pages 112–119. IEEE, 2010. 11, 51, 77

[95] A. F. McDaid, D. Greene, and N. Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515*, 2011. 32, 41, 48, 54

[96] M. Meilă. Comparing clusteringsâĂŤan information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007. 31, 32, 36, 41

[97] G. Milligan and M. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985. 20

[98] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pages 251–260, New York, NY, USA, 2010. ACM. 89, 92

[99] F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining cohesive patterns from graphs with feature vectors. In *SDM*, volume 9, pages 593–604, 2009. 89

[100] S. Mussmann, J. Moore, J. J. P. III, and J. Neville. Incorporating assortativity and degree dependence into scalable network models. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015. 59

[101] M. Newman. *Networks: An Introduction.* Oxford University Press, Inc., 2010. 60, 71

[102] M. Newman. Community detection and graph partitioning. *arXiv preprint arXiv:1305.4974*, 2013. 10

[103] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003. 1, 62

[104] M. E. Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69 (6), 2004. 8, 51, 54, 77, 90

[105] M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3), 2006. 9, 10

[106] M. E. Newman. Spectral methods for community detection and graph partitioning. *Physical Review E*, 88(4):042822, 2013. 10

[107] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006. 2, 13, 16

[108] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), 2004. 3, 8, 9, 20, 60, 62, 98

[109] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99 Suppl 1:2566–2572, 2002. 60

[110] F. Nielsen and R. Nock. On the chi square and higher-order chi distances for approximating f-divergences. *Signal Processing Letters, IEEE*, 21(1):10–13, Jan 2014. 39

[111] W. d. Nooy, A. Mrvar, and V. Batagelj. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, 2004. ISBN 0521602629. 21

[112] J.-P. Onnela, D. J. Fenn, S. Reid, M. A. Porter, P. J. Mucha, M. D. Fricker, and N. S. Jones. Taxonomies of Networks. *ArXiv e-prints*, June 2010. 29

[113] J.-P. Onnela, S. Arbesman, M. C. González, A.-L. Barabási, and N. A. Christakis. Geographic constraints on social network groups. *PLoS one*, 6(4), 2011. 98

[114] G. K. Orman and V. Labatut. A comparison of community detection algorithms on artificial networks. In *Discovery Science*, pages 242–256. Springer, 2009. 62

[115] G. K. Orman, V. Labatut, and H. Cherifi. Qualitative comparison of community detection algorithms. In *International Conference on Digital Information and Communication Technology and Its Applications*, volume 167, pages 265–279, 2011. 13

[116] M. Pakhira and A. Dutta. Computing approximate value of the pbm index for counting number of clusters using genetic algorithm. In *International Conference on Recent Trends in Information Systems*, pages 241–245, 2011. 19

[117] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005. 8

[118] G. Palla, L. Lovász, and T. Vicsek. Multifractal network generator. *Proceedings of the National Academy of Sciences*, 107(17):7640–7645, 2010. 61

[119] K. Palla, D. Knowles, and Z. Ghahramani. An infinite latent attribute model for network data. *arXiv preprint arXiv:1206.6416*, 2012. 64

[120] P. Pons and M. Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences*, pages 284–293. Springer, 2005. 2, 51, 77, 90

[121] M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the AMS*, 56(9): 1082–1097, 2009. 11, 83

[122] R. Quere, H. Le Capitaine, N. Fraisseix, and C. Frelicot. On Normalizing Fuzzy Coincidence Matrices to Compare Fuzzy and/or Possibilistic Partitions with the Rand Index. In *2010 IEEE International Conference on Data Mining*, pages 977–982. IEEE, Dec. 2010. ISBN 978-1-4244-9131-5. 41

[123] R. Rabbany and O. Zaïane. Generalization of clustering agreements and distances for overlapping clusters and network communities. *Data Mining and Knowledge Discovery*, 29(5):1458–1485, 2015. 4, 30, 78, 79

[124] R. Rabbany and O. R. Zaïane. A diffusion of innovation-based closeness measure for network associations. In *IEEE International Conference on Data Mining Workshops*, pages 381–388, 2011. 17, 93

[125] R. Rabbany and O. R. Zaïane. Evaluation of community mining algorithms in the presence of attributes. In X.-L. Li, T. Cao, E.-P. Lim, Z.-H. Zhou, T.-B. Ho, and D. Cheung, editors, *Trends and Applications in Knowledge Discovery and Data Mining*, pages 152–163. Springer International Publishing, 2015. 83, 88

[126] R. Rabbany and O. R. Zaïane. Evaluation of community mining algorithms in the presence of attributes. In *Quality Issues, Measures of Interestingness and Evaluation of Data Mining Models Workshop, at the 19th Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, May 2015. 6, 83

[127] R. Rabbany, J. Chen, and O. R. Zaïane. Top leaders community detection approach in information networks. In *Proceedings of the 4th Workshop on Social Network Mining and Analysis*, 2010. 2, 10, 89, 93

[128] R. Rabbany, M. Takaffoli, and O. R. Zaïane. Analyzing participation of students in online courses using social network analysis techniques. In *Proceedings of Educational Data Mining*, pages 21–30, 2011. 83, 85

[129] R. Rabbany, M. Takaffoli, and O. R. Zaïane. Social network analysis and mining to support the assessment of on-line student participation. *ACM SIGKDD Explorations Newsletter*, 13(2):20–29, December 2011. 83, 85

[130] R. Rabbany, M. Takaffoli, J. Fagnan, O. Zaiane, and R. Campello. Relative validity criteria for community mining algorithms. In *International Conference on Advances in Social Networks Analysis and Mining*, Aug 2012. 4, 7, 114

[131] R. Rabbany, M. Takaffoli, J. Fagnan, O. R. Zaïane, and R. J. Campello. Communities validity: methodical evaluation of community mining algorithms. *Social Network Analysis and Mining*, 3(4):1039–1062, 2013. 4, 7

[132] R. Rabbany, S. Elatia, M. Takaffoli, and O. R. Zaïane. Collaborative learning of students in online discussion forums: A social network analysis perspective. In A. Peña-Ayala, editor, *Educational Data Mining: Applications and Trends*, pages 441–466. Springer International Publishing, 2014. 83, 85

[133] R. Rabbany, M. Takaffoli, J. Fagnan, O. R. Zaïane, and R. Campello. *Encyclopedia of Social Network Analysis and Mining*, chapter Relative Validity Criteria for Community Mining Algorithms, pages 1562–1576. Springer New York, New York, NY, 2014. 4, 7

[134] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3), 2007. 10

[135] S. Ravanbakhsh, R. Rabbany, and R. Greiner. Augmentative message passing for traveling salesman problem and graph partitioning. In *Advances in Neural Information Processing Systems*, pages 289–297, 2014. 8

[136] 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, 2002. 15

[137] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74 (1), 2006. 10

[138] P. Ronhovde and Z. Nussinov. Multiresolution community detection for megascale networks by information-based replica correlations. *Physical Review E*, 80(1):016109, 2009. 51

[139] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104(18):7327–7331, 2007. 2, 3, 13, 98

[140] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008. 11, 51, 77, 90

[141] M. Rosvall and C. T. Bergstrom. Mapping change in large networks. *PloS one*, 5(1), 2010. 2

[142] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987. 13, 19

[143] J. Ruths and D. Ruths. Control profiles of complex networks. *Science*, 343(6177):1373–1376, 2014. 59

[144] A. Sallaberry, F. Zaidi, and G. Melançon. Model for generating artificial social networks having community structures with small-world and scale-free properties. *Social Network Analysis and Mining*, pages 1–13, 2013. 29

[145] C. Seshadhri, T. G. Kolda, and A. Pinar. Community structure and scale-free collections of erdős-rényi graphs. *Physical Review E*, 85(5), 2012. 63

[146] J. Shao, Z. Han, Q. Yang, and T. Zhou. Community detection based on distance dynamics. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1075–1084, 2015. 60, 64

[147] V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences*, 100(21):12123–12128, 2003. 2, 3, 96

[148] M. Takaffoli, R. Rabbany, and O. R. Zaıane. Incremental local community identification in dynamic social networks. In *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2013. 11, 84, 87

[149] M. Takaffoli, R. Rabbany, and O. R. Zaïane. Community evolution prediction in dynamic social networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 9–16. IEEE, 2014. 11, 84, 87

[150] M. Tepper and G. Sapiro. From local to global communities in large networks through consensus. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 659–666. Springer, 2015. 11

[151] S. Theodoridis and K. Koutroumbas. Cluster validity. In *Pattern Recognition*, chapter 16. Elsevier Science, 4 edition, 2009. ISBN 9781597492720. 13

[152] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM review*, 53(3):526–543, 2011. 5, 60, 89, 90

[153] A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012. 6, 90, 91

[154] A. Vázquez. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67(5), May 2003. 60

[155] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka. Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, 3(4):209–235, 2010. 13, 17, 19, 20, 21

[156] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1073–1080, New York, NY, USA, 2009. ACM. 37, 51

[157] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010. 22, 31, 34, 36, 118

[158] D. L. Wallace. A method for comparing two hierarchical clusterings: Comment. *Journal of the American Statistical Association*, 78(383):569–576, 1983. 34

[159] M. Warrens. On similarity coefficients for 2x2 tables and correction for chance. *Psychometrika*, 73: 487–502, 2008. 10.1007/s11336-008-9059-y. 36

[160] M. J. Warrens. On the equivalence of cohen's kappa and the hubert-arabie adjusted rand index. *Journal of Classification*, 25:177–183, 2008. 36

[161] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994. 14

[162] D. J. Watts and S. H. Strogatz. Collective dynamics of âĂŸsmall-worldâĂŹnetworks. *Nature*, 393 (6684):440–442, 1998. 61

[163] L. H. Wong, P. Pattison, and G. Robins. A spatial model for social networks. *Physica A: Statistical Mechanics and its Applications*, 360(1):99–120, 2006. 63

[164] J. Wu, H. Xiong, and J. Chen. Adapting the right measures for k-means clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 877–886, New York, NY, USA, 2009. ACM. 34, 36

[165] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 824–833, 2007. 2

[166] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 3, 2012. 5, 60, 89

[167] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013. 11, 32, 41, 51, 77

[168] J. Yang, J. McAuley, and J. Leskovec. Community detection in networks with node attributes. *2013 IEEE 13th International Conference on Data Mining*, pages 1151–1156, 2013. 64

[169] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 927–936, 2009. 89

[170] Y. Yang, Y. Sun, S. Pandit, N. V. Chawla, and J. Han. Perspective on measurement metrics for community detection algorithms. In *Mining Social Networks and Security Informatics*, pages 227–242. Springer, 2013. 89

[171] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977. 21

[172] J. Zhao, G.-H. Ding, L. Tao, H. Yu, Z.-H. Yu, J.-H. Luo, Z.-W. Cao, and Y.-X. Li. Modular co-evolution of metabolic networks. *BMC bioinformatics*, 8(1):311, 2007. 2, 84

[173] D. Zhou, J. Li, and H. Zha. A new mallows distance based metric for comparing clusterings. In *Proceedings of the 22nd international conference on Machine learning*, pages 1028–1035. ACM, 2005. 40

[174] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *Proceedings of the Very Large Data Bases Endowment*, 2(1):718–729, 2009. 89, 92

[175] K. Zuev, M. Boguñá, G. Bianconi, and D. Krioukov. Emergence of soft communities from geometric preferential attachment. *Scientific reports*, 5, 2015. 59

# Appendix A

# Appendix of Chapter 2

## A.1 Sampling the Partitioning Space

We sample the partitioning space by randomizing the true partitioning, $p_d^*$, *i.e.*, by randomly merging and splitting communities and swapping nodes between them. The detailed procedure of sampling is described bellow.

```
$Q \leftarrow \{p_d^*\}$
**while** $|Q| < m/2$ **do**
  {generate splitted variations}
  **for** $p \in Q$ **do**
    $vp \leftarrow p$
    **for** $p_1 \in vp$ **do**
      **if** $random < split\_chance$ **then**
        $vp.splitRandom(p_1)$
    $Q.add(vp)$
  {generate merged variations}
  **for** $p \in Q$ **do**
    $vp \leftarrow p$
    **for** $p_1, p_2 \in vp$ **do**
      **if** $random < merge\_chance$ **then**
        $vp.merge(p_1, P2)$
    $Q.add(vp)$
{generate swapped variations}
**while** $|Q| < m/2$ **do**
  **for** $p \in Q$ **do**
    $vp \leftarrow p$
    **for** $p_1 \in vp$ **do**
      **for** $dp \in p1$ **do**
        **if** $random < swap\_chance$ **then**
          $vp.remove(dp)$
          $P_{random}.add(dp)$
    $Q.add(vp)$
```

**Algorithm 3:** Generating random partitionings

Code for this procedure, and all the other experiments reported in this thesis is available from:

- `https://github.com/rabbanyk/CommunityEvaluation`, and
- `https://github.com/rabbanyk/FARZ`

## A.2 Extended Results for a Subset of Criteria

Here, we report the results from [130], which incorporated a subset of criteria and distance combinations discussed in Chapter 2.

### A.2.1 Results on Real World Datasets

| Dataset | $K^*$ | # | $\overline{K}$ | $\overline{AMI}$ |
|---|---|---|---|---|
| karate | 2 | 60 | 3.57±1.23∈[2,6] | 0.46±0.27∈[-0.02,1] |
| strike | 3 | 60 | 3.17±1∈[2,5] | 0.59±0.27∈[-0.04,1] |
| polboks | 3 | 60 | 3.17±1.13∈[2,6] | 0.44±0.25∈[0.04,1] |
| football | 11 | 60 | 10.17±4.55∈[4,19] | 0.68±0.16∈[0.4,1] |

***Table A.1:*** *Statistics for sample partitionings of each real world dataset. For example, for the Karate Club dataset which has 2 communities in its ground truth, we have generated 60 different partitionings with average 3.57±1.23 clusters ranging from 2 to 6 and the "goodness" of the samples is on average 0.46±0.27 in terms of their AMI agreement.*

| Rank | Criterion | $\text{AMI}_{corr}$ | ARI | Jaccard | NMI |
|---|---|---|---|---|---|
| 1 | CIndex PCD | 0.907±0.058 | 1 | 1 | 1 |
| 2 | SWC2 NOD | 0.857±0.031 | 4 | 4 | 2 |
| 3 | Q | 0.85±0.083 | 2 | 2 | 3 |
| 4 | CIndex ARD | 0.826±0.162 | 6 | 15 | 5 |
| 5 | CIndex SPD | 0.811±0.126 | 3 | 10 | 4 |
| 6 | ASWC2 NOD | 0.809±0.043 | 5 | 11 | 6 |
| 7 | CIndex NOD | 0.794±0.096 | 12 | 3 | 9 |
| 8 | SWC2 PCD | 0.789±0.103 | 7 | 7 | 8 |
| 9 | SWC4 NOD | 0.778±0.075 | 9 | 5 | 7 |
| 10 | ASWC2 PCD | 0.772±0.088 | 10 | 9 | 10 |
| 11 | SWC2 SPD | 0.751±0.121 | 8 | 6 | 11 |
| 12 | Dunn01 ICD | 0.742±0.111 | 18 | 24 | 12 |
| 13 | ASWC2 SPD | 0.733±0.116 | 11 | 8 | 13 |
| 14 | Dunn00 PCD | 0.721±0.1 | 21 | 30 | 14 |
| 15 | DB ICD | 0.712±0.063 | 24 | 22 | 16 |
| 16 | Dunn00 ICD | 0.707±0.133 | 28 | 28 | 15 |
| 17 | Dunn03 ICD | 0.703±0.055 | 25 | 23 | 17 |
| 18 | SWC4 PCD | 0.7±0.072 | 14 | 12 | 21 |
| 19 | SWC4 SPD | 0.681±0.081 | 15 | 13 | 23 |
| 20 | SWC2 ARD | 0.681±0.302 | 17 | 26 | 19 |

***Table A.2:*** *Overall ranking of criteria on the real world datasets, based on the average Spearman's correlation of criteria with the AMI external index, $AMI_{corr}$. Ranking based on correlation with other external indexes is also reported.*

<table>
<tr><th colspan="6">Near Optimal Samples</th></tr>
<tr><th>Rank</th><th>Criterion</th><th>$\text{AMI}_{corr}$</th><th>ARI</th><th>Jaccard</th><th>NMI</th></tr>
<tr><td>1</td><td>Q</td><td>0.736±0.266</td><td>5</td><td>5</td><td>2</td></tr>
<tr><td>2</td><td>CIndex PCD</td><td>0.72±0.326</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>3</td><td>SWC2 SPD</td><td>0.718±0.389</td><td>3</td><td>3</td><td>4</td></tr>
<tr><td>4</td><td>CIndex SPD</td><td>0.716±0.14</td><td>4</td><td>4</td><td>1</td></tr>
<tr><td>5</td><td>SWC2 ICD</td><td>0.713±0.396</td><td>2</td><td>2</td><td>5</td></tr>
<tr><td>6</td><td>ASWC2 ICD</td><td>0.687±0.334</td><td>11</td><td>10</td><td>7</td></tr>
<tr><td>7</td><td>Dunn04 ICD</td><td>0.669±0.161</td><td>15</td><td>14</td><td>6</td></tr>
<tr><td>8</td><td>SWC2 PCD</td><td>0.651±0.383</td><td>6</td><td>6</td><td>10</td></tr>
<tr><td>9</td><td>ASWC2 SPD</td><td>0.65±0.352</td><td>12</td><td>12</td><td>9</td></tr>
<tr><td>10</td><td>SWC4 NOD</td><td>0.636±0.291</td><td>7</td><td>8</td><td>8</td></tr>
<tr><th colspan="6">Medium Far Samples</th></tr>
<tr><th>Rank</th><th>Criterion</th><th>$\text{AMI}_{corr}$</th><th>ARI</th><th>Jaccard</th><th>NMI</th></tr>
<tr><td>1</td><td>CIndex PCD</td><td>0.608±0.202</td><td>8</td><td>18</td><td>1</td></tr>
<tr><td>2</td><td>CIndex NOD</td><td>0.58±0.053</td><td>39</td><td>13</td><td>2</td></tr>
<tr><td>3</td><td>CIndex ARD</td><td>0.513±0.313</td><td>26</td><td>62</td><td>5</td></tr>
<tr><td>4</td><td>Dunn01 ICD</td><td>0.457±0.173</td><td>58</td><td>83</td><td>8</td></tr>
<tr><td>5</td><td>SWC2 NOD</td><td>0.447±0.19</td><td>5</td><td>9</td><td>3</td></tr>
<tr><td>6</td><td>ASWC2 PCD</td><td>0.446±0.191</td><td>7</td><td>3</td><td>9</td></tr>
<tr><td>7</td><td>SWC2 PCD</td><td>0.446±0.19</td><td>6</td><td>2</td><td>10</td></tr>
<tr><td>8</td><td>Dunn03 ICD</td><td>0.439±0.109</td><td>43</td><td>37</td><td>11</td></tr>
<tr><td>9</td><td>Dunn31 SPD</td><td>0.437±0.177</td><td>56</td><td>47</td><td>15</td></tr>
<tr><td>10</td><td>Dunn01 SPD</td><td>0.434±0.205</td><td>29</td><td>67</td><td>7</td></tr>
<tr><td>11</td><td>Q</td><td>0.409±0.353</td><td>4</td><td>7</td><td>16</td></tr>
<tr><td>12</td><td>DB ICD</td><td>0.405±0.072</td><td>40</td><td>38</td><td>18</td></tr>
<tr><td>13</td><td>Dunn00 ICD</td><td>0.404±0.17</td><td>127</td><td>112</td><td>6</td></tr>
<tr><td>14</td><td>Dunn41 ICD</td><td>0.391±0.216</td><td>87</td><td>101</td><td>19</td></tr>
<tr><td>15</td><td>CIndex SPD</td><td>0.389±0.268</td><td>1</td><td>27</td><td>4</td></tr>
<tr><th colspan="6">Far Far Samples</th></tr>
<tr><th>Rank</th><th>Criterion</th><th>$\text{AMI}_{corr}$</th><th>ARI</th><th>Jaccard</th><th>NMI</th></tr>
<tr><td>1</td><td>SWC2 NOD</td><td>0.634±0.217</td><td>3</td><td>13</td><td>1</td></tr>
<tr><td>2</td><td>ASWC2 NOD</td><td>0.583±0.191</td><td>5</td><td>21</td><td>2</td></tr>
<tr><td>3</td><td>Q</td><td>0.498±0.179</td><td>4</td><td>38</td><td>5</td></tr>
<tr><td>4</td><td>CIndex PCD</td><td>0.493±0.282</td><td>2</td><td>4</td><td>13</td></tr>
<tr><td>5</td><td>CIndex SPD</td><td>0.437±0.291</td><td>1</td><td>11</td><td>4</td></tr>
<tr><td>6</td><td>SWC4 NOD</td><td>0.436±0.344</td><td>8</td><td>2</td><td>25</td></tr>
<tr><td>7</td><td>SWC2 ARD</td><td>0.421±0.43</td><td>15</td><td>35</td><td>20</td></tr>
<tr><td>8</td><td>ASWC2 SPD</td><td>0.411±0.316</td><td>6</td><td>1</td><td>27</td></tr>
<tr><td>9</td><td>ASWC2 ARD</td><td>0.405±0.4</td><td>19</td><td>32</td><td>21</td></tr>
<tr><td>10</td><td>SWC2 PCD</td><td>0.376±0.371</td><td>12</td><td>17</td><td>32</td></tr>
</table>

***Table A.3:*** *Difficulty analysis of the results: considering ranking for partitionings near optimal ground truth, medium far and very far. Reported result are based on AMI and the Spearman's correlation.*

### A.2.2 Synthetic Benchmarks Datasets

| Dataset | $K^*$ | # | $\overline{K}$ | $\overline{AMI}$ |
|---|---|---|---|---|
| network1 | 4 | 60 | 3.4±1.17∈[2,6] | 0.46±0.23∈[0,1] |
| network2 | 3 | 60 | 3.1±1.27∈[2,7] | 0.49±0.22∈[0.13,1] |
| network3 | 2 | 60 | 3.3±1.13∈[2,6] | 0.47±0.23∈[0.11,1] |
| network4 | 7 | 60 | 5.17±2.49∈[2,12] | 0.57±0.2∈[0.18,1] |
| network5 | 2 | 60 | 3.5±1.36∈[2,8] | 0.44±0.22∈[0.11,1] |
| network6 | 5 | 60 | 5.8±2.55∈[2,12] | 0.68±0.2∈[0.27,1] |
| network7 | 4 | 60 | 5.2±2.65∈[2,12] | 0.47±0.19∈[0.13,1] |
| network8 | 5 | 60 | 5.37±2.04∈[2,10] | 0.67±0.21∈[0.32,1] |
| network9 | 5 | 60 | 5.5±2.05∈[2,10] | 0.69±0.19∈[0.37,1] |
| network10 | 6 | 60 | 5.33±2.51∈[2,11] | 0.63±0.19∈[0.24,1] |

***Table A.4:*** *Statistics for sample partitionings of each synthetic dataset. The benchmark generation parameters: 100 nodes with average degree 5 and maximum degree 50, where size of each community is between 5 and 50 and mixing parameter is* 0.1.

| Rank | Criterion | AMI$_{corr}$ | ARI | Jaccard | NMI |
|---|---|---|---|---|---|
| 1 | PB PCD | 0.454±0.15 | 1 | 1 | 1 |
| 2 | PB NOD | 0.448±0.146 | 2 | 2 | 2 |
| 3 | PB SPD | 0.445±0.144 | 3 | 3 | 4 |
| 4 | PB ARD | 0.44±0.149 | 4 | 4 | 5 |
| 5 | VRC ICD | 0.424±0.117 | 5 | 5 | 3 |
| 6 | Q | 0.391±0.381 | 17 | 6 | 12 |
| 7 | CIndex ARD | 0.365±0.173 | 6 | 7 | 6 |
| 8 | ASWC4 SPD | 0.358±0.101 | 12 | 12 | 7 |
| 9 | DB PCD | 0.358±0.108 | 15 | 9 | 10 |
| 10 | ASWC4 NOD | 0.357±0.114 | 10 | 10 | 8 |
| 11 | ASWC4 ARD | 0.356±0.1 | 13 | 8 | 9 |
| 12 | ASWC2 NOD | 0.341±0.128 | 16 | 17 | 11 |
| 13 | ZIndex SPD | 0.31±0.13 | 7 | 14 | 13 |
| 14 | ZIndex NOD | 0.299±0.131 | 8 | 16 | 14 |
| 15 | ZIndex ARD | 0.297±0.134 | 9 | 18 | 15 |
| 16 | ZIndex PCD | 0.292±0.131 | 14 | 19 | 16 |
| 17 | VRC ED | 0.285±0.124 | 18 | 15 | 17 |
| 18 | PBM ICD | 0.278±0.111 | 19 | 13 | 20 |
| 19 | CIndex ICD | 0.275±0.215 | 11 | 11 | 18 |
| 20 | SWC2 NOD | 0.25±0.16 | 20 | 23 | 19 |

***Table A.5:*** *Overall ranking of criteria based on AMI & Spearman's Correlation on the synthetic benchmarks with the same parameters as in Table A.4 but much higher mixing parameter, .7. We can see that in these settings, PB indexes outperform modularity Q.*

| Overall Results | | | | | |
|---|---|---|---|---|---|
| Rank | Criterion | AMI$_{corr}$ | ARI | Jaccard | NMI |
| 1 | Q | 0.894±0.018 | 1 | 2 | 1 |
| 2 | ASWC2 NOD | 0.854±0.056 | 3 | 4 | 2 |
| 3 | SWC2 NOD | 0.854±0.051 | 4 | 3 | 3 |
| 4 | CIndex PCD | 0.826±0.07 | 2 | 1 | 4 |
| 5 | CIndex SPD | 0.746±0.137 | 8 | 24 | 5 |
| 6 | SWC2 PCD | 0.743±0.047 | 5 | 5 | 6 |
| 7 | ASWC2 PCD | 0.739±0.048 | 6 | 6 | 7 |
| 8 | Dunn00 PCD | 0.707±0.11 | 11 | 26 | 8 |
| 9 | SWC4 NOD | 0.699±0.131 | 7 | 7 | 9 |
| 10 | SWC4 ARD | 0.689±0.124 | 9 | 8 | 10 |
| 11 | ASWC2 ARD | 0.683±0.108 | 15 | 21 | 11 |
| 12 | ASWC2 ED | 0.665±0.139 | 10 | 11 | 12 |
| 13 | SWC2 SPD | 0.657±0.124 | 14 | 16 | 13 |
| 14 | ASWC2 SPD | 0.651±0.196 | 16 | 17 | 15 |
| 15 | Dunn03 NOD | 0.645±0.156 | 23 | 33 | 14 |

| Near Optimal Results | | | | | |
|---|---|---|---|---|---|
| Rank | Criterion | AMI$_{corr}$ | ARI | Jaccard | NMI |
| 1 | CIndex PCD | 0.729±0.17 | 1 | 1 | 1 |
| 2 | Q | 0.722±0.111 | 6 | 5 | 5 |
| 3 | SWC2 SPD | 0.717±0.185 | 18 | 18 | 2 |
| 4 | SWC4 NOD | 0.709±0.201 | 5 | 6 | 4 |
| 5 | SWC2 ICD | 0.704±0.216 | 15 | 15 | 3 |
| 6 | SWC4 ARD | 0.674±0.183 | 7 | 7 | 6 |
| 7 | ASWC2 NOD | 0.66±0.261 | 20 | 19 | 7 |
| 8 | SWC2 NOD | 0.649±0.264 | 14 | 14 | 9 |
| Medium Far Results | | | | | |
| Rank | Criterion | AMI$_{corr}$ | ARI | Jaccard | NMI |
| 1 | SWC2 NOD | 0.455±0.191 | 5 | 11 | 3 |
| 2 | CIndex PCD | 0.453±0.245 | 1 | 2 | 5 |
| 3 | Q | 0.45±0.236 | 2 | 9 | 2 |
| 4 | ASWC2 NOD | 0.435±0.187 | 4 | 14 | 1 |
| 5 | Dunn00 ARD | 0.386±0.243 | 119 | 111 | 7 |
| 6 | Dunn00 PCD | 0.38±0.195 | 58 | 91 | 6 |
| 7 | CIndex NOD | 0.373±0.213 | 7 | 1 | 14 |
| 8 | Dunn01 NOD | 0.358±0.146 | 108 | 95 | 15 |
| Far Far Results | | | | | |
| Rank | Criterion | AMI$_{corr}$ | ARI | Jaccard | NMI |
| 1 | Q | 0.63±0.139 | 1 | 4 | 2 |
| 2 | ASWC2 NOD | 0.596±0.164 | 2 | 2 | 3 |
| 3 | SWC2 NOD | 0.57±0.159 | 3 | 3 | 5 |
| 4 | CIndex SPD | 0.565±0.132 | 4 | 25 | 1 |
| 5 | CIndex PCD | 0.446±0.142 | 5 | 1 | 21 |
| 6 | CIndex ARD | 0.433±0.25 | 10 | 106 | 4 |
| 7 | ASWC4 NOD | 0.397±0.119 | 15 | 63 | 11 |
| 8 | SWC2 PCD | 0.356±0.143 | 6 | 6 | 25 |

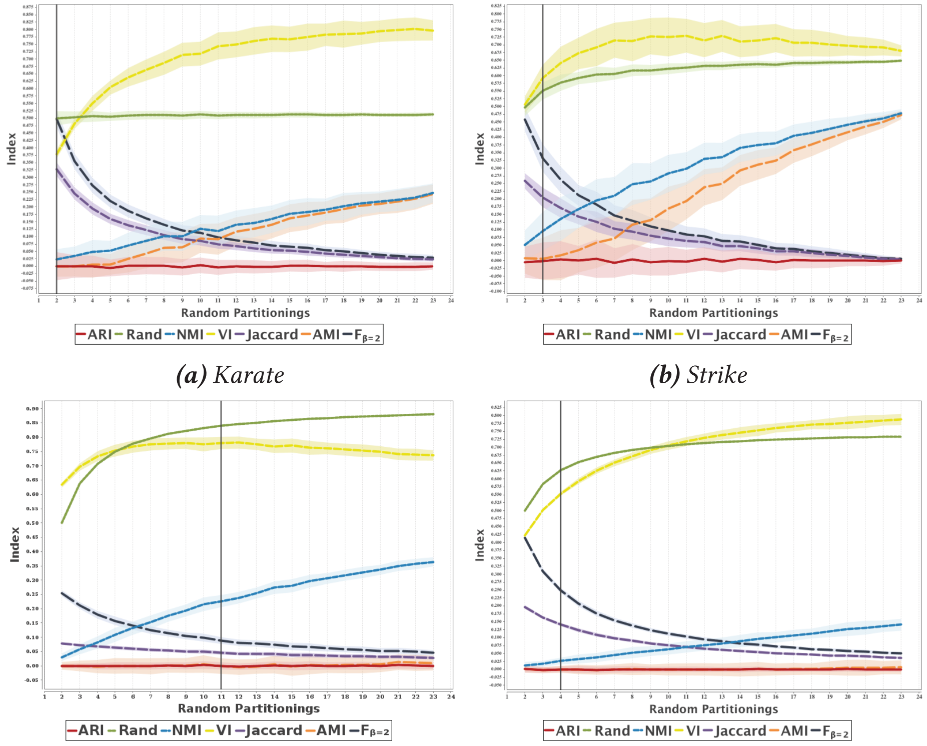
***Table A.6:*** *Overall ranking and difficulty analysis of the synthetic results. Here communities are well-separated with mixing parameter of .1. Similar to the last experiment, reported result are based on AMI and the Spearman's correlation.*

## A.3 Agreement Indexes Experiments

Here we first examine two desired properties for general clustering agreement indexes, and then we illustrate these properties in two adapted indexes for graphs.

### A.3.1 Bias of Unadjusted Indexes

In Figure A.1, we show *the bias of the unadjusted indexes*, where the average agreement of random partitionings to a true partitioning is plotted as a function of number of clusters, similar to the experiment performed in [157]. We can see that the average agreement increases for the unadjusted indexes when the number of clusters increases, while the adjusted rand index, *ARI*, is unaffected. Interestingly, we do not observe the same behaviour from *AMI* in all the datasets, while it is unaffected in football and GN datasets (where $k \ll N$), it increases with the number of clusters in the strike and karate dataset (where $k \ll N$ is not true).



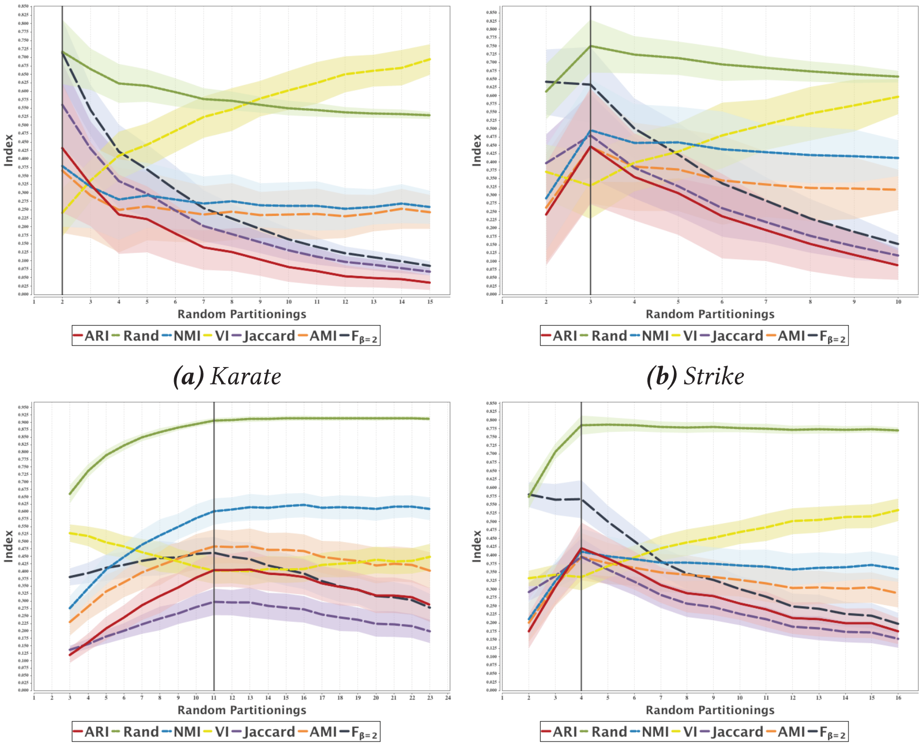
***(a)** Karate*

***(b)** Strike*

***(c)** Football*

***(d)** GN*

***Figure A.1:** Necessity of adjustment of external indexes for agreement at chance. Here we generated* 100 *sample partitionings for each k, then for each sample, we computed its agreement with the true partitioning for that dataset. The average and variance of these agreements are plotted as a function of the number of clusters. We can see that the unadjusted measures of Rand, $VI$, $Jaccard$, Fmeasure and NMI tend to increase/decrease as the the number of clusters in the random partitionings increases. While the Adjusted Rand Index (ARI) is unaffected and always returns zero for agreements at random.*



***(a)*** *Karate* ***(b)*** *Strike*

***(c)*** *Football* ***(d)*** *GN*

***Figure A.2:*** *Behaviour of different external indexes around the true number of clusters. We can see that the ARI exhibits a clear knee behaviour, i.e., , its values are relatively lower for partitionings with too many or too few clusters. While others such as NMI and Rand comply less with this knee shape.*

### A.3.2 Knee Shape Behaviour

Figure A.2, illustrates the behaviour of these criteria on different fragmentations of the ground-truth as a function of the number of clusters. The ideal behaviour is that the index should return relatively low scores for partitionings/fragmentations in which the number of clusters is much lower or higher than what we have in the ground-truth. In this figure, we can see that *ARI* exhibits this *knee shape* while *NMI* does not show this clearly. Table A.7, reports the average correlation of these external indexes over these four datasets. Here we used the similar sampling procedure described before but we generate merge and split versions separately, so that the obtained samples are fragmentations of the ground-truth obtained from repeated merging or splitting. Refer to the Appendix A.1 for the detailed sampling procedure.

There are different ways to compute the correlation between two vectors. The classic options are Pearson Product Moment coefficient or the Spearman's Rank correlation coefficient. The reported results in our experiments are based on the Spearman's Correlation, since we are interested in the correlation of rankings that an index provides for different partitionings and not the actual values of that index.

| Index | ARI | Rand | NMI | VI | Jaccard | AMI | $F_{\beta=2}$ |
|---|---|---|---|---|---|---|---|
| ARI | 1 | 0.73±0.18 | 0.67±0.07 | -0.80±0.17 | 0.85±0.08 | 0.76±0.15 | 0.64±0.16 |
| Rand | 0.73±0.18 | 1 | 0.83±0.12 | -0.46±0.42 | 0.41±0.32 | 0.71±0.11 | 0.13±0.46 |
| NMI | 0.67±0.07 | 0.83±0.12 | 1 | -0.43±0.27 | 0.31±0.17 | 0.93±0.07 | 0.04±0.10 |
| VI | -0.80±0.17 | -0.46±0.42 | -0.43±0.27 | 1 | -0.93±0.02 | -0.54±0.27 | -0.82±0.21 |
| Jaccard | 0.85±0.08 | 0.41±0.32 | 0.31±0.17 | -0.93±0.02 | 1 | 0.46±0.28 | 0.90±0.13 |
| AMI | 0.76±0.15 | 0.71±0.11 | 0.93±0.07 | -0.54±0.27 | 0.46±0.28 | 1 | 0.25±0.13 |
| $F_{\beta=2}$ | 0.64±0.16 | 0.13±0.46 | 0.04±0.10 | -0.82±0.21 | 0.90±0.13 | 0.25±0.13 | 1 |

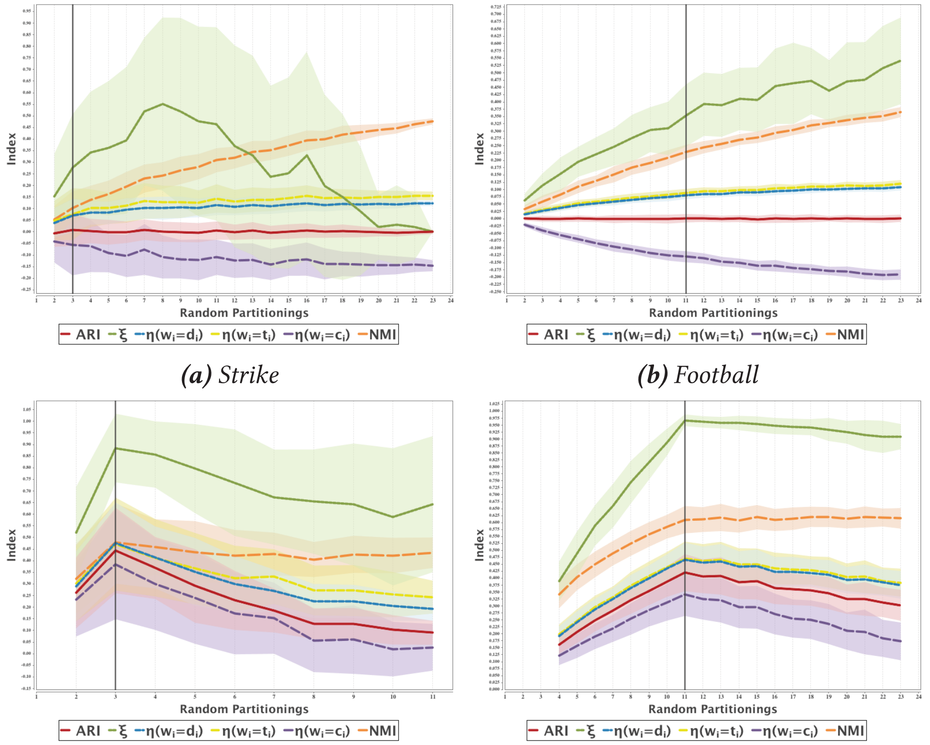
***Table A.7:*** *Correlation between external indexes averaged for datasets of Figure A.2, computed based on Spearman's Correlation. Here we can see for example that ARI behaves more similar to, has a higher correlation with, AMI compared to NMI respectively.*

### A.3.3 Graph Partitioning Agreement Indexes

We define a weighted version of the clustering agreement measures here; where nodes with more importance affect the agreement measure more. This is prior to generalizations proposed in Chapter 3. Here, we alter these measures to directly assess the structural similarity of these sub-graphs by focusing on the edges instead of nodes. More specifically, instead of $n_{ij} = |U_i \cap V_j|$, we first use: $\eta_{ij} = \sum_{l \in U_i \cap V_j} w_l$, where $w_l$ is the weight of item $l$. If we assume all items are weighted equally as 1, then $\eta_{ij} = n_{ij}$. Instead, we can consider weight of a node equal to its degree in the graph. Using this degree weighted index can be more informative for comparing agreements between community mining results, since nodes with different degrees have different importance in the network, and therefore should be weighted differently in the agreement index. Another possibility is to use the clustering coefficient of a node as its weight, so that nodes that contribute to more triangles – have more connected neighbours – weight more. Second, we consider the structure in a more direct way by counting the edges that are common between $U_i$ and $V_j$. More formally, we define: $\xi_{ij} = \sum_{k,l \in U_i \cap V_j} A_{kl}$, which sums all the edges in the overlap of cluster $U_i$ and $V_j$. Figure A.3 shows the constant baseline of these adapted criteria for agreements at random, and also the knee shape of the adapted measures around the true number of clusters, same as what we have for the original ARI. Therefore, one can safely apply one of these measures depending on the application at hand. Table A.8 summarizes the correlation between each pair of the external measures.

| Index | ARI | $\xi$ | $\eta_{w_i=d_i}$ | $\eta_{w_i=t_i}$ | $\eta_{w_i=c_i}$ | NMI |
|---|---|---|---|---|---|---|
| ARI | 1±0 | 0.571±0.142 | 0.956±0.031 | 0.819±0.135 | 0.838±0.087 | 0.736±0.096 |
| $\xi$ | 0.571±0.142 | 1±0 | 0.623±0.133 | 0.572±0.169 | 0.45±0.109 | 0.497±0.2 |
| $\eta_{w_i=d_i}$ | 0.956±0.031 | 0.623±0.133 | 1±0 | 0.876±0.097 | 0.777±0.106 | 0.787±0.094 |
| $\eta_{w_i=t_i}$ | 0.819±0.135 | 0.572±0.169 | 0.876±0.097 | 1±0 | 0.848±0.056 | 0.759±0.107 |
| $\eta_{w_i=c_i}$ | 0.838±0.087 | 0.45±0.109 | 0.777±0.106 | 0.848±0.056 | 1±0 | 0.6±0.064 |
| NMI | 0.736±0.096 | 0.497±0.2 | 0.787±0.094 | 0.759±0.107 | 0.6±0.064 | 1±0 |

***Table A.8:*** *Correlation between adapted external indexes on karate and strike datasets, computed based on Spearman's Correlation. Here, $\eta_{w_i=d_i}$, $\eta_{w_i=t_i}$, and $\eta_{w_i=c_i}$ denote the weighted ARI where each node is weighted respectively by, its degree, the number of triangles it belongs to, or its clustering coefficient. The $\xi$, on the other hand, stands for the structural agreement based on number of edges.*



***(a)** Strike*

***(b)** Football*

***(c)** Strike*

***(d)** Football*

***Figure A.3:** Adapted agreement measures for graphs. On top we see that the adapted measures, specially the weighted indexes by degree ($d_i$) and the number of triangles ($t_i$), are adjusted by chance, which can not be seen for the structural edge based version ($\xi$). The bottom figures illustrate the perseverance of the knee behaviour in the adapted measures.*

## A.4 Extended Results for All Combinations

Here, we report extended results for the tables in Chapter 2, *i.e.*, the following tables correspond to the tables reported in Chapter 2, but are expanded to include more criteria/rows.

### A.4.1 Results on Real World Datasets

***Table A.9:*** *Extended results for Table 2.2, which shows the overall ranking of criteria on the real world datasets.*

| Rank | Criterion | $ARI_{corr}$ | Rand | Jaccard | NMI | AMI |
|---|---|---|---|---|---|---|
| 1 | ZIndex' TO | 0.925±0.018 | 9 | 148 | 9 | 7 |
| 2 | ZIndex' $\hat{PC}$ | 0.923±0.012 | 2 | 197 | 2 | 2 |
| 3 | ZIndex' $\hat{NPC}$ | 0.923±0.012 | 3 | 198 | 1 | 1 |
| 4 | ZIndex' IC2 | 0.922±0.024 | 8 | 182 | 5 | 3 |
| 5 | ZIndex' $\hat{TO}$ | 0.922±0.016 | 10 | 153 | 8 | 8 |
| 6 | ZIndex' $\hat{NPO}$ | 0.921±0.014 | 6 | 204 | 3 | 4 |
| 7 | ZIndex' ICV2 | 0.919±0.04 | 18 | 163 | 12 | 10 |
| 8 | ZIndex' PC | 0.918±0.018 | 4 | 207 | 10 | 11 |
| 9 | ZIndex' IC3 | 0.918±0.039 | 19 | 165 | 15 | 12 |
| 10 | ZIndex' $\hat{NOV}$ | 0.915±0.014 | 11 | 213 | 6 | 9 |
| 11 | ZIndex' IC1 | 0.912±0.02 | 5 | 235 | 13 | 20 |
| 12 | ZIndex' NPE2.0 | 0.911±0.03 | 26 | 168 | 21 | 15 |
| 13 | ZIndex' NOV | 0.91±0.023 | 12 | 225 | 18 | 21 |
| 14 | ZIndex' ICV1 | 0.91±0.023 | 13 | 226 | 19 | 22 |
| 15 | ZIndex' $\hat{NPE}$2.0 | 0.91±0.025 | 23 | 184 | 22 | 19 |
| 16 | ZIndex' NPL2.0 | 0.909±0.02 | 24 | 202 | 14 | 13 |
| 17 | ZIndex' M | 0.908±0.028 | 25 | 149 | 26 | 23 |
| 18 | ZIndex' ICV3 | 0.908±0.057 | 29 | 176 | 28 | 25 |
| 19 | ZIndex' NP2.0 | 0.907±0.021 | 20 | 212 | 16 | 14 |
| 20 | ZIndex' $\hat{NPL}$2.0 | 0.906±0.022 | 21 | 216 | 17 | 17 |
| 21 | ZIndex' $\hat{NP}$2.0 | 0.906±0.022 | 22 | 217 | 20 | 18 |
| 22 | ZIndex' $\hat{NO}$ | 0.905±0.022 | 16 | 253 | 11 | 16 |
| 23 | ZIndex' NO | 0.904±0.034 | 7 | 250 | 23 | 31 |
| 24 | ZIndex' $\hat{MM}$ | 0.903±0.037 | 17 | 233 | 24 | 30 |
| 25 | CIndex SP | 0.9±0.02 | 1 | 251 | 31 | 42 |
| 26 | ZIndex' $\hat{NPL}$3.0 | 0.899±0.032 | 30 | 200 | 27 | 24 |
| 27 | ZIndex' $\hat{NP}$3.0 | 0.899±0.033 | 33 | 196 | 29 | 27 |
| 28 | ZIndex' $\hat{NPE}$3.0 | 0.899±0.048 | 31 | 205 | 35 | 33 |
| 29 | ZIndex $\hat{AR}$ | 0.898±0.035 | 14 | 264 | 30 | 36 |
| 30 | ZIndex' NPE3.0 | 0.897±0.052 | 35 | 187 | 39 | 34 |
| 31 | ZIndex' NPL3.0 | 0.897±0.038 | 36 | 170 | 32 | 28 |
| 32 | ZIndex SP | 0.895±0.036 | 28 | 215 | 40 | 41 |
| 33 | ZIndex' NP3.0 | 0.895±0.039 | 37 | 166 | 34 | 29 |
| 34 | ZIndex AR | 0.895±0.039 | 15 | 255 | 36 | 38 |
| 35 | ZIndex' A | 0.894±0.045 | 32 | 158 | 38 | 35 |
| 36 | ZIndex' MD | 0.894±0.048 | 34 | 179 | 33 | 32 |
| 37 | ZIndex' $\hat{A}$ | 0.891±0.05 | 27 | 241 | 37 | 37 |
| 38 | Q | 0.878±0.034 | 45 | 110 | 45 | 44 |
| 39 | CIndex' NPE3.0 | 0.876±0.054 | 43 | 9 | 4 | 6 |
| 40 | CIndex' ICV3 | 0.869±0.069 | 44 | 4 | 7 | 5 |
| 41 | CIndex AR | 0.864±0.031 | 40 | 268 | 42 | 40 |
| 42 | CIndex $\hat{AR}$ | 0.861±0.032 | 42 | 266 | 41 | 39 |
| 43 | CIndex' $\hat{NPE}$3.0 | 0.858±0.07 | 47 | 8 | 25 | 26 |
| 44 | ZIndex' $\hat{MD}$ | 0.856±0.101 | 38 | 323 | 43 | 45 |
| 45 | SWC0 IC1 | 0.847±0.09 | 41 | 108 | 46 | 47 |
| 46 | SWC0 IC2 | 0.838±0.092 | 49 | 11 | 50 | 49 |
| 47 | SWC0 NO | 0.837±0.106 | 39 | 146 | 48 | 50 |
| 48 | SWC0 IC3 | 0.819±0.104 | 57 | 7 | 58 | 52 |
| 49 | SWC0 NOV | 0.814±0.094 | 52 | 26 | 54 | 56 |
| 50 | SWC0 ICV1 | 0.814±0.094 | 53 | 27 | 55 | 57 |

***Table A.10:*** *Extended results for Table 2.3, which gives a difficulty analysis of the results.*

| Near Optimal Samples | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | ZIndex' $\hat{NPC}$ | 0.851±0.081 | 1 | 3 | 4 | 5 |
| 2 | ZIndex' $\hat{PC}$ | 0.851±0.081 | 2 | 4 | 3 | 3 |
| 3 | ZIndex SP | 0.847±0.084 | 18 | 2 | 8 | 8 |
| 4 | ZIndex' $\hat{NPO}$ | 0.845±0.088 | 3 | 9 | 6 | 6 |
| 5 | DB ICV2 | 0.845±0.065 | 30 | 1 | 31 | 30 |
| 6 | ZIndex' $\hat{NPE}$3.0 | 0.842±0.082 | 10 | 5 | 2 | 2 |
| 7 | ZIndex' ICV3 | 0.839±0.084 | 4 | 20 | 20 | 21 |
| 8 | ZIndex' $\hat{NOV}$ | 0.835±0.093 | 11 | 14 | 15 | 15 |
| 9 | ZIndex' $\hat{TO}$ | 0.835±0.09 | 9 | 10 | 7 | 7 |
| 10 | ZIndex' $\hat{NPE}$2.0 | 0.834±0.089 | 13 | 8 | 1 | 1 |
| 11 | ZIndex' TO | 0.834±0.089 | 7 | 16 | 11 | 11 |
| 12 | ZIndex' IC2 | 0.834±0.095 | 5 | 23 | 18 | 18 |
| 13 | ZIndex' NPL2.0 | 0.834±0.089 | 15 | 7 | 5 | 4 |
| 14 | ZIndex' NPE3.0 | 0.834±0.089 | 17 | 6 | 9 | 10 |
| 15 | ZIndex' NPE2.0 | 0.833±0.089 | 12 | 11 | 10 | 9 |
| 16 | ZIndex' ICV2 | 0.829±0.091 | 6 | 24 | 23 | 23 |
| 17 | ZIndex' IC3 | 0.829±0.091 | 8 | 27 | 24 | 22 |
| 18 | ZIndex' PC | 0.823±0.092 | 14 | 28 | 17 | 17 |
| 19 | ZIndex' $\hat{NPL}$3.0 | 0.818±0.104 | 24 | 19 | 14 | 14 |
| 20 | ZIndex' $\hat{NP}$3.0 | 0.817±0.104 | 23 | 18 | 16 | 16 |
| 21 | ZIndex' NPL3.0 | 0.817±0.103 | 25 | 12 | 19 | 19 |
| 22 | ZIndex' $\hat{NP}$2.0 | 0.816±0.097 | 20 | 22 | 12 | 12 |
| 23 | ZIndex' IC1 | 0.815±0.098 | 16 | 34 | 22 | 24 |
| 24 | ZIndex' NP3.0 | 0.813±0.108 | 27 | 15 | 21 | 20 |
| 25 | ZIndex' NP2.0 | 0.806±0.102 | 19 | 33 | 26 | 25 |
| 26 | ZIndex' $\hat{NPL}$2.0 | 0.8±0.113 | 29 | 32 | 13 | 13 |
| 27 | ZIndex' NOV | 0.798±0.112 | 21 | 37 | 27 | 27 |
| 28 | ZIndex' ICV1 | 0.798±0.112 | 22 | 38 | 28 | 28 |
| 29 | ZIndex' $\hat{NO}$ | 0.794±0.108 | 28 | 39 | 25 | 26 |
| 30 | ZIndex' NO | 0.788±0.118 | 26 | 40 | 29 | 29 |
| 31 | CIndex' ICV3 | 0.785±0.097 | 31 | 25 | 37 | 38 |
| 32 | CIndex' IC3 | 0.78±0.098 | 34 | 17 | 42 | 42 |
| 33 | CIndex' ICV2 | 0.769±0.089 | 36 | 30 | 32 | 32 |
| 34 | PB' $\hat{PC}$ | 0.766±0.273 | 41 | 13 | 54 | 52 |
| 35 | DB IC2 | 0.765±0.136 | 32 | 31 | 44 | 43 |
| 36 | ZIndex' M | 0.763±0.139 | 33 | 29 | 30 | 31 |
| 37 | Q | 0.762±0.166 | 39 | 21 | 41 | 41 |
| 38 | DB ICV3 | 0.757±0.126 | 37 | 35 | 38 | 36 |
| 39 | DB IC3 | 0.753±0.176 | 35 | 36 | 39 | 39 |
| 40 | PB' PC | 0.753±0.289 | 45 | 26 | 71 | 71 |
| 41 | DB $\hat{TO}$ | 0.735±0.114 | 43 | 45 | 66 | 61 |
| 42 | DB $\hat{NPO}$ | 0.721±0.136 | 38 | 69 | 49 | 48 |
| 43 | ZIndex' A | 0.718±0.158 | 42 | 42 | 35 | 34 |
| 44 | ASWC1' NPE3.0 | 0.716±0.114 | 49 | 44 | 68 | 66 |
| 45 | CIndex' NPE3.0 | 0.71±0.051 | 52 | 48 | 34 | 35 |
| 46 | SWC1 $\hat{NO}$ | 0.71±0.112 | 53 | 59 | 48 | 50 |
| 47 | DB $\hat{PC}$ | 0.706±0.172 | 50 | 78 | 59 | 58 |
| 48 | DB $\hat{NPC}$ | 0.706±0.171 | 51 | 79 | 60 | 59 |
| 49 | ZIndex $\hat{AR}$ | 0.705±0.264 | 46 | 86 | 43 | 44 |
| 50 | ZIndex' $\hat{MM}$ | 0.704±0.173 | 44 | 92 | 33 | 33 |

***Table A.11:*** *Extended results for Table 2.3, which gives a difficulty analysis of the results.*

| Medium Far Samples | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | ZIndex’ TO | 0.775±0.087 | 5 | 361 | 22 | 20 |
| 2 | ZIndex’ $\hat{TO}$ | 0.771±0.091 | 6 | 386 | 19 | 17 |
| 3 | ZIndex’ IC3 | 0.768±0.134 | 2 | 372 | 16 | 13 |
| 4 | ZIndex’ ICV2 | 0.766±0.124 | 3 | 370 | 2 | 2 |
| 5 | ZIndex’ NPL3.0 | 0.762±0.079 | 12 | 349 | 28 | 27 |
| 6 | ZIndex’ ICV3 | 0.757±0.12 | 4 | 376 | 21 | 19 |
| 7 | ZIndex’ NP3.0 | 0.756±0.085 | 15 | 354 | 29 | 28 |
| 8 | ZIndex’ $\hat{PC}$ | 0.755±0.122 | 9 | 417 | 4 | 4 |
| 9 | ZIndex’ $N\hat{PC}$ | 0.755±0.122 | 11 | 418 | 3 | 3 |
| 10 | ZIndex’ NPE2.0 | 0.753±0.107 | 10 | 373 | 14 | 14 |
| 11 | ZIndex’ NPE3.0 | 0.746±0.093 | 8 | 369 | 24 | 24 |
| 12 | ZIndex’ $N\hat{PO}$ | 0.744±0.123 | 14 | 437 | 5 | 5 |
| 13 | ZIndex’ PC | 0.743±0.12 | 13 | 421 | 7 | 7 |
| 14 | ZIndex’ $NP\hat{L}3.0$ | 0.742±0.098 | 16 | 416 | 27 | 26 |
| 15 | ZIndex’ $NP\hat{E}3.0$ | 0.742±0.102 | 7 | 397 | 15 | 15 |
| 16 | ZIndex’ $N\hat{P}3.0$ | 0.741±0.098 | 17 | 420 | 26 | 25 |
| 17 | ZIndex’ IC2 | 0.737±0.145 | 19 | 446 | 1 | 1 |
| 18 | ZIndex’ $N\hat{OV}$ | 0.735±0.127 | 20 | 452 | 6 | 6 |
| 19 | ZIndex’ $NP\hat{E}2.0$ | 0.733±0.13 | 18 | 432 | 10 | 10 |
| 20 | ZIndex’ M | 0.728±0.128 | 30 | 322 | 44 | 41 |
| 21 | ZIndex’ NPL2.0 | 0.726±0.116 | 23 | 441 | 20 | 21 |
| 22 | ZIndex’ NOV | 0.725±0.128 | 21 | 450 | 12 | 11 |
| 23 | ZIndex’ ICV1 | 0.725±0.128 | 22 | 451 | 13 | 12 |
| 24 | ZIndex’ $NP\hat{L}2.0$ | 0.718±0.125 | 25 | 462 | 25 | 23 |
| 25 | ZIndex’ $N\hat{P}2.0$ | 0.717±0.125 | 24 | 463 | 23 | 22 |
| 26 | ZIndex’ NP2.0 | 0.71±0.128 | 27 | 469 | 31 | 29 |
| 27 | ZIndex’ IC1 | 0.701±0.161 | 28 | 486 | 8 | 8 |
| 28 | ZIndex’ NO | 0.699±0.158 | 29 | 489 | 17 | 18 |
| 29 | ZIndex’ $\hat{MM}$ | 0.694±0.168 | 31 | 458 | 40 | 32 |
| 30 | Q | 0.69±0.151 | 58 | 70 | 79 | 72 |
| 31 | ZIndex’ A | 0.69±0.144 | 34 | 366 | 58 | 54 |
| 32 | ZIndex’ MD | 0.689±0.141 | 36 | 319 | 51 | 44 |
| 33 | ZIndex’ $N\hat{O}$ | 0.683±0.183 | 32 | 500 | 9 | 9 |
| 34 | ZIndex’ $\hat{A}$ | 0.681±0.165 | 33 | 471 | 41 | 38 |
| 35 | CIndex’ ICV2 | 0.67±0.357 | 38 | 57 | 30 | 30 |
| 36 | ZIndex SP | 0.666±0.095 | 26 | 472 | 37 | 35 |
| 37 | CIndex’ PC | 0.663±0.413 | 59 | 38 | 65 | 59 |
| 38 | CIndex’ IC3 | 0.661±0.389 | 44 | 50 | 54 | 50 |
| 39 | CIndex’ IC2 | 0.657±0.384 | 61 | 34 | 71 | 70 |
| 40 | PB’ $\hat{PC}$ | 0.649±0.03 | 101 | 4 | 150 | 137 |
| 41 | CIndex’ $\hat{PC}$ | 0.648±0.395 | 65 | 54 | 57 | 52 |
| 42 | CIndex’ $N\hat{PC}$ | 0.648±0.395 | 64 | 55 | 56 | 51 |
| 43 | CIndex’ ICV3 | 0.642±0.347 | 35 | 173 | 11 | 16 |
| 44 | CIndex’ $N\hat{PO}$ | 0.635±0.39 | 78 | 56 | 63 | 60 |
| 45 | CIndex’ $NP\hat{E}3.0$ | 0.626±0.359 | 37 | 149 | 42 | 45 |
| 46 | PB’ PC | 0.623±0.06 | 112 | 28 | 200 | 157 |
| 47 | CIndex’ $N\hat{OV}$ | 0.622±0.397 | 88 | 42 | 75 | 74 |
| 48 | CIndex’ NPE3.0 | 0.614±0.377 | 39 | 220 | 35 | 34 |
| 49 | CIndex SP | 0.614±0.078 | 1 | 517 | 45 | 58 |
| 50 | CIndex’ $\hat{TO}$ | 0.614±0.397 | 87 | 58 | 70 | 69 |

***Table A.12:*** *Extended results for Table 2.3, which gives a difficulty analysis of the results.*

| Far Far Samples | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | $\text{ARI}_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | ZIndex' ICV2 | 0.724±0.066 | 36 | 520 | 4 | 9 |
| 2 | ZIndex' IC3 | 0.72±0.062 | 40 | 523 | 11 | 19 |
| 3 | ZIndex' ICV3 | 0.717±0.059 | 47 | 511 | 23 | 25 |
| 4 | ZIndex' IC2 | 0.715±0.072 | 35 | 540 | 3 | 6 |
| 5 | ZIndex' TO | 0.706±0.064 | 49 | 519 | 16 | 14 |
| 6 | ZIndex' $\hat{NPO}$ | 0.704±0.076 | 44 | 547 | 1 | 3 |
| 7 | ZIndex' $\hat{TO}$ | 0.704±0.062 | 51 | 522 | 13 | 5 |
| 8 | ZIndex' NPE2.0 | 0.701±0.057 | 55 | 505 | 15 | 7 |
| 9 | ZIndex' $\hat{NPC}$ | 0.698±0.083 | 45 | 552 | 6 | 10 |
| 10 | ZIndex' $\hat{PC}$ | 0.697±0.083 | 46 | 553 | 9 | 11 |
| 11 | ZIndex' $\hat{NPE2.0}$ | 0.688±0.047 | 57 | 521 | 24 | 23 |
| 12 | ZIndex' NPL2.0 | 0.688±0.072 | 58 | 529 | 12 | 4 |
| 13 | ZIndex' $\hat{NOV}$ | 0.684±0.081 | 48 | 554 | 14 | 15 |
| 14 | ZIndex' $M$ | 0.684±0.067 | 71 | 486 | 33 | 24 |
| 15 | ZIndex' $\hat{NPL3.0}$ | 0.682±0.081 | 59 | 518 | 17 | 12 |
| 16 | ZIndex' NPL3.0 | 0.682±0.083 | 65 | 496 | 21 | 13 |
| 17 | ZIndex' $\hat{NP3.0}$ | 0.682±0.077 | 61 | 516 | 19 | 17 |
| 18 | ZIndex' NP2.0 | 0.68±0.098 | 53 | 542 | 10 | 8 |
| 19 | ZIndex' $\hat{NPL2.0}$ | 0.68±0.075 | 62 | 539 | 20 | 20 |
| 20 | ZIndex' $\hat{NP2.0}$ | 0.68±0.074 | 56 | 541 | 18 | 18 |
| 21 | ZIndex' $A$ | 0.678±0.106 | 72 | 482 | 32 | 21 |
| 22 | ZIndex' PC | 0.677±0.117 | 39 | 557 | 27 | 32 |
| 23 | ZIndex' NP3.0 | 0.676±0.085 | 69 | 490 | 25 | 16 |
| 24 | ZIndex' $\hat{NPE3.0}$ | 0.672±0.046 | 60 | 524 | 36 | 37 |
| 25 | ZIndex' NPE3.0 | 0.667±0.052 | 63 | 517 | 37 | 35 |
| 26 | ZIndex' NOV | 0.663±0.13 | 41 | 559 | 28 | 33 |
| 27 | ZIndex' ICV1 | 0.663±0.13 | 42 | 560 | 29 | 34 |
| 28 | ZIndex' $\hat{MM}$ | 0.66±0.091 | 67 | 546 | 40 | 39 |
| 29 | ZIndex' $\hat{A}$ | 0.655±0.102 | 70 | 533 | 39 | 38 |
| 30 | ZIndex' IC1 | 0.655±0.132 | 43 | 566 | 34 | 40 |
| 31 | ZIndex' $\hat{NO}$ | 0.651±0.106 | 52 | 567 | 22 | 26 |
| 32 | Q | 0.643±0.033 | 86 | 444 | 50 | 45 |
| 33 | ZIndex' NO | 0.638±0.158 | 38 | 572 | 38 | 47 |
| 34 | ZIndex' $MD$ | 0.63±0.099 | 78 | 513 | 43 | 41 |
| 35 | ZIndex SP | 0.618±0.101 | 68 | 543 | 61 | 85 |
| 36 | ZIndex $\hat{AR}$ | 0.6±0.159 | 66 | 573 | 48 | 67 |
| 37 | SWC0 IC1 | 0.598±0.169 | 80 | 405 | 30 | 30 |
| 38 | ZIndex AR | 0.591±0.171 | 64 | 570 | 56 | 79 |
| 39 | SWC0 $\hat{NO}$ | 0.588±0.145 | 104 | 424 | 26 | 1 |
| 40 | SWC0 NO | 0.584±0.191 | 77 | 459 | 31 | 44 |
| 41 | CIndex SP | 0.578±0.107 | 54 | 571 | 46 | 103 |
| 42 | ZIndex' $\hat{MD}$ | 0.562±0.158 | 74 | 575 | 57 | 69 |
| 43 | CIndex' NPE3.0 | 0.544±0.161 | 129 | 183 | 7 | 2 |
| 44 | SWC0 IC2 | 0.54±0.186 | 126 | 195 | 60 | 43 |
| 45 | ASWC0 IC2 | 0.529±0.243 | 110 | 218 | 42 | 31 |
| 46 | ASWC0 IC1 | 0.524±0.2 | 81 | 451 | 2 | 28 |
| 47 | ASWC0 IC3 | 0.521±0.254 | 120 | 176 | 54 | 49 |
| 48 | ASWC0 NO | 0.51±0.212 | 79 | 480 | 5 | 42 |
| 49 | CIndex' ICV3 | 0.506±0.25 | 148 | 130 | 41 | 29 |
| 50 | CIndex' $\hat{NPE3.0}$ | 0.504±0.177 | 144 | 155 | 35 | 27 |
| | | ⋮ | | | | |
| 117 | PB' PC | 0.372±0.126 | 197 | 170 | 159 | 129 |

### A.4.2 Synthetic Benchmarks Datasets

***Table A.13:*** *Extended results for Table 2.5, where communities are well-separated with* $\mu = 0.1$.

| Overall Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | ZIndex' ICV2 | 0.96±0.029 | 5 | 32 | 3 | 3 |
| 2 | ZIndex' IC3 | 0.958±0.028 | 4 | 42 | 2 | 2 |
| 3 | ZIndex' IC2 | 0.958±0.033 | 1 | 58 | 1 | 1 |
| 4 | ZIndex' $\hat{P}C$ | 0.953±0.04 | 3 | 78 | 6 | 6 |
| 5 | ZIndex' $N\hat{P}C$ | 0.953±0.04 | 2 | 79 | 7 | 7 |
| 6 | ZIndex' ICV3 | 0.953±0.027 | 8 | 44 | 4 | 5 |
| 7 | ZIndex' $N\hat{P}O$ | 0.951±0.041 | 6 | 83 | 9 | 9 |
| 8 | ZIndex' $\hat{T}O$ | 0.949±0.045 | 13 | 60 | 17 | 17 |
| 9 | ZIndex' $N\hat{O}V$ | 0.949±0.042 | 7 | 90 | 8 | 8 |
| 10 | ZIndex' TO | 0.948±0.046 | 16 | 50 | 21 | 21 |
| 11 | ZIndex' PC | 0.947±0.043 | 10 | 77 | 16 | 15 |
| 12 | ZIndex' $NP\hat{E}2.0$ | 0.947±0.042 | 11 | 68 | 13 | 13 |
| 13 | ZIndex' NPE2.0 | 0.946±0.043 | 17 | 51 | 20 | 20 |
| 14 | ZIndex' NOV | 0.941±0.047 | 14 | 95 | 18 | 18 |
| 15 | ZIndex' ICV1 | 0.941±0.047 | 15 | 96 | 19 | 19 |
| 16 | ZIndex' $N\hat{O}$ | 0.939±0.052 | 9 | 125 | 11 | 12 |
| 17 | ZIndex' $NP\hat{L}2.0$ | 0.938±0.052 | 19 | 98 | 22 | 23 |
| 18 | ZIndex' $N\hat{P}2.0$ | 0.938±0.051 | 20 | 92 | 24 | 25 |
| 19 | ZIndex' NPL2.0 | 0.938±0.049 | 21 | 81 | 25 | 26 |
| 20 | ZIndex' IC1 | 0.937±0.05 | 12 | 115 | 14 | 14 |
| 21 | ZIndex' NO | 0.933±0.052 | 18 | 122 | 23 | 24 |
| 22 | ZIndex' NP2.0 | 0.932±0.051 | 22 | 94 | 28 | 27 |
| 23 | ZIndex' $NP\hat{E}3.0$ | 0.913±0.066 | 27 | 114 | 27 | 28 |
| 24 | ZIndex' M | 0.913±0.036 | 23 | 127 | 5 | 4 |
| 25 | ZIndex' A | 0.911±0.036 | 26 | 117 | 12 | 11 |
| 26 | ZIndex' $NP\hat{L}3.0$ | 0.909±0.064 | 28 | 111 | 33 | 33 |
| 27 | ZIndex' $N\hat{P}3.0$ | 0.907±0.066 | 29 | 112 | 35 | 35 |
| 28 | ZIndex' NPE3.0 | 0.901±0.081 | 30 | 123 | 29 | 31 |
| 29 | ZIndex' NPL3.0 | 0.895±0.072 | 31 | 121 | 38 | 37 |
| 30 | Q | 0.893±0.046 | 33 | 33 | 26 | 22 |
| 31 | ZIndex' NP3.0 | 0.89±0.076 | 32 | 130 | 39 | 39 |
| 32 | ZIndex' $\hat{A}$ | 0.89±0.055 | 25 | 267 | 15 | 16 |
| 33 | ZIndex' $\hat{M}M$ | 0.884±0.057 | 24 | 280 | 10 | 10 |
| 34 | CIndex' ICV3 | 0.876±0.095 | 34 | 4 | 30 | 29 |
| 35 | CIndex' $NP\hat{E}3.0$ | 0.849±0.125 | 39 | 7 | 41 | 41 |
| 36 | SWC0 $N\hat{O}$ | 0.841±0.065 | 36 | 126 | 32 | 32 |
| 37 | CIndex' NPE3.0 | 0.841±0.141 | 41 | 14 | 50 | 54 |
| 38 | CIndex' ICV2 | 0.838±0.11 | 45 | 1 | 40 | 38 |
| 39 | CIndex' IC3 | 0.837±0.115 | 53 | 2 | 42 | 40 |
| 40 | SWC0 $N\hat{O}V$ | 0.82±0.068 | 54 | 23 | 49 | 45 |
| 41 | SWC0 IC1 | 0.82±0.084 | 48 | 93 | 53 | 50 |
| 42 | ZIndex SP | 0.818±0.091 | 35 | 292 | 34 | 36 |
| 43 | SWC0 $\hat{P}C$ | 0.816±0.071 | 58 | 15 | 55 | 52 |
| 44 | SWC0 $N\hat{P}C$ | 0.816±0.071 | 59 | 16 | 56 | 53 |
| 45 | SWC0 NO | 0.816±0.088 | 49 | 109 | 58 | 59 |

***Table A.14:*** *Extended results for Table 2.5, where communities are well-separated with* $\mu = 0.1$.

| Near Optimal Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | ZIndex' IC2 | 0.826±0.227 | 2 | 10 | 4 | 6 |
| 2 | CIndex' ICV2 | 0.822±0.132 | 7 | 1 | 11 | 7 |
| 3 | ZIndex' IC3 | 0.821±0.232 | 1 | 16 | 5 | 9 |
| 4 | CIndex' ICV3 | 0.818±0.237 | 4 | 9 | 3 | 5 |
| 5 | ZIndex' ICV2 | 0.816±0.232 | 3 | 18 | 7 | 10 |
| 6 | ZIndex' $\hat{A}$ | 0.813±0.225 | 5 | 19 | 2 | 2 |
| 7 | CIndex' IC3 | 0.8±0.2 | 31 | 2 | 13 | 8 |
| 8 | ZIndex' A | 0.795±0.177 | 30 | 20 | 6 | 4 |
| 9 | ZIndex' $\hat{MM}$ | 0.794±0.221 | 9 | 33 | 1 | 1 |
| 10 | ZIndex' $\hat{NO}$ | 0.793±0.218 | 8 | 32 | 14 | 18 |
| 11 | ZIndex' ICV3 | 0.793±0.25 | 6 | 43 | 12 | 15 |
| 12 | ZIndex' $\hat{NOV}$ | 0.793±0.228 | 12 | 28 | 15 | 17 |
| 13 | ZIndex' $\hat{PC}$ | 0.792±0.227 | 17 | 26 | 18 | 22 |
| 14 | ZIndex' $\hat{NPC}$ | 0.792±0.227 | 18 | 27 | 19 | 23 |
| 15 | CIndex' NP2.0 | 0.791±0.132 | 16 | 8 | 79 | 121 |
| 16 | ZIndex' $\hat{NPL}2.0$ | 0.79±0.219 | 26 | 37 | 30 | 38 |
| 17 | ZIndex' $\hat{NPO}$ | 0.789±0.228 | 15 | 31 | 21 | 28 |
| 18 | ZIndex' NPE2.0 | 0.788±0.224 | 14 | 34 | 26 | 34 |
| 19 | ZIndex' $\hat{NP}2.0$ | 0.788±0.217 | 28 | 38 | 33 | 39 |
| 20 | ZIndex' $\hat{NPE}2.0$ | 0.787±0.225 | 13 | 39 | 20 | 27 |
| 21 | ZIndex' NPL2.0 | 0.786±0.221 | 32 | 36 | 38 | 46 |
| 22 | CIndex' $\hat{NPE}3.0$ | 0.784±0.14 | 10 | 21 | 57 | 95 |
| 23 | CIndex' $\hat{TO}$ | 0.784±0.138 | 22 | 4 | 41 | 41 |
| 24 | CIndex' NPL2.0 | 0.783±0.13 | 37 | 5 | 70 | 79 |
| 25 | CIndex' $\hat{NP}2.0$ | 0.778±0.129 | 46 | 3 | 55 | 45 |
| 26 | ZIndex' IC1 | 0.778±0.231 | 29 | 46 | 27 | 33 |
| 27 | ASWC1' ICV2 | 0.778±0.176 | 11 | 41 | 9 | 11 |
| 28 | CIndex' $\hat{NPE}2.0$ | 0.777±0.158 | 27 | 6 | 39 | 55 |
| 29 | ASWC1' ICV3 | 0.777±0.155 | 20 | 52 | 25 | 19 |
| 30 | ZIndex' $\hat{TO}$ | 0.776±0.226 | 34 | 47 | 29 | 40 |
| 31 | ZIndex' NP2.0 | 0.775±0.215 | 43 | 56 | 46 | 65 |
| 32 | ZIndex' TO | 0.774±0.228 | 36 | 48 | 31 | 42 |
| 33 | ZIndex' PC | 0.771±0.229 | 38 | 53 | 35 | 49 |
| 34 | ZIndex' NOV | 0.771±0.228 | 41 | 54 | 36 | 52 |
| 35 | ZIndex' ICV1 | 0.771±0.228 | 42 | 55 | 37 | 53 |
| 36 | CIndex' NPE2.0 | 0.771±0.133 | 25 | 22 | 69 | 111 |
| 37 | ZIndex' NO | 0.77±0.228 | 40 | 57 | 34 | 47 |
| 38 | ZIndex' M | 0.769±0.169 | 54 | 25 | 8 | 3 |
| 39 | CIndex' $\hat{NPO}$ | 0.765±0.126 | 50 | 11 | 56 | 48 |
| 40 | CIndex' TO | 0.763±0.158 | 39 | 17 | 83 | 126 |
| 41 | CIndex' IC2 | 0.76±0.122 | 61 | 12 | 40 | 14 |
| 42 | CIndex' $\hat{PC}$ | 0.759±0.129 | 55 | 13 | 60 | 43 |
| 43 | CIndex' $\hat{NPC}$ | 0.759±0.129 | 56 | 14 | 61 | 44 |
| 44 | CIndex' $\hat{NPL}2.0$ | 0.757±0.138 | 60 | 7 | 53 | 35 |
| 45 | CIndex' $\hat{NOV}$ | 0.754±0.131 | 58 | 15 | 65 | 50 |
| 46 | ASWC1' $\hat{NPE}3.0$ | 0.754±0.153 | 47 | 68 | 43 | 56 |
| 47 | ZIndex' $\hat{NPE}3.0$ | 0.754±0.241 | 24 | 77 | 42 | 69 |
| 48 | CIndex' NPE3.0 | 0.753±0.148 | 44 | 50 | 85 | 146 |
| 49 | ZIndex' NPE3.0 | 0.749±0.23 | 33 | 93 | 51 | 90 |
| 50 | CIndex' $\hat{NPL}3.0$ | 0.744±0.14 | 62 | 35 | 123 | 144 |
| | | ⋮ | | | | |
| 206 | SWC1' $\hat{NO}$ | 0.591±0.179 | 225 | 194 | 244 | 233 |
| 207 | Q | 0.589±0.161 | 222 | 198 | 138 | 110 |

***Table A.15:*** *Extended results for Table 2.5, where communities are well-separated with* $\mu = 0.1$.

| Medium Far Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | ZIndex' ICV2 | 0.741±0.177 | 4 | 231 | 22 | 22 |
| 2 | ZIndex' IC2 | 0.738±0.181 | 1 | 247 | 16 | 20 |
| 3 | ZIndex' IC3 | 0.728±0.188 | 5 | 252 | 18 | 21 |
| 4 | ZIndex' ICV3 | 0.721±0.177 | 8 | 258 | 21 | 23 |
| 5 | ZIndex' $\hat{PC}$ | 0.719±0.204 | 3 | 285 | 30 | 35 |
| 6 | ZIndex' $N\hat{P}C$ | 0.719±0.204 | 2 | 286 | 31 | 36 |
| 7 | CIndex' ICV3 | 0.713±0.151 | 28 | 21 | 33 | 27 |
| 8 | ZIndex' $N\hat{P}O$ | 0.709±0.205 | 7 | 278 | 32 | 38 |
| 9 | ZIndex' $\hat{TO}$ | 0.703±0.216 | 12 | 240 | 42 | 48 |
| 10 | ZIndex' TO | 0.702±0.217 | 14 | 239 | 45 | 53 |
| 11 | ZIndex' PC | 0.701±0.207 | 10 | 270 | 44 | 51 |
| 12 | ZIndex' $N\hat{O}V$ | 0.698±0.205 | 9 | 294 | 26 | 29 |
| 13 | CIndex' IC3 | 0.698±0.186 | 33 | 2 | 67 | 70 |
| 14 | CIndex' ICV2 | 0.696±0.188 | 31 | 4 | 62 | 63 |
| 15 | ZIndex' $NP\hat{E}2.0$ | 0.694±0.214 | 13 | 257 | 39 | 45 |
| 16 | ZIndex' NPE2.0 | 0.692±0.217 | 17 | 243 | 50 | 57 |
| 17 | ZIndex' NOV | 0.678±0.215 | 15 | 291 | 46 | 55 |
| 18 | ZIndex' ICV1 | 0.678±0.215 | 16 | 292 | 47 | 56 |
| 19 | ZIndex' $N\hat{O}$ | 0.673±0.213 | 6 | 310 | 29 | 33 |
| 20 | ZIndex' $N\hat{P}2.0$ | 0.666±0.207 | 21 | 272 | 58 | 64 |
| 21 | ZIndex' $NP\hat{L}2.0$ | 0.664±0.208 | 20 | 280 | 55 | 60 |
| 22 | ZIndex' IC1 | 0.663±0.221 | 11 | 308 | 38 | 43 |
| 23 | ZIndex' NPL2.0 | 0.66±0.205 | 25 | 264 | 64 | 78 |
| 24 | CIndex' IC2 | 0.658±0.199 | 49 | 1 | 108 | 105 |
| 25 | CIndex' $NP\hat{E}2.0$ | 0.654±0.223 | 45 | 5 | 103 | 108 |
| 26 | CIndex' $NP\hat{E}3.0$ | 0.647±0.218 | 35 | 35 | 79 | 86 |
| 27 | ZIndex' $NP\hat{E}3.0$ | 0.645±0.227 | 24 | 289 | 43 | 54 |
| 28 | CIndex' $\hat{PC}$ | 0.643±0.216 | 38 | 13 | 69 | 75 |
| 29 | CIndex' $N\hat{P}C$ | 0.643±0.216 | 39 | 14 | 70 | 76 |
| 30 | CIndex' NPE3.0 | 0.643±0.236 | 40 | 29 | 98 | 109 |
| 31 | ZIndex' NO | 0.642±0.235 | 18 | 311 | 54 | 58 |
| 32 | ZIndex' NP2.0 | 0.641±0.193 | 26 | 287 | 75 | 93 |
| 33 | CIndex' NPE2.0 | 0.64±0.221 | 37 | 22 | 85 | 91 |
| 34 | CIndex' $N\hat{P}O$ | 0.636±0.23 | 44 | 11 | 82 | 84 |
| 35 | ZIndex' NPE3.0 | 0.635±0.245 | 27 | 281 | 57 | 65 |
| 36 | CIndex' $N\hat{O}V$ | 0.63±0.225 | 48 | 6 | 102 | 103 |
| 37 | Q | 0.62±0.139 | 42 | 167 | 56 | 47 |

***Table A.16:*** *Extended results for Table 2.5, where communities are well-separated with $\mu = 0.1$.*

| Far Far Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | $\text{ARI}_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | ZIndex' ICV2 | 0.834±0.062 | 9 | 464 | 5 | 3 |
| 2 | ZIndex' IC3 | 0.832±0.06 | 7 | 469 | 4 | 2 |
| 3 | ZIndex' TO | 0.825±0.098 | 22 | 423 | 29 | 27 |
| 4 | ZIndex' ICV3 | 0.823±0.063 | 12 | 458 | 6 | 6 |
| 5 | ZIndex' $\hat{TO}$ | 0.823±0.096 | 18 | 446 | 27 | 25 |
| 6 | ZIndex' $N\hat{P}C$ | 0.822±0.083 | 2 | 502 | 11 | 10 |
| 7 | ZIndex' $\hat{PC}$ | 0.822±0.083 | 3 | 501 | 12 | 11 |
| 8 | ZIndex' PC | 0.817±0.09 | 11 | 479 | 23 | 19 |
| 9 | ZIndex' IC2 | 0.816±0.069 | 4 | 497 | 2 | 1 |
| 10 | ZIndex' $N\hat{P}O$ | 0.813±0.085 | 6 | 508 | 13 | 12 |
| 11 | ZIndex' NPE2.0 | 0.802±0.086 | 21 | 449 | 22 | 23 |
| 12 | ZIndex' $N\hat{O}V$ | 0.8±0.091 | 5 | 519 | 10 | 8 |
| 13 | ZIndex' $NP\hat{E}2.0$ | 0.796±0.088 | 16 | 478 | 17 | 15 |
| 14 | ZIndex' NOV | 0.789±0.096 | 13 | 494 | 20 | 21 |
| 15 | ZIndex' ICV1 | 0.789±0.096 | 14 | 495 | 21 | 22 |
| 16 | ZIndex' NPL2.0 | 0.779±0.106 | 25 | 471 | 32 | 28 |
| 17 | ZIndex' NP2.0 | 0.769±0.112 | 30 | 474 | 51 | 31 |
| 18 | ZIndex' IC1 | 0.768±0.105 | 8 | 530 | 16 | 14 |
| 19 | ZIndex' NO | 0.761±0.106 | 10 | 531 | 19 | 20 |
| 20 | ZIndex' $N\hat{P}2.0$ | 0.761±0.119 | 19 | 500 | 26 | 26 |
| 21 | ZIndex' $\hat{NO}$ | 0.76±0.119 | 1 | 546 | 8 | 9 |
| 22 | ZIndex' $NP\hat{L}2.0$ | 0.758±0.122 | 17 | 515 | 24 | 24 |
| 23 | ZIndex' $NP\hat{E}3.0$ | 0.733±0.109 | 50 | 467 | 31 | 30 |
| 24 | ZIndex' $NP\hat{L}3.0$ | 0.722±0.129 | 71 | 429 | 87 | 49 |
| 25 | ZIndex' $N\hat{P}3.0$ | 0.719±0.129 | 73 | 424 | 88 | 55 |
| 26 | ZIndex' NPE3.0 | 0.718±0.112 | 66 | 448 | 52 | 34 |
| 27 | ZIndex' NPL3.0 | 0.7±0.132 | 88 | 411 | 90 | 70 |
| 28 | ZIndex' NP3.0 | 0.689±0.137 | 89 | 410 | 92 | 75 |
| 29 | ZIndex' A | 0.646±0.132 | 45 | 525 | 18 | 16 |
| 30 | ZIndex' M | 0.638±0.151 | 31 | 537 | 9 | 4 |
| 31 | Q | 0.581±0.155 | 95 | 368 | 69 | 32 |
| 32 | ZIndex SP | 0.58±0.158 | 72 | 539 | 25 | 29 |
| 33 | CIndex' $NP\hat{E}3.0$ | 0.577±0.263 | 107 | 57 | 127 | 123 |

***Table A.17:*** *Extended results for Table 2.6, where communities are well-separated with* $\mu = 0.4$.

| Overall Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | Q | 0.854±0.039 | 11 | 1 | 4 | 2 |
| 2 | ZIndex' M | 0.839±0.067 | 2 | 5 | 1 | 1 |
| 3 | ZIndex' A | 0.813±0.071 | 4 | 11 | 3 | 3 |
| 4 | ZIndex' $\hat{MM}$ | 0.785±0.115 | 1 | 63 | 2 | 4 |
| 5 | ZIndex' $\hat{A}$ | 0.767±0.101 | 3 | 86 | 5 | 5 |
| 6 | ZIndex' $\hat{PC}$ | 0.748±0.19 | 5 | 108 | 7 | 7 |
| 7 | ZIndex' $\hat{NPC}$ | 0.748±0.19 | 6 | 109 | 8 | 8 |
| 8 | ZIndex' $\hat{NPO}$ | 0.745±0.191 | 7 | 110 | 9 | 9 |
| 9 | ZIndex' $\hat{TO}$ | 0.738±0.197 | 13 | 88 | 16 | 15 |
| 10 | ZIndex' $\hat{NOV}$ | 0.738±0.197 | 8 | 134 | 10 | 10 |
| 11 | ZIndex' $\hat{NPE2.0}$ | 0.738±0.193 | 14 | 81 | 17 | 17 |
| 12 | ZIndex' $\hat{NPL2.0}$ | 0.73±0.179 | 12 | 113 | 18 | 20 |
| 13 | ZIndex' TO | 0.724±0.216 | 18 | 100 | 24 | 23 |
| 14 | ZIndex' $\hat{NP2.0}$ | 0.719±0.19 | 16 | 135 | 26 | 25 |
| 15 | ASWC0 $\hat{PC}$ | 0.717±0.147 | 27 | 71 | 12 | 12 |
| 16 | ASWC0 $\hat{NPC}$ | 0.717±0.147 | 28 | 72 | 11 | 11 |
| 17 | ZIndex' NPE2.0 | 0.717±0.211 | 24 | 107 | 31 | 30 |
| 18 | ASWC0 $\hat{NPO}$ | 0.715±0.146 | 29 | 74 | 15 | 13 |
| 19 | SWC0 $\hat{NPC}$ | 0.713±0.143 | 35 | 45 | 19 | 18 |
| 20 | SWC0 $\hat{PC}$ | 0.713±0.143 | 36 | 46 | 20 | 19 |
| 21 | ZIndex' MD | 0.711±0.154 | 10 | 156 | 6 | 6 |
| 22 | SWC0 $\hat{NPO}$ | 0.709±0.145 | 38 | 49 | 25 | 22 |
| 23 | ZIndex' $\hat{NO}$ | 0.709±0.208 | 9 | 192 | 13 | 16 |
| 24 | ZIndex' NPL2.0 | 0.708±0.2 | 23 | 138 | 34 | 32 |
| 25 | ASWC0 $\hat{NOV}$ | 0.705±0.16 | 26 | 124 | 14 | 14 |
| 26 | ZIndex' IC2 | 0.702±0.213 | 17 | 165 | 23 | 24 |
| 27 | SWC0 $\hat{NOV}$ | 0.702±0.158 | 34 | 87 | 22 | 21 |
| 28 | ZIndex' PC | 0.702±0.229 | 19 | 157 | 27 | 27 |
| 29 | ZIndex' ICV2 | 0.7±0.217 | 25 | 153 | 30 | 31 |
| 30 | ZIndex' NOV | 0.7±0.231 | 20 | 167 | 28 | 28 |

***Table A.18:*** *Extended results for Table 2.6, where communities are well-separated with $\mu = 0.4$.*

| Near Optimal Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | ZIndex' M | 0.825±0.105 | 1 | 1 | 1 | 1 |
| 2 | ZIndex' A | 0.8±0.184 | 2 | 2 | 2 | 2 |
| 3 | ZIndex' $\hat{MM}$ | 0.768±0.166 | 3 | 4 | 3 | 3 |
| 4 | ZIndex' $\hat{A}$ | 0.76±0.192 | 4 | 6 | 4 | 4 |
| 5 | Q | 0.72±0.209 | 34 | 3 | 34 | 34 |
| 6 | ASWC0 $\hat{NPL}2.0$ | 0.719±0.248 | 22 | 8 | 5 | 5 |
| 7 | SWC0 $\hat{NPL}2.0$ | 0.718±0.247 | 23 | 9 | 6 | 6 |
| 8 | ZIndex' $\hat{NPE}2.0$ | 0.714±0.259 | 5 | 21 | 7 | 8 |
| 9 | ASWC0 SP | 0.71±0.286 | 28 | 5 | 29 | 26 |
| 10 | ZIndex' $\hat{NPL}2.0$ | 0.702±0.261 | 6 | 29 | 13 | 18 |
| 11 | SWC0 $\hat{NP}2.0$ | 0.702±0.242 | 26 | 12 | 16 | 12 |
| 12 | ASWC0 $\hat{NP}2.0$ | 0.702±0.242 | 27 | 13 | 17 | 13 |
| 13 | ZIndex' $\hat{NP}2.0$ | 0.702±0.26 | 7 | 27 | 14 | 20 |
| 14 | ZIndex' IC2 | 0.7±0.263 | 8 | 31 | 11 | 11 |
| 15 | SWC0 $\hat{A}$ | 0.696±0.187 | 42 | 7 | 23 | 15 |
| 16 | ZIndex' $\hat{NPO}$ | 0.695±0.27 | 9 | 39 | 12 | 14 |
| 17 | ZIndex' $\hat{TO}$ | 0.692±0.266 | 10 | 42 | 21 | 24 |
| 18 | ZIndex' $\hat{NOV}$ | 0.691±0.276 | 11 | 45 | 8 | 7 |
| 19 | ZIndex' $\hat{PC}$ | 0.691±0.283 | 12 | 43 | 10 | 10 |
| 20 | ZIndex' $\hat{NPC}$ | 0.691±0.283 | 13 | 44 | 9 | 9 |
| 21 | SWC0 $\hat{MM}$ | 0.69±0.178 | 45 | 10 | 26 | 19 |
| 22 | ZIndex' NPE2.0 | 0.689±0.272 | 14 | 40 | 25 | 28 |
| 23 | ZIndex' ICV2 | 0.688±0.265 | 17 | 38 | 22 | 23 |
| 24 | ZIndex' TO | 0.685±0.275 | 16 | 41 | 28 | 29 |
| 25 | ZIndex' NPL2.0 | 0.684±0.266 | 15 | 46 | 24 | 27 |
| 26 | ASWC0 $\hat{TO}$ | 0.684±0.222 | 41 | 14 | 44 | 47 |
| 27 | SWC0 $\hat{PC}$ | 0.683±0.29 | 29 | 33 | 18 | 16 |
| 28 | SWC0 $\hat{NPC}$ | 0.683±0.29 | 30 | 34 | 19 | 17 |
| 29 | SWC0 $\hat{NPO}$ | 0.683±0.288 | 31 | 32 | 20 | 21 |
| 30 | ZIndex' NP2.0 | 0.676±0.275 | 18 | 57 | 30 | 31 |

***Table A.19:*** *Extended results for Table 2.6, where communities are well-separated with $\mu = 0.4$.*

| Medium Far Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | $\text{ARI}_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | Q | 0.578±0.124 | 106 | 22 | 3 | 1 |
| 2 | CIndex' $N\hat{P}C$ | 0.522±0.146 | 154 | 12 | 78 | 69 |
| 3 | CIndex' $\hat{P}C$ | 0.521±0.146 | 155 | 13 | 79 | 70 |
| 4 | CIndex' $N\hat{P}O$ | 0.519±0.142 | 176 | 5 | 120 | 100 |
| 5 | CIndex' $N\hat{O}V$ | 0.501±0.14 | 209 | 4 | 142 | 135 |
| 6 | ZIndex' M | 0.498±0.199 | 4 | 364 | 2 | 2 |
| 7 | CIndex' IC2 | 0.492±0.146 | 227 | 9 | 176 | 173 |
| 8 | CIndex' ICV2 | 0.483±0.193 | 149 | 79 | 119 | 115 |
| 9 | CIndex' IC3 | 0.478±0.191 | 187 | 43 | 148 | 146 |
| 10 | CIndex' TO | 0.478±0.175 | 179 | 31 | 204 | 203 |
| 11 | CIndex' $N\hat{O}$ | 0.475±0.126 | 295 | 3 | 227 | 210 |
| 12 | SWC1 ICV3 | 0.475±0.162 | 148 | 92 | 32 | 22 |
| 13 | SWC1 $NP\hat{L}2.0$ | 0.466±0.092 | 239 | 29 | 132 | 113 |
| 14 | SWC1 $N\hat{P}2.0$ | 0.466±0.105 | 219 | 39 | 92 | 80 |
| 15 | CIndex' $NP\hat{E}2.0$ | 0.465±0.152 | 291 | 6 | 308 | 276 |
| 16 | CIndex' $T\hat{O}$ | 0.464±0.177 | 253 | 15 | 254 | 236 |
| 17 | ASWC1 $NP\hat{L}2.0$ | 0.461±0.088 | 242 | 32 | 133 | 112 |
| 18 | ASWC1 $N\hat{P}2.0$ | 0.458±0.099 | 224 | 47 | 94 | 81 |
| 19 | SWC1 IC3 | 0.457±0.191 | 118 | 134 | 12 | 8 |
| 20 | SWC1' IC3 | 0.456±0.134 | 204 | 60 | 85 | 65 |
| 21 | SWC1' ICV3 | 0.454±0.134 | 214 | 61 | 83 | 63 |
| 22 | ASWC1' $N\hat{P}C$ | 0.453±0.12 | 208 | 54 | 71 | 49 |
| 23 | SWC1 $\hat{P}C$ | 0.45±0.179 | 142 | 106 | 21 | 13 |
| 24 | SWC1 $N\hat{P}C$ | 0.45±0.179 | 143 | 107 | 20 | 12 |
| 25 | SWC1 $N\hat{P}O$ | 0.45±0.176 | 138 | 109 | 28 | 16 |
| 26 | SWC1 $NP\hat{E}2.0$ | 0.446±0.112 | 296 | 23 | 144 | 129 |
| 27 | SWC1' IC2 | 0.444±0.148 | 205 | 65 | 128 | 94 |
| 28 | CIndex' NPE2.0 | 0.444±0.163 | 215 | 48 | 318 | 288 |
| 29 | SWC1 ICV2 | 0.444±0.222 | 159 | 114 | 37 | 30 |
| 30 | SWC1' $NP\hat{L}2.0$ | 0.444±0.134 | 288 | 27 | 159 | 147 |

***Table A.20:*** *Extended results for Table 2.6, where communities are well-separated with $\mu = 0.4$.*

| Far Far Results | | | | | | |
|---|---|---|---|---|---|---|
| Rank | Criterion | ARI$_{corr}$ | Rand | Jaccard | NMI | AMI |
| 1 | ZIndex’ $\hat{PC}$ | 0.527±0.169 | 61 | 501 | 5 | 4 |
| 2 | ZIndex’ $\hat{NPC}$ | 0.527±0.169 | 62 | 502 | 6 | 5 |
| 3 | Q | 0.523±0.192 | 128 | 73 | 93 | 25 |
| 4 | ZIndex’ M | 0.522±0.121 | 77 | 465 | 8 | 2 |
| 5 | ZIndex’ $\hat{NPO}$ | 0.518±0.168 | 63 | 504 | 10 | 6 |
| 6 | ZIndex’ $\hat{NOV}$ | 0.515±0.166 | 60 | 518 | 11 | 7 |
| 7 | ZIndex’ $\hat{TO}$ | 0.489±0.171 | 78 | 485 | 15 | 9 |
| 8 | ZIndex’ $\hat{NPE2.0}$ | 0.481±0.168 | 79 | 491 | 24 | 14 |
| 9 | ZIndex’ $\hat{MM}$ | 0.48±0.15 | 30 | 553 | 2 | 3 |
| 10 | ZIndex’ $\hat{NO}$ | 0.48±0.17 | 43 | 552 | 7 | 8 |
| 11 | ZIndex’ PC | 0.477±0.157 | 76 | 503 | 26 | 18 |
| 12 | ZIndex’ NOV | 0.475±0.16 | 70 | 511 | 29 | 21 |
| 13 | ZIndex’ ICV1 | 0.475±0.16 | 71 | 512 | 30 | 22 |
| 14 | ZIndex’ TO | 0.475±0.176 | 89 | 477 | 21 | 12 |
| 15 | ZIndex’ NPE2.0 | 0.472±0.166 | 91 | 475 | 40 | 23 |
| 16 | ZIndex’ IC1 | 0.472±0.167 | 59 | 545 | 14 | 13 |
| 17 | ZIndex’ A | 0.456±0.11 | 92 | 481 | 20 | 11 |
| 18 | ZIndex’ NO | 0.451±0.166 | 64 | 547 | 25 | 27 |
| 19 | ZIndex’ NPL2.0 | 0.451±0.175 | 81 | 494 | 37 | 28 |
| 20 | ZIndex’ IC2 | 0.45±0.179 | 67 | 538 | 19 | 16 |
| 21 | ZIndex’ $\hat{NP2.0}$ | 0.449±0.167 | 75 | 514 | 27 | 20 |
| 22 | ZIndex’ $\hat{NPL2.0}$ | 0.448±0.163 | 69 | 523 | 23 | 17 |
| 23 | ZIndex SP | 0.444±0.175 | 90 | 496 | 61 | 32 |
| 24 | ZIndex’ ICV2 | 0.443±0.176 | 82 | 507 | 32 | 24 |
| 25 | ZIndex’ MD | 0.439±0.146 | 66 | 542 | 1 | 1 |
| 26 | ASWC0 $\hat{PC}$ | 0.438±0.17 | 100 | 455 | 57 | 29 |
| 27 | ASWC0 $\hat{NPC}$ | 0.438±0.17 | 101 | 456 | 58 | 30 |
| 28 | ZIndex’ NP2.0 | 0.434±0.179 | 87 | 500 | 56 | 35 |
| 29 | ASWC0 $\hat{NPO}$ | 0.428±0.171 | 103 | 459 | 69 | 34 |
| 30 | SWC0 $\hat{PC}$ | 0.427±0.19 | 115 | 420 | 94 | 41 |

# Appendix B

# Appendix of Chapter 3

# B.1 Proofs

## B.1.1 Proof of Proposition 3.3.1:

From the definition of Variation of information we have:

$$VI(U,V) = H(U) + H(V) - 2I(U,V) = 2H(U,V) - H(U) - H(V) = \mathbf{H(V|U)} + \mathbf{H(U|V)}$$

On the other hand, from Equation 3.2 we see that $RI$ is proportional to:

$$\begin{aligned} RI(U,V) &\propto \frac{1}{n^2-n}\Big(\sum_{i=1}^{k}[\sum_{j=1}^{r} n_{ij}^2 - (\sum_{j=1}^{r} n_{ij})^2] + \sum_{j=1}^{r}[\sum_{i=1}^{k} n_{ij}^2 - (\sum_{i=1}^{k} n_{ij})^2]\Big) \\ &\overset{*}{\propto} \sum_{i=1}^{k}[E_j(n_{ij}^2) - E_j(n_{ij})^2] + \sum_{j=1}^{r}[E_i(n_{ij}^2) - E_i(n_{ij})^2] \\ &\overset{*}{\propto} \sum_{i=1}^{k} Var_j(n_{ij}) + \sum_{j=1}^{r} Var_i(n_{ij}) \quad \overset{**}{\propto} \quad \mathbf{Var(V|U)} + \mathbf{Var(U|V)} \end{aligned}$$

($*$) $E_j/Var_j$ shows the average/variance of values in the $j^{th}$ column of the contingency table.
($**$) The RI is in fact proportional to the average variance of rows/columns values in the contingency table, which we denote by conditional variance. For other forms of conditional variance for categorical data see Light and Margolin [90].

□

## B.1.2 Proof of Corollary 3.3.2:

We first show that $0 \leq \mathcal{D}_{\varphi}^{\eta}(U||V)$ which also results in the lower bound 0 for $\mathcal{D}_{\varphi}^{\eta}(U,V)$ since, $\mathcal{D}_{\varphi}^{\eta}(U,V) = \mathcal{D}_{\varphi}^{\eta}(U||V) + \mathcal{D}_{\varphi}^{\eta}(V||U)$. From the superadditivity of $\varphi$ we have:

$$\sum_{u\in U} \varphi(\eta_{uv}) \leq \varphi(\sum_{u\in U} \eta_{uv}) \implies \sum_{v\in V}\left[\varphi(\sum_{u\in U}\eta_{uv}) - \sum_{u\in U}\varphi(\eta_{uv})\right] \geq 0 \implies \mathcal{D}_{\varphi}^{\eta}(\mathbf{U}||\mathbf{V}) \geq \mathbf{0}$$

Similarly for the upper bound, from positivity and super-additivity we get respectively:

$$\mathcal{D}_{\varphi}^{\eta}(U||V) = \sum_{v\in V}\varphi(\sum_{u\in U}\eta_{uv}) - \sum_{v\in V}\sum_{u\in U}\varphi(\eta_{uv}) \leq \sum_{v\in V}\varphi(\sum_{u\in U}\eta_{uv}) \leq \varphi(\sum_{v\in V}\sum_{u\in U}\eta_{uv})$$

□

### B.1.3 Proof of Identity 3.3.3:

The proof is elementary, if we write the definition for $\varphi = x\log x$, we get:

$$\begin{aligned}
\mathcal{ND}^{|\cap|}_{x\log x}(U,V) &= \frac{\sum_{v\in V}\sum_{u\in U}|u\cap v|\,[\log(\sum_{u\in U}|u\cap v|)-\log(|u\cap v|)]}{(\sum_{v\in V}\sum_{u\in U}|u\cap v|)\log(\sum_{v\in V}\sum_{u\in U}|u\cap v|)} \\
&+ \frac{\sum_{u\in U}\sum_{v\in V}|u\cap v|\,[\log(\sum_{v\in V}|u\cap v|)-\log(|u\cap v|)]}{(\sum_{u\in U}\sum_{v\in V}|u\cap v|)\log(\sum_{u\in U}\sum_{v\in V}|u\cap v|)} \\
&\overset{*}{=} \frac{\sum_j^r\sum_i^k n_{ij}\left[\log(\sum_i^k n_{ij})+\log(\sum_j^r n_{ij})-2\log(n_{ij})\right]}{(\sum_i^k\sum_j^r n_{ij})\log(\sum_i^k\sum_j^r n_{ij})} \\
&\overset{**}{=} \frac{1}{\log n}\sum_j^r\sum_i^k\frac{n_{ij}}{n}\log(\frac{n_{i.}n_{.j}}{n_{ij}^2}) = \frac{VI(U,V)}{\log n}
\end{aligned}$$

$(*)$ slight change of notation, *i.e.*, from $\sum_{u\in U}$ to $\sum_i^k$, $\sum_{v\in V}$ to $\sum_j^r$ and $|u\cap v|$ to $n_{ij}$.
$(**)$ assuming disjoint covering partitionings: $\sum_i^k\sum_j^r n_{ij} = n$, $\sum_i^k n_{ij} = n_{.j}$ and $\sum_j^r n_{ij} = n_{i.}$.

□

### B.1.4 Proof of Identity 3.3.4:

Similar to the previous proof from the definition we derive:

$$\begin{aligned}
\mathcal{ND}^{|\cap|}_{\binom{x}{2}}(U,V) &\overset{*}{=} \frac{\sum_j^r\left[(\sum_i^k n_{ij})^2-\sum_i^k n_{ij}^2\right]+\sum_i^k\left[(\sum_j^r n_{ij})^2-\sum_j^r n_{ij}^2\right]}{(\sum_i^k\sum_j^r n_{ij})^2-\sum_i^k\sum_j^r n_{ij}} \\
&\overset{**}{=} \frac{1}{n^2-n}[\sum_j^r(n_{.j})^2+\sum_i^k(n_{i.})^2-2\sum_j^r\sum_i^k n_{ij}^2] = 1-RI(U,V)
\end{aligned}$$

$(*)$, $(**)$ same as previous proof.

□

### B.1.5 Proof of Identity 3.3.5 and 3.3.6:

$$\begin{aligned}
\mathcal{AD}^{\eta}_{\varphi} &= \frac{\sum_{v\in V}\varphi(\eta_{.v})+\sum_{u\in U}\varphi(\eta_{u.})-2\sum_{v\in V}\sum_{u\in U}\varphi(\eta_{uv})}{\sum_{v\in V}\varphi(\eta_{.v})+\sum_{u\in U}\varphi(\eta_{u.})-2\sum_{u\in U}\sum_{v\in V}\varphi\left(\frac{\eta_{.v}\eta_{u.}}{\sum_{u\in U}\sum_{v\in V}\eta_{uv}}\right)} \\
\Rightarrow 1-\mathcal{AD}^{\eta}_{\varphi}(U,V) &= \frac{\sum_{v\in V}\sum_{u\in U}\varphi(\eta_{uv})-\sum_{u\in U}\sum_{v\in V}\varphi\left(\frac{\eta_{.v}\eta_{u.}}{\sum_{u\in U}\sum_{v\in V}\eta_{uv}}\right)}{\frac{1}{2}\left[\sum_{v\in V}\varphi(\eta_{.v})+\sum_{u\in U}\varphi(\eta_{u.})\right]-\sum_{u\in U}\sum_{v\in V}\varphi\left(\frac{\eta_{.v}\eta_{u.}}{\sum_{u\in U}\sum_{v\in V}\eta_{uv}}\right)}
\end{aligned}$$

This formula resembles the adjuctment for chance in Equation 3.4, where the measure being adjusted is $\sum_{v\in V}\sum_{u\in U}\varphi(\eta_{uv})$, the upper bound used for it is $\frac{1}{2}[\sum_{v\in V}\varphi(\eta_{.v})+\sum_{u\in U}\varphi(\eta_{u.})]$, and the expectation is defined as:

$$E[\sum_{v\in V}\sum_{u\in U}\varphi(\eta_{uv})] = \sum_{u\in U}\sum_{v\in V}\varphi\left(\frac{\eta_{.v}\eta_{u.}}{\sum_{u\in U}\sum_{v\in V}\eta_{uv}}\right)$$

Now if we have $\varphi(xy) = \varphi(x)\varphi(y)$, which is true for $\varphi(x) = x^2$, we get:

$$E[\sum_{v\in V}\sum_{u\in U}\varphi(\eta_{uv})] = \sum_{u\in U}\sum_{v\in V}\frac{\varphi(\eta_{.v})\varphi(\eta_{u.})}{\varphi(\sum_{u\in U}\sum_{v\in V}\eta_{uv})} = \frac{\sum_{v\in V}\varphi(\eta_{.v})\sum_{u\in U}\varphi(\eta_{u.})}{\varphi(\sum_{u\in U}\sum_{v\in V}\eta_{uv})}$$

Using this expecation, if we substitute $\varphi = x^2$ we get the $ARI'$ of Equation 3.6, and using the $\varphi = \binom{x}{2}$ and the later reformulation of $E$, we get the original $ARI$ of Equation 3.5, as:

$$1 - \mathcal{AD}^{|\cap|}_{\binom{x}{2}}(U,V) = \frac{\sum\limits_{v\in V}\sum\limits_{u\in U}\binom{|u\cap v|}{2} - E(\sum\limits_{v\in V}\sum\limits_{u\in U}\binom{|u\cap v|}{2})}{\frac{1}{2}\left[\sum\limits_{v\in V}\binom{\sum\limits_{u\in U}|u\cap v|}{2} + \sum\limits_{u\in U}\binom{\sum\limits_{v\in V}|u\cap v|}{2}\right] - E(\sum\limits_{v\in V}\sum\limits_{u\in U}\binom{|u\cap v|}{2})}$$

$$where \quad E(\sum_{v\in V}\sum_{u\in U}\binom{|u\cap v|}{2}) = \frac{\sum\limits_{v\in V}\binom{\sum\limits_{u\in U}|u\cap v|}{2}\sum\limits_{u\in U}\binom{\sum\limits_{v\in V}|u\cap v|}{2}}{\binom{n}{2}}$$

$$\Rightarrow 1 - \mathcal{AD}^{|\cap|}_{\binom{x}{2}}(U,V) \overset{*,**}{=} \frac{\sum_j^r\sum_i^k\binom{n_{ij}}{2} - \sum_j^r\binom{n_{.j}}{2}\sum_i^k\binom{n_{i.}}{2}/\binom{n}{2}}{\frac{1}{2}\left[\sum_j^r\binom{n_{.j}}{2} + \sum_i^k\binom{n_{i.}}{2}\right] - \sum_j^r\binom{n_{.j}}{2}\sum_i^k\binom{n_{i.}}{2}/\binom{n}{2}} = ARI(U,V)$$

$(*)$, $(**)$ same as proof of identity 1. On the other hand for the $NMI$, we have:

$$1 - \mathcal{AD}^{|\cap|}_{x\log x}(U,V) = \frac{\sum\limits_{v\in V}\sum\limits_{u\in U}n_{uv}\log n_{uv} - E(\sum\limits_{v\in V}\sum\limits_{u\in U}n_{uv}\log n_{uv})}{\frac{1}{2}\left[\sum\limits_{v\in V}n_{.v}\log n_{.v} + \sum\limits_{u\in U}n_{u.}\log n_{u.}\right] - E(\sum\limits_{v\in V}\sum\limits_{u\in U}n_{uv}\log n_{uv})}$$

$$where\ E(\sum_{v\in V}\sum_{u\in U}n_{uv}\log n_{uv}) = \sum_{u\in U}\sum_{v\in V}\left(\frac{\eta_{.v}\eta_{u.}}{\sum_{u\in U}\sum_{v\in V}\eta_{uv}}\right)\log\left(\frac{\eta_{.v}\eta_{u.}}{\sum_{u\in U}\sum_{v\in V}\eta_{uv}}\right)$$

$$
\begin{aligned}
\Rightarrow 1-\mathcal{AD}^{|\cap|}_{x\log x}(U,V) &\overset{*,**}{=} \frac{\sum_j^r\sum_i^k n_{ij}\log n_{ij}-\sum_i^k\sum_j^r \frac{n_{.j}n_{i.}}{n}\log\frac{n_{.j}n_{i.}}{n}}{\frac{1}{2}\left[\sum_j^r n_{.j}\log n_{.j}+\sum_i^k n_{i.}\log n_{i.}\right]-\sum_i^k\sum_j^r\frac{n_{.j}n_{i.}}{n}\log\frac{n_{.j}n_{i.}}{n}}\\
&= \frac{n\sum_j^r\sum_i^k\frac{n_{ij}}{n}\log\frac{n_{ij}}{n}+n\log n-\sum_i^k\sum_j^r\frac{n_{.j}n_{i.}}{n}[\log\frac{n_{.j}}{n}+\log\frac{n_{i.}}{n}+\log n]}{\frac{n}{2}\left[\sum_j^r\frac{n_{.j}}{n}\log\frac{n_{.j}}{n}+\sum_i^k\frac{n_{i.}}{n}\log\frac{n_{i.}}{n}+2\log n\right]-\sum_i^k\sum_j^r\frac{n_{.j}n_{i.}}{n}[\log\frac{n_{.j}}{n}+\log\frac{n_{i.}}{n}+\log n]}\\
&= \frac{-H(U,V)+\log n-\sum_i^k\frac{n_{i.}}{n}\sum_j^r\frac{n_{.j}}{n}\log\frac{n_{.j}}{n}+\sum_i^k\frac{n_{.j}}{n}-\sum_j^r\frac{n_{i.}}{n}\log\frac{n_{i.}}{n}-\sum_i^k\sum_j^r\frac{n_{.j}n_{i.}}{n^2}\log n}{\frac{1}{2}[-H(U)-H(V)]+\log n+\sum_i^k\frac{n_{i.}}{n}H(V)+\sum_i^k\frac{n_{.j}}{n}H(U)-\log n}\\
&= \frac{-H(U,V)+H(V)+H(U)}{-\frac{1}{2}[H(U)+H(V)]+H(V)+H(U)} = \frac{I(U,V)}{\frac{1}{2}[H(U)+H(V)]} = NMI_{sum}(U,V)
\end{aligned}
$$

$(*)$, $(**)$ same as proof of identity 1.

□

### B.1.6 Proof of Identity 3.4.1 and 3.4.2:

First we prove that in general cases we have:

$$||UU^T-VV^T||_F^2 = ||U^TU||_F^2+||V^TV||_F^2-2||U^TV||_F^2$$

where $||.||_F^2$ is squared Frob norm. This holds since we have:

$$
\begin{aligned}
||UU^T-VV^T||_F^2 &= \sum_{ij}(UU^T-VV^T)_{ij}^2\\
&= \sum_{ij}(UU^T)_{ij}^2+\sum_{ij}(VV^T)_{ij}^2-2\sum_{ij}(UU^T)_{ij}(VV^T)_{ij}\\
&= ||UU^T||_F^2+||VV^T||_F^2-2|UU^T\circ VV^T|
\end{aligned}
$$

Where the $\circ$ is element-wise matrix product, a.k.a. hadamard product, and $|.|$ is sum of all elements in the matrix[1]. The proof is complete with showing:

$$
\begin{aligned}
|UU^T\circ VV^T| = tr((UU^T)^TVV^T) = tr(V^TUU^TV) = tr((U^TV)^TU^TV) = ||U^TV||_F^2\\
||UU^T||_F^2 = tr((UU^T)^TUU^T) = tr(U^TUU^TU) = tr((U^TU)^TU^TU) = ||U^TU||_F^2
\end{aligned}
$$

Now, we can prove the identities for the cases of disjoint hard clusters, using the notation $n_{ij} = (U^TV)_{ij}$, we have $||U^TV||_F^2 = \sum_{ij}n_{ij}^2$ and:

$$||U^TU||_F^2 = \sum_{ij} < U_{.i}, U_{.j} >^2 = \sum_{ij}(\sum_k u_{ki}u_{kj})^2 \overset{*}{=} \sum_i(\sum_k u_{ki}^2)^2 \overset{**}{=} \sum_i(\sum_k u_{ki})^2 \overset{***}{=} \sum_i n_{i.}^2$$

[1] This equality is also useful in the implementation to improve the scalability.

$(*)$ with assumption that clusters are disjoint, $u_{ki}u_{kj}$ is only non-zero iff $i = j$
$(**)$ with the assumption that memberships are hard, $u_{ki}$ is either 0 or 1, therefore $u_{ki} = u_{ki}^2$
$(***)$ marginals of $N$ give cluster sizes in $U$ and $V$, *i.e.*, $n_{i.} = \sum_j n_{ij} = \sum_k u_{ki} = |V_i|$

Therefore for disjoint hard clusters we get:

$$||UU^T - VV^T||_F^2 = \sum_i n_{i.}^2 + \sum_j n_{.j}^2 - 2\sum_{ij} n_{ij}^2$$

The *RI* normalization assumes that all pairs are in disagreement, *i.e.*, $|\mathbf{1}_{n\times n}| = n^2$, since $max(UU^T) = 1$ and, $max(VV^T) = 1$. The *ARI* normalization compares Δ to the difference where the two random variable of $UU_{ij}^T$ and $VV_{ij}^T$ are independent, in which case we would have:

$$E(UU_{ij}^T VV_{ij}^T) = E((UU^T)_{ij})E((VV^T)_{ij})$$

which is calculated by:

$$\frac{\sum_{ij}((UU^T)_{ij}(VV^T)_{ij})}{n^2} = \frac{\sum_{ij}(UU^T)_{ij}}{n^2}\frac{\sum_{ij}(VV^T)_{ij}}{n^2}$$

Since $\Delta = ||UU^T - VV^T||_F^2 = ||UU^T||_F^2 + ||VV^T||_F^2 - 2Sum(UU^T \circ VV^T)$, we have $ARI = 0$ or normalized distance 1, *i.e.*, agreement no better than chance, when this independence condition holds, *i.e.*, :

$$Sum(UU^T \circ VV^T) = \frac{|UU^T||VV^T|}{n^2}$$

□

### B.1.7 Proof of Identity 3.5.1 :

ARI of Equation 3.6 for two clusterings with $k$ and $r$ disjoint clusters is formulated as:

$$ARI = \frac{\sum_{p=1}^{k}\sum_{q=1}^{r} n_{pq}^2 - (\sum_{p=1}^{k} n_{p.}^2)(\sum_{q=1}^{r} n_{.q}^2)/n^2}{\frac{1}{2}[\sum_{p=1}^{k} n_{p.}^2 + \sum_{q=1}^{r} n_{.q}^2] - (\sum_{p=1}^{k} n_{p.}^2)(\sum_{q=1}^{r} n_{.q}^2)/n^2}$$

Where $n_{pq}$ measures the overlap between the $p^{th}$ cluster in the first clustering and the $q^{th}$ cluster in the second clustering, which we respectively denote by $u$ and $v$, for short. Hence we have $n_{pq} = |u \cap v| = \sum_{i \in u\cap v} 1$. When clusterings are disjoint, we have $u_{\leftarrow i} = 1$ iff $i \in u$ and zero otherwise; therefore we can write: $n_{pq} = \sum_{i\in u\cap v} u_{\leftarrow i} \times v_{\leftarrow i} = o_{uv}$. Therefore for disjoint clusters, and when $\varphi(x) = x^2$, we have:

$$\sum_{p=1}^{k}\sum_{q=1}^{r} n_{pq}^2 = \sum_{u\in U}\sum_{v\in V} o_{uv}^2 = \mathcal{O}_{UV}$$

On the other hand, $n_{p.} = \sum_{q=1}^{r} n_{pq}$, which represent the size of cluster $p$. In case of disjoint covering clusters, which is the assumption of the ARI, we can see that $n_{p.} = \sum_{i\in u} u_{\leftarrow i} = o_u$. Hence, with $\varphi(x) = x^2$, we also have:

$$(\sum_{p=1}^{k} n_{p.}^2)(\sum_{q=1}^{r} n_{.q}^2)/n^2 = \sum_{p=1}^{k}\sum_{q=1}^{r}(n_{p.}n_{.q}/n)^2 = \sum_{u}\sum_{v}(o_u o_v/n)^2 = \mathcal{E}_{UV}$$

Furthermore, since clusters are disjoint, $o_{uu'} = o_u$ iff $u = u'$ and zero otherwise, *i.e.*, disjoint clusters only have overlap with themselves which is equal to their size, we can further write:

$$\sum_{p=1}^{k} n_{p.}^2 = \sum_{u\in U} o_u^2 = \sum_{u\in U}\sum_{u'\in U} o_{uu'}^2 = \mathcal{O}_{UU}$$

Substituting these terms in the *ARI* of Equation 3.6 results in the *CRI* in Equation 3.25, which concludes the proof.

□

### B.1.8 Proof of Identity 3.5.2 :

$NMI_{sum}$ of Equation 3.9 is defined for disjoint clusters as:

$$NMI_{sum}(U,V) = \frac{I(U,V)}{\frac{1}{2}\left[H(U) + H(V)\right]} = \frac{H(U,V) - H(V) - H(U)}{\frac{1}{2}\left[H(U) + H(V)\right] - H(V) - H(U)}$$

Where $H(U)$ denotes the entropy of clustering $U$; $I(U,V)$ denotes the mutual information between two clustering $U$ and $V$, and $H(U,V)$ denotes their joint entropy. Where the entropy, mutual information, and joint entropy of clusterings are calculated based on their confusion matrix, *i.e.*, their pairwise cluster overlaps, which is assumed to denote the joint probability distribution of, memberships of data points in, clustering $U$ and $V$. More formally we have:

$$H(U,V) = -\sum_{i=1}^{k}\sum_{j=1}^{r}\frac{n_{ij}}{n}\log(\frac{n_{ij}}{n})$$

Similar to the ARI, $n_{ij}$ denotes the overlap between the $i^{th}$ cluster in $U$ and the $j^{th}$ cluster in $V$, which we call $u$ and $v$ for short, and then write:

$$\begin{aligned} H(U,V) &= -\sum_{u\in U}\sum_{v\in V}\frac{o_{uv}}{n}\log(\frac{o_{uv}}{n}) = -\frac{1}{n}\sum_{u\in U}\sum_{v\in V}o_{uv}(\log(o_{uv})-\log(n)) \\ &= -\frac{1}{n}\sum_{u\in U}\sum_{v\in V}o_{uv}\log(o_{uv}) + \frac{\log(n)}{n}\sum_{u\in U}\sum_{v\in V}o_{uv} \end{aligned}$$

Since clusters are covering and disjoint we have $\sum\limits_{u\in U}\sum\limits_{v\in V}o_{uv} = n$, hence we get:

$$H(U,V) = -\frac{1}{n}\sum_{u\in U}\sum_{v\in V}o_{uv}\log(o_{uv}) + \log(n)$$

Similarly, for disjoint covering clusters we also have $\sum_{u\in U} o_u = n$, and we can rewrite $H(U)$ as:

$$H(U) = -\sum_{i=1}^{k}\frac{n_{i.}}{n}\log(\frac{n_{i.}}{n}) = -\sum_{u\in U}\frac{o_u}{n}\log(\frac{o_u}{n}) = -\frac{1}{n}\sum_{u\in U}o_u\log(o_u) + \log(n)$$

Since when clusters are disjoint we have $o_{uu'} = o_u$ iff $u = u'$ and zero otherwise, $H(U)$ is further equivalent to (when $0\log 0 = 0$):

$$H(U) = -\frac{1}{n}\sum_{u\in U}o_u\log(o_u) + \log(n) = -\frac{1}{n}\sum_{u\in U}\sum_{u'\in U}o_{uu'}\log(o_{uu'}) + \log(n)$$

Therefore we can write:

$$H(U) + H(V) = -\frac{1}{n}(\sum_{u,u'\in U}o_{uu'}\log(o_{uu'}) + \sum_{v,v'\in V}o_{vv'}\log(o_{vv'})) + 2\log(n)$$

On the other hand, for $H(V) + H(U)$ we also have:

$$H(U) + H(V) = -\sum_{u\in U}\frac{o_u}{n}\log(\frac{o_u}{n}) - \sum_{v\in V}\frac{o_v}{n}\log(\frac{o_v}{n})$$

We further show$^{(*)}$ that for disjoint covering clusters, we have:

$$-\sum_{u\in U}\frac{o_u}{n}\log(\frac{o_u}{n}) - \sum_{v\in V}\frac{o_v}{n}\log(\frac{o_v}{n}) = -\frac{1}{n}\sum_{u\in U}\sum_{v\in V}\frac{o_u o_v}{n}\log(\frac{o_u o_v}{n}) + \log(n)$$

By substituting these terms in the $NMI_{sum}$ of Equation 3.9, we get the $CMI$ formula in Equation 3.26 which concludes the proof.

□

$$\begin{aligned}
&(*) -\frac{1}{n}\sum_{u\in U}\sum_{v\in V}\frac{o_u o_v}{n}\log(\frac{o_u o_v}{n}) + \log(n)\\
&= -\frac{1}{n}\sum_{u\in U}\sum_{v\in V}\frac{o_u o_v}{n}\left[\log(\frac{o_u}{n}) + \log(\frac{o_v}{n}) + \log(n)\right] + \log(n)\\
&= -\frac{1}{n}\left[\sum_{u\in U}\sum_{v\in V}\frac{o_u o_v}{n}\log(\frac{o_u}{n}) + \sum_{u\in U}\sum_{v\in V}\frac{o_u o_v}{n}\log(\frac{o_v}{n}) + \sum_{u\in U}\sum_{v\in V}\frac{o_u o_v}{n}\log(n)\right] + \log(n)\\
&= -\frac{1}{n}\left[(\sum_{v\in V} o_v)\sum_{u\in U}\frac{o_u}{n}\log(\frac{o_u}{n}) + (\sum_{u\in U} o_u)\sum_{v\in V}\frac{o_v}{n}\log(\frac{o_v}{n}) + \frac{\log(n)}{n}\sum_{u\in U}\sum_{v\in V} o_u o_v\right] + \log(n)\\
&= -\frac{1}{n}\left[n\sum_{u\in U}\frac{o_u}{n}\log(\frac{o_u}{n}) + n\sum_{v\in V}\frac{o_v}{n}\log(\frac{o_v}{n}) + \frac{\log(n)}{n}(\sum_{u\in U} o_u)(\sum_{v\in V} o_v)\right] + \log(n)\\
&= -\sum_{u\in U}\frac{o_u}{n}\log(\frac{o_u}{n}) - \sum_{v\in V}\frac{o_v}{n}\log(\frac{o_v}{n}) - \log(n) + \log(n)
\end{aligned}$$

### B.1.9 Proof of Identity 3.5.3 :

First, since we have $(UU^T)_{ij} = U_{i.}U^T_{.j} = U_{i.}U_{j.} = \sum_{p=1}^{k} U_{ip}U_{jp}$ , we can write:

$$\|UU^T\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\left((UU^T)_{ij}\right)^2 = \sum_{p=1}^{k}\sum_{p'=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{n} U_{ip}U_{jp}U_{ip'}U_{jp'}$$

The expression $U_{ip}U_{jp}U_{ip'}U_{jp'}$ is zero if either $i$ or $j$ does not belong to cluster $p$ or $p'$, *i.e.*, one of the terms becomes zero, hence we further have:

$$\|UU^T\|_F^2 = \sum_{p=1}^{k}\sum_{p'=1}^{k}\sum_{i,j\in p\cap p'} U_{ip}U_{jp}U_{ip'}U_{jp'} = \sum_{p=1}^{k}\sum_{p'=1}^{k}\left(\sum_{i\in p\cap p'} U_{ip}U_{ip'}\right)^2$$

For clarity, we use $p$ to denote both the index of the cluster and the cluster itself, *i.e.*, we simply write $p \cap p'$ instead of $U_{.p} \cap U_{.p'}$. Now, we can rewrite the above formula with our notations as:

$$\|UU^T\|_F^2 = \sum_{u\in U}\sum_{u'\in U}\left(\sum_{i\in u\cap u'} u_{\leftarrow i} \times u'_{\leftarrow i}\right)^2 = \sum_{u\in U}\sum_{u'\in U} o_{uu'}^2 \overset{*}{=} O_{UU}$$

whereas $\overset{*}{=}$ is true when $\varphi(x) = x^2$. Similarly, we can show that:

$$
\begin{aligned}
\|UU^T - VV^T\|_F^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n} \left((UU^T)_{ij} - (VV^T)_{ij}\right)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \left[\left((UU^T)_{ij}\right)^2 + \left((VV^T)_{ij}\right)^2 - 2\left((UU^T)_{ij}(VV^T)_{ij}\right)\right] \\
&\overset{*}{=} \mathcal{O}_{UU} + \mathcal{O}_{VV} - 2\sum_{i=1}^{n}\sum_{j=1}^{n}\left((\sum_{p=1}^{k} U_{ip}U_{jp})(\sum_{q=1}^{r} V_{iq}V_{jq})\right) \\
\text{where} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\left((\sum_{p=1}^{k} U_{ip}U_{jp})(\sum_{q=1}^{r} V_{iq}V_{jq})\right) &= \sum_{p=1}^{k}\sum_{q=1}^{r}\sum_{i,j\in p\cap q} U_{ip}U_{jp}V_{iq}V_{jq} \\
&= \sum_{p=1}^{k}\sum_{q=1}^{r}\left(\sum_{i\in p\cap q} U_{ip}V_{iq}\right)^2 = \sum_{p=1}^{k}\sum_{q=1}^{r} o_{pq}^2 \overset{*}{=} \mathcal{O}_{UV}
\end{aligned}
$$

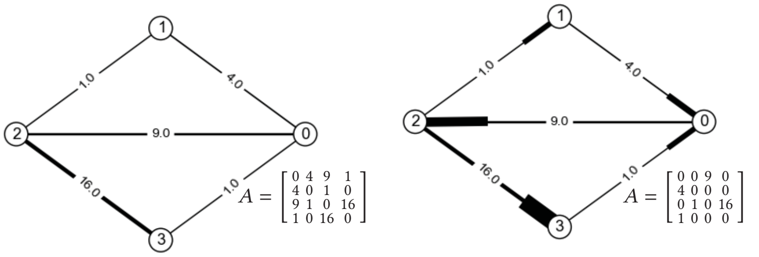
On the other hand we have

$$
\begin{aligned}
&|UU^T||VV^T| = \left(\sum_{i=1}^{n}\sum_{j=1}^{n}(UU^T)_{ij}\right)\left(\sum_{i=1}^{n}\sum_{j=1}^{n}(VV^T)_{ij}\right) \\
&= \left(\sum_{p=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{n} U_{ip}U_{jp}\right)\left(\sum_{q=1}^{r}\sum_{i=1}^{n}\sum_{j=1}^{n} V_{iq}V_{jq}\right) = \left(\sum_{p=1}^{k}\sum_{i,j\in p} U_{ip}U_{jp}\right)\left(\sum_{q=1}^{r}\sum_{i,j\in q} V_{iq}V_{jq}\right) \\
&= \sum_{p=1}^{k}\left(\sum_{i\in p} U_{ip}\right)^2 \sum_{q=1}^{r}\left(\sum_{i\in q} V_{iq}\right)^2 = \sum_{p=1}^{k}\left(o_p\right)^2 \sum_{q=1}^{r}\left(o_q\right)^2 = \sum_{p=1}^{k}\sum_{q=1}^{r}\left(o_p o_q\right)^2 = n^2\mathcal{E}_{UV}
\end{aligned}
$$

Therefore we can re-formulate Equation 3.16 as:

$$
1 - \frac{\|UU^T - VV^T\|_F^2}{\|UU^T\|_F^2 + \|VV^T\|_F^2 - \frac{2}{n^2}|UU^T||VV^T|} \overset{*}{=} 1 - \frac{\mathcal{O}_{uu} + \mathcal{O}_{vv} - 2\mathcal{O}_{uv}}{\mathcal{O}_{uu} + \mathcal{O}_{vv} - 2\mathcal{E}_{UV}}
$$

□



$$A = \begin{bmatrix} 0 & 4 & 9 & 1 \\ 4 & 0 & 1 & 0 \\ 9 & 1 & 0 & 16 \\ 1 & 0 & 16 & 0 \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 0 & 9 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 16 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$N = \begin{bmatrix} 2 & 3 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 4 \end{bmatrix}, NN^T = \begin{bmatrix} 14 & 4 & 9 & 1 \\ 4 & 5 & 1 & 0 \\ 9 & 1 & 26 & 16 \\ 1 & 0 & 16 & 17 \end{bmatrix}, M = \begin{bmatrix} 3 & -2 & 0 & 0 & -1 \\ 0 & 2 & -1 & 0 & 0 \\ -3 & 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & -4 & 1 \end{bmatrix}, Mabs(M^T) = \begin{bmatrix} 4 & -4 & 9 & -1 \\ 4 & 3 & -1 & 0 \\ -9 & 1 & 8 & 16 \\ 1 & 0 & -16 & 15 \end{bmatrix}$$

***Figure B.1:*** *Example for incident matrix of a weighted and or directed graph. For undirected weighted graph we use unsigned incident matrix N with square roots of weights in the actual graph. For a directed graph, and in case the clustering differentiates between the two directions, we can use the oriented incidence matrix M, where $A - A^T + D_o - D_i = M \times abs(M^T)$.*

## B.2 Incidence Matrix of Graph

Here, we elaborate on the unsigned incidence matrix used to represent the structure of a network. This structure is usually represented with an adjacency matrix, $A_{n\times n}$, where $a_{ij}$ represents the association between node $i$ and $j$, *e.g.*, the existence of an edge. An alternative representation is the incidence matrix $N_{n\times m}$, where each column corresponds to an edge and marks the nodes that it connects. This representation is very similar to the clustering representation used in the Chapter 3. Therefore we use the incidence matrix to compare a clustering with the structure of the graph. In more detail, we define incidence matrix of a given Graph $G(V = [v_1, v_2, \ldots, v_n], E = [e_1, e_2, \ldots, e_m])$, as a $n \times m$ matrix $N$, where $N_{ik} = 1$ if node $v_i$ is incident with edge $e_k = (v_i, v_j)$, and zero otherwise. The incidence matrix $N$ and the adjacency matrix $A$ are related by:

$$D + A = NN^T \tag{B.1}$$

Where $D$ is the diagonal matrix of the node degrees. This definition extends for a weighted network, $G(V, E, W = [w_1, w_2, \ldots w_m])$, by defining $N_{ik} = \sqrt{w_k}$; for which Equation B.1 still holds, *i.e.*,

$$D + A = NN^T, \quad D_{ii} = \sum_j A_{ij}, \quad A_{ij} = w_k, \forall\, e_k = (v_i, v_j)$$

The extension for directed networks is not trivial and depends on the application. Here we present a discussion on possible extensions. The first choice is simply assuming the undirected version of the graph. For a directed network, we can consider two separate $n \times m$ matrices of in-incidence and out-incidence, $B_i$ and $B_o$, which respectively mark all sink nodes and source nodes in their corresponding column. Then for adjacency, in-degree, and out-degree matrices we respectively have: $A = B_o B_i^T$, $D_i = B_i B_i^T$ and $D_o = B_o B_o^T$. If we consider undirected version of this graph, then the oriented incidence matrix derives as $M = B_o - B_i$, where the unsigned incidence matrix is $N = B_o + B_i$. We further have $D = \frac{1}{2}(MM^T + NN^T) = D_i + D_o$, and :

$$A + A^T + D = NN^T$$

This extension ignores the directions, which is also the case with the current definition of a network clustering. Since the co-membership matrix calculated in the clustering distances is symmetric by definition, hence can not differentiate between pairs of nodes −the directions.

Alternatively, we can modify clustering of a network to mark each member as a source or destination with positive or negative signs, similar to the oriented incident matrix. With this modification the co-membership matrix for a clustering $U$ should be calculated as $U \times abs(U^T)$, where $abs(U)$ denotes element-wise absolute value. Such co-membership could be compared with the oriented incident matrix, to compare the agreement of the directed structure with the directed clustering; see Figure B.1. One may also note that Equation B.1 resembles the definition of the Laplacian matrix $L = D - A$, which is also defined based on an oriented incidence matrix, *i.e.*, $L = MM^T$. The Laplacian matrix is well-established in graph theory. Here we have $N = abs(M)$, which can be extended similar to the generalization proposed for the Laplacian form of directed graphs [11, 27].

# Appendix C

# Appendix of Chapter 4

## C.1 Algorithms

### C.1.1 Generalized 3-Pass Benchmark

Algorithm 4 summarizes the **assignment** of nodes to communities, which has two main phases, *1) Determine the community sizes:* By sampling from the community size distribution. Sample communities until the sum of all community sizes equals the total number of nodes.
*2) Assign nodes to the communities:* First select a random node in the network and add it, if its degree is less than the size of the community. Any vertex that is too big for the current community gets put aside. At the end of the community assignment, any nodes that left over are randomly added to a community that is big enough to contain their degree. If that community is already full, then it ejects a smaller-degree node and adds in the left over node. The ejected node then gets added into the leftover node queue. The process continues until there is no left over nodes.

```
 1: $\{s_1, s_2 \ldots s_m\}$ based on $\theta^C$ and $G$ { determine capacity of communities}
 2: $\{c_1 = \emptyset, c_2 = \emptyset \ldots c_m = \emptyset\}$ { initialize the communities}
 3: $H \leftarrow \emptyset$ { initialize the homeless queue}
 4: for $v \in G$ do { assign nodes to communities}
 5:     randomly choose $c_i$ from $C$ where $s_i > |c_i|$
 6:     if $s_i \geq (1 - \mu) \times G.degree(v)$ then $c_i.add(v)$
 7:     else $H.add(v)$
 8: while $H$ is not empty do { assign homeless nodes }
 9:     randomly choose $v$ from $H$
10:     $i \leftarrow rand(1, m)$ { pick a community at random}
11:     if $s_i \geq (1 - \mu) \times G.degree(v)$ then { if internal degree is smaller than the community size}
12:         $c_i.add(v)$
13:         $H.remove(v)$
14:         if $|c_i| > s_i$ then { kick out a random node}
15:             randomly choose $u$ from $c_i$
16:             $c_i.remove(u)$
17:             $H.add(u)$
```

**Algorithm 4:** Assign($G$, $\theta^C$)

*1) For the CN assignment*, line 5 of the Algorithm 4 is changed so that the community $c_i$ is chosen from $C$, for node $v$, with probabilities proportional to the current neighbours of $v$ in $c_i$; see Algorithm 5 for the details.

*2) For NE assignment*, we change line 6-7 of the Algorithm 4, so that after community $c_i$ is chosen for node $v$, we expand from node $v$ using a BFS(Breadth-first search), to also add its neighbours to $c_i$, until $c_i$ is full or the expansion is ended; see Algorithm 6 for details.

Note that this way of assigning nodes to communities results in far fewer re-wirings because all of the internal edges are already there. Fewer re-wirings yields more realistic network structure.

```
...
$P \leftarrow 1\ \forall c_i \in C$
for $i \in G.neigh(v)$ do
    $p_{c_i} \leftarrow p_{c_i} + 1$
$P \leftarrow P/sum(P)$
randomly choose $c_i$ from $C$ where $s_i > |c_i|$, and probabilities are proportional to $P$
...
```

**Algorithm 5:** CN Assign

```
...
$M \leftarrow [0] \times G.n$
$Q \leftarrow v$
$M_v \leftarrow 1$
while $len(Q) > 0$ and $c_i < s_i$ do
    $v \leftarrow Q.pop$
    $c_i.add(v)$
    if $s_i \geq (1 - \mu) \times G.degree(v)$ then
        $Q.add(v)$
        for $u \in G.neigh(v)$ do
            if $M_u == 0$ then
                $Q.add(u)$
                $M_u \leftarrow 1$
...
```

**Algorithm 6:** NE Assign

Algorithm 7 summarizes **overlay** of communities on the network through rewirings. This procedure is not described in details in the authors' original paper [79], and the available code for it reflects the modified LFR version[78]; hence the procedure presented here, is our best guess of the original method.

*1) Determine rewirings:* Each vertex should have $\mu$% edges leading out of the community (between), and $(1 - \mu)$% edges leading to other nodes in the community (within). So we start by computing how many edges need to rewired for each vertex. For example, Vertex $v$ might need 2 external edges swapped to internal edges; thus has a desired internal change of $\delta^w = +2$, and a desired external change of $\delta^b = -2$.

*2) Rewire edges within communities:* After computing all the desired changes, iterate through all of the vertices. For each vertex, while it requires more internal edges (i.e. internal change $> 0$ ), we loop through all other vertices that require more internal edges. If there is another vertex in the same community and it requires more internal edges, then we pair them up and add an edge between them (i.e. internal change decreases by one for both). Moreover, we remove the excess internal edges, with a similar process; see line 14-21 of procedure 7 for more details.

*3) Rewire edges between communities:* The same process occurs for external change. In this case we look for other vertices that need an external edge and are out not in the current community of the target vertex. We add an external edge between these two vertices.

### C.1.2 LFR Derivation From 3-Pass Framework

Here we describe LFR in terms of this generalization. The network model used in the LFR benchmark is the CF model, described in details in the related work section. More specifically, for the LFR, $\theta^G = \{N, k_{avg}, k_{max}, \gamma\}$, which are respectively: number of nodes in the graph, average G.degree, maximum G.degree and exponent of power law G.degree distribution. These parameters are basically used to determine the G.degree sequence of $G$; from which the graph is then synthesized using the CF model.

On the other hand, the parameters for communities are $\theta^C = \{\mu, \beta, c_{min}, c_{max}\}$, which are respectively: mixing parameter, exponent of power law distribution for community sizes, minimum size and maximum size for communities. The latter three are used to determine the capacity of communities; whereas the mixing parameter $\mu$ controls the difficulty of problem, which is used in the rewiring phase.

Key issues of LFR benchmarks are discussed in the chapter. From a technical point of view, the implementation of these benchmarks is heuristic and complex, which rules out modifications as well as analytical analysis. When in fact this sampling process of structured networks, does not need to be complicated. An example of unnecessary effort in the original LFR implementation is many rewirings in order to strictly stick to the exact degree distribution of the nodes, when this degree distribution is itself generated/sampled randomly in the first place. In our implementation of the LFR, we relaxed these restrictions to reach a simpler approach, which is as effective as the original one if not better, since it reduces the amount of rewirings.

### C.1.3 FARZ Model

The procedures for FARZ generator is described in the Chapter 4. The assign algorithm is left out due to its triviality which is described here.

```
$\mu \leftarrow \theta^C$
for $v \in G$ do { determine rewirings per node}
    $\delta^b[v] \leftarrow \lfloor \mu \times deg(v) - deg^b(v, c) \rfloor$
    $\delta^w[v] \leftarrow -\delta^b[v]$ { desired within changes}
for $c \in C$ do { rewire edges within communities}
    $I \leftarrow \{v \mid v \in c \wedge \delta^w[v] > 0\}$ { add internal edges}
    while $|I| \geq 2$ do
        randomly choose $v \neq u$ from $I$
        add $edge(u, v)$ to $G$
        $\delta^w[v] \leftarrow \delta^w[v] - 1$
        $\delta^w[u] \leftarrow \delta^w[u] - 1$
        if $\delta^w[v] = 0$ then $I.remove(v)$
        if $\delta^w[u] = 0$ then $I.remove(u)$
    for $\{v \mid v \in c \wedge \delta^w[v] < 0\}$ do { remove excess edges}
        $I \leftarrow \{u \mid edge(u, v) \in G \wedge u \in c \wedge \delta^w[u] < 0\}$
        while $|I| \geq 1 \wedge \delta^w[v] < 0$ do
            randomly choose $u$ from $I$
            remove $edge(u, v)$ from $G$
            $\delta^w[v] \leftarrow \delta^w[v] + 1$
            $\delta^w[u] \leftarrow \delta^w[u] + 1$
            $I.remove(u)$
for $c \in C$ do { rewire edges between communities }
    $I \leftarrow \{v \mid v \in c \wedge \delta^b[v] > 0\}$ { add between edges}
    $O \leftarrow \{v \mid v \notin c \wedge \delta^b[v] > 0\}$
    while $|I| \geq 1 \wedge |O| \geq 1$ do
        randomly choose $v$ from $I$
        randomly choose $u$ from $O$
        add $edge(u, v)$ to $G$
        $\delta^b[v] \leftarrow \delta^b[v] - 1$
        $\delta^b[u] \leftarrow \delta^b[u] - 1$
        if $\delta^b[v] = 0$ then $I.remove(v)$
        if $\delta^b[u] = 0$ then $O.remove(u)$
    for $\{v \mid v \in c \wedge \delta^b[v] < 0\}$ do { remove excess edges}
        $O \leftarrow \{u \mid edge(u, v) \in G \wedge u \notin c \wedge \delta^b[u] < 0\}$
        while $|O| \geq 1 \wedge \delta^b[v] < 0$ do
            randomly choose $u$ from $O$
            remove $edge(u, v)$ from $G$
            $\delta^b[v] \leftarrow \delta^b[v] + 1$
            $\delta^b[u] \leftarrow \delta^b[u] + 1$
            $O.remove(u)$
```

**Algorithm 7:** Overlay($G$, $C$, $\theta^C$)

```
for [1 . . . r] do
    $cid \leftarrow select(\{1 \dots k\}, p_u = \frac{|u|+\phi}{\sum_{v}(|v|+\phi)})$
    if $i \notin C[cid]$ then $C[cid] \leftarrow i$
```

**Algorithm 8:** FARZ Assign (i, C)

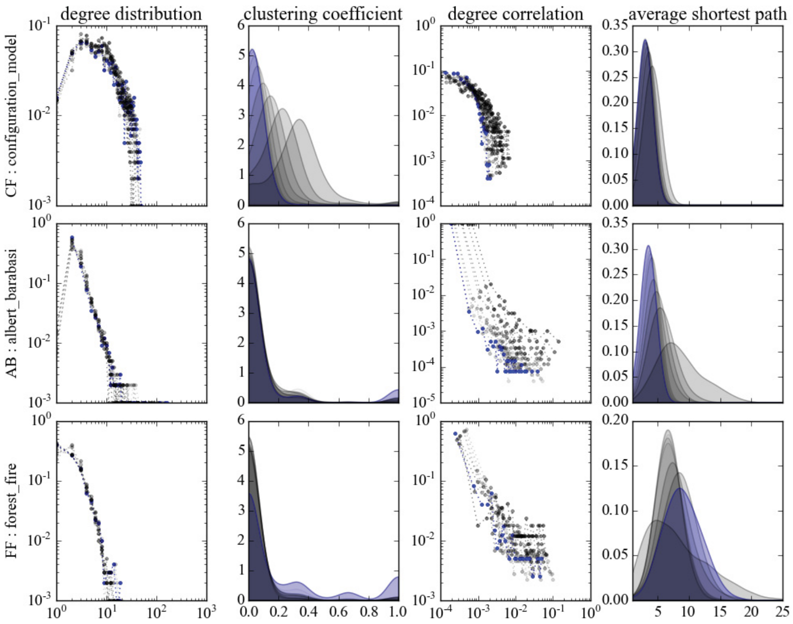
## C.2 Extended Results

Here we report extended results for the experiments reported in the Chapter 4, i.e. distribution plots for all the properties of networks generated by the generalized 3-pass benchmark, and rankings of algorithms in a wider set of settings for the FARZ.

### C.2.1 Generalized 3-Pass Benchmark

Complete plots for the properties of the synthesized networks from the different variations derived from our generalized 3-pass model, described in the Chapter 4. The probability density estimation for parameters, as they change by $\mu$. In each plot, rows are different models: CF(top), BA(middle), FF(last). Initial network is marked by blue. This corresponds to Figure 4.3 in the Chapter 4.

***Figure C.1:*** *Original LFR Assignment*



### C.2.2 FARZ Extended Results

***Figure C.2:*** *CN Assignment*

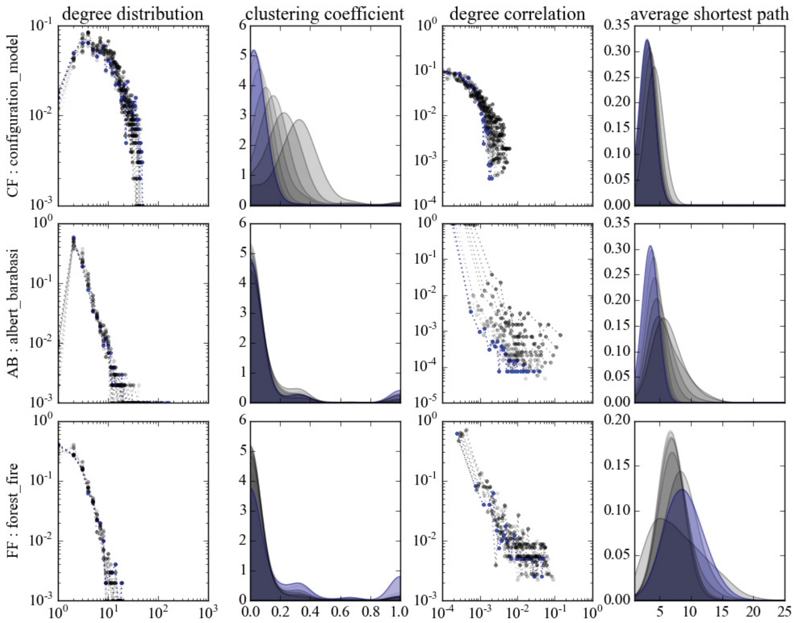


***Figure C.3:*** *NE Assignment*

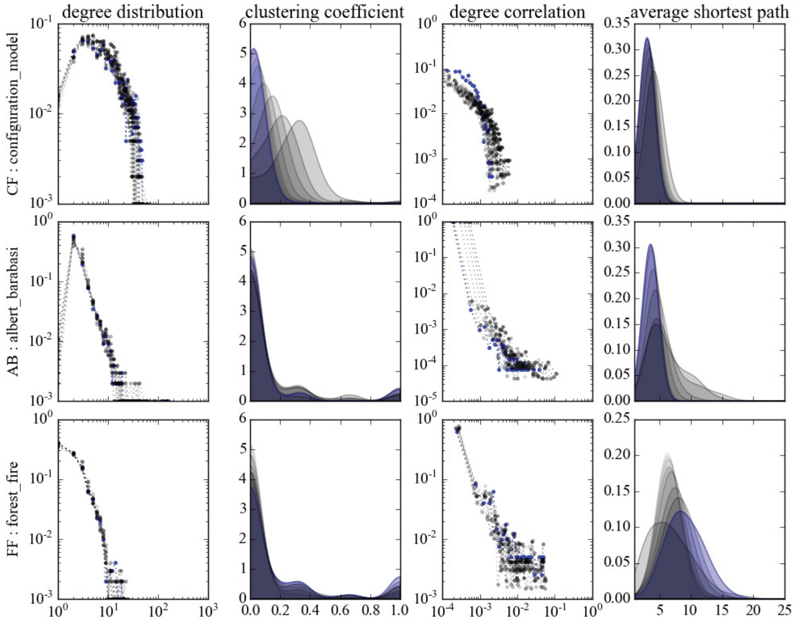
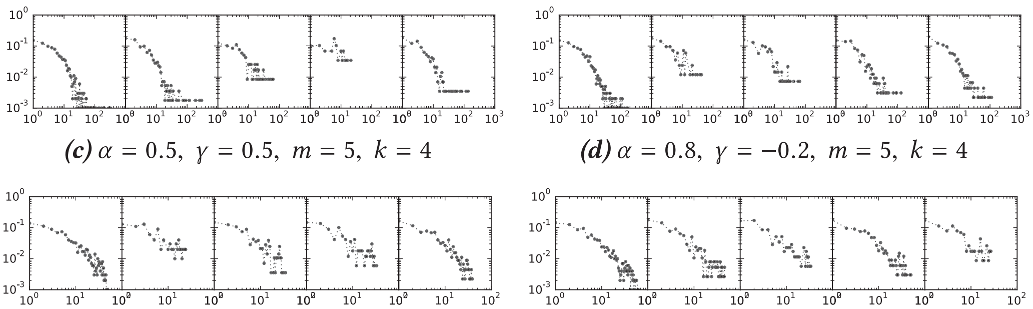


***Figure C.4:** Degree distributions per community for synthetic networks generated by FARZ for 4 different settings of Figure 4.6. The first plot reports the degree distribution for the overall network, and the subsequent subplots show the degree distribution per community.*


*(a)* $\alpha = 0.2,\ \gamma = -0.8,\ m = 5,\ k = 4$

*(b)* $\alpha = 0.5,\ \gamma = -0.5,\ m = 5,\ k = 4$

*(c)* $\alpha = 0.5,\ \gamma = 0.5,\ m = 5,\ k = 4$

*(d)* $\alpha = 0.8,\ \gamma = -0.2,\ m = 5,\ k = 4$

***Figure C.5:** Degree distributions per community for the synthetic network generated by LFR of Figure 4.6.*

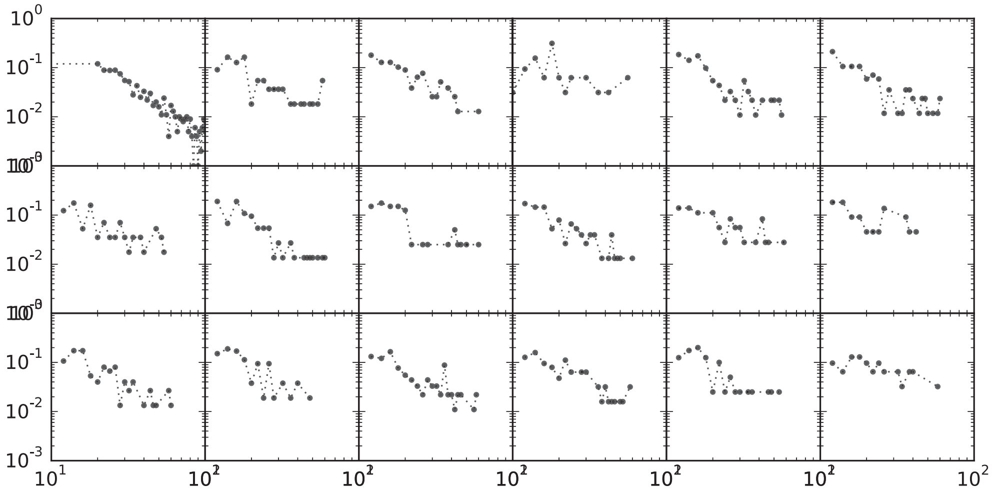
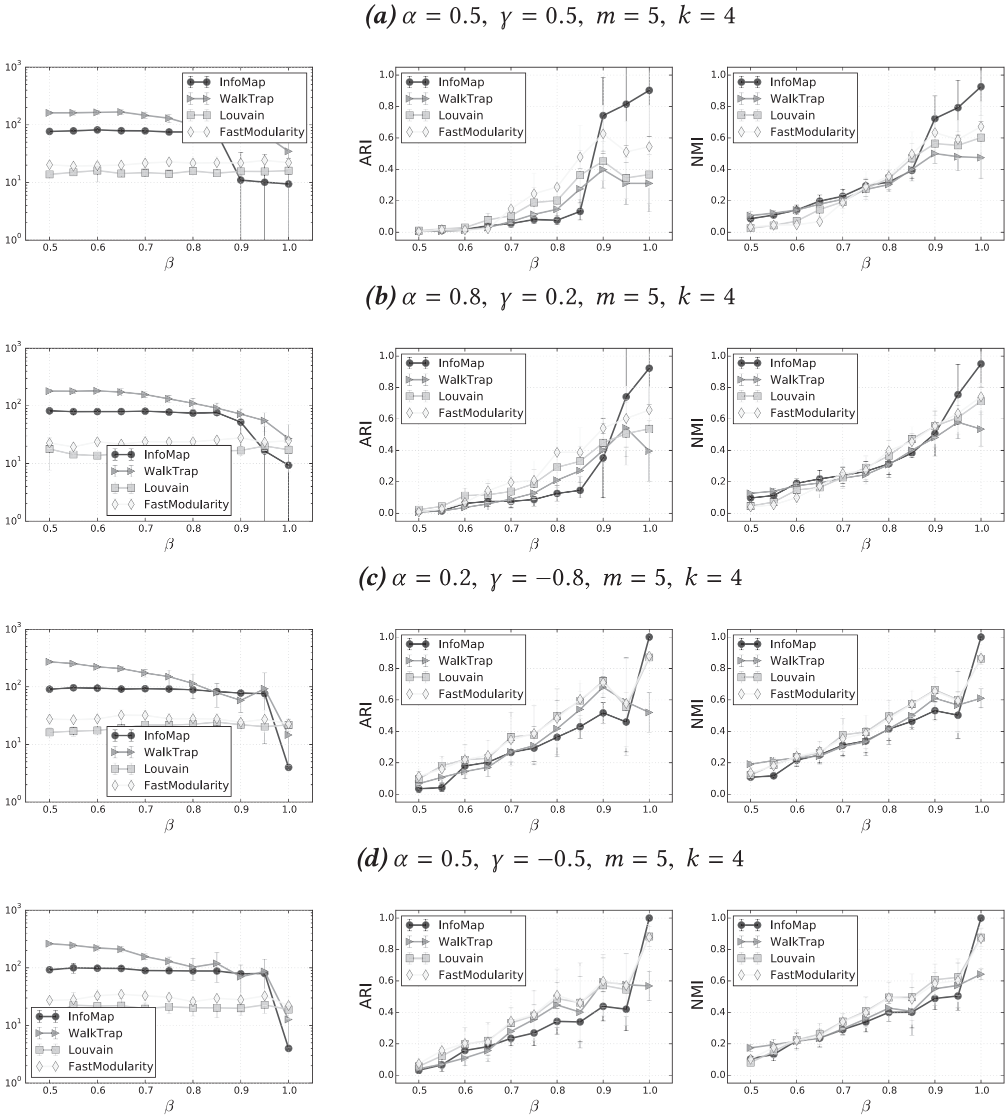


***Figure C.6:*** *Comparing performance of community mining algorithms on benchmarks with* ***positive and negative degree correlation****, all the four settings. Also reporting the number of clusters found by each method.*



***(a)*** $\alpha = 0.5,\ \gamma = 0.5,\ m = 5,\ k = 4$

***(b)*** $\alpha = 0.8,\ \gamma = 0.2,\ m = 5,\ k = 4$

***(c)*** $\alpha = 0.2,\ \gamma = -0.8,\ m = 5,\ k = 4$

***(d)*** $\alpha = 0.5,\ \gamma = -0.5,\ m = 5,\ k = 4$

***Figure C.7:*** *Same algorithms compared on LFR, setting is the 1000B used in [76], i.e. -N 1000 -k 20 -maxk 50 -t1 2 -t2 1 -minc 20 -maxc 100.*

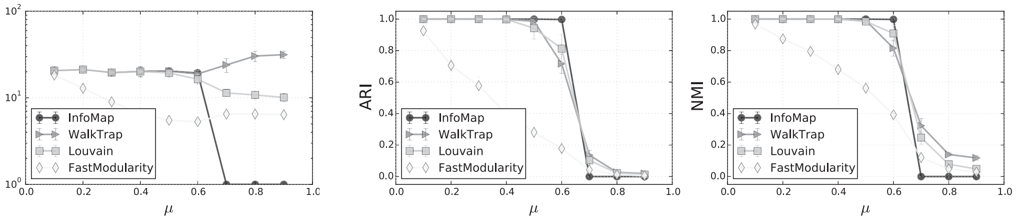
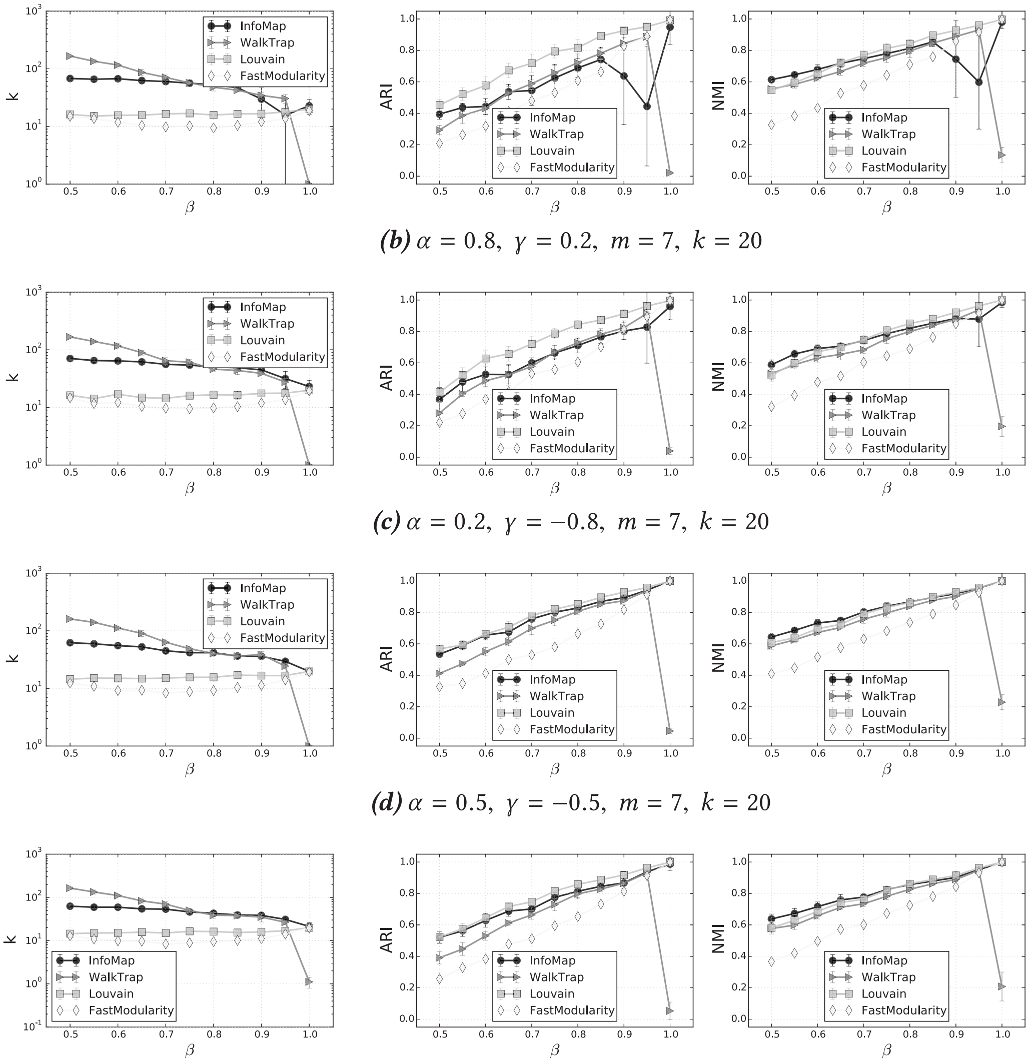


***Figure C.8:** Comparing performance of community mining algorithms similar to Figure C.6 but on* ***denser benchmarks (**$m = 6$**) with more communities (**$k = 20$**)**. *In this setting Louvain clearly performs the best in particular in networks with positive degree correlation. The drop in $\beta = 1$ is due to the communities not linked together which makes the network disconnected and causes problem for the WalkTrap algorithm. The other random walk based method, InfoMap, also seems to have difficulty when communities are well separated, i.e. when $\beta \in [.85, .95]$ and $\gamma > 0$.*

***(a)*** $\alpha = 0.5,\ \gamma = 0.5,\ m = 7,\ k = 20$

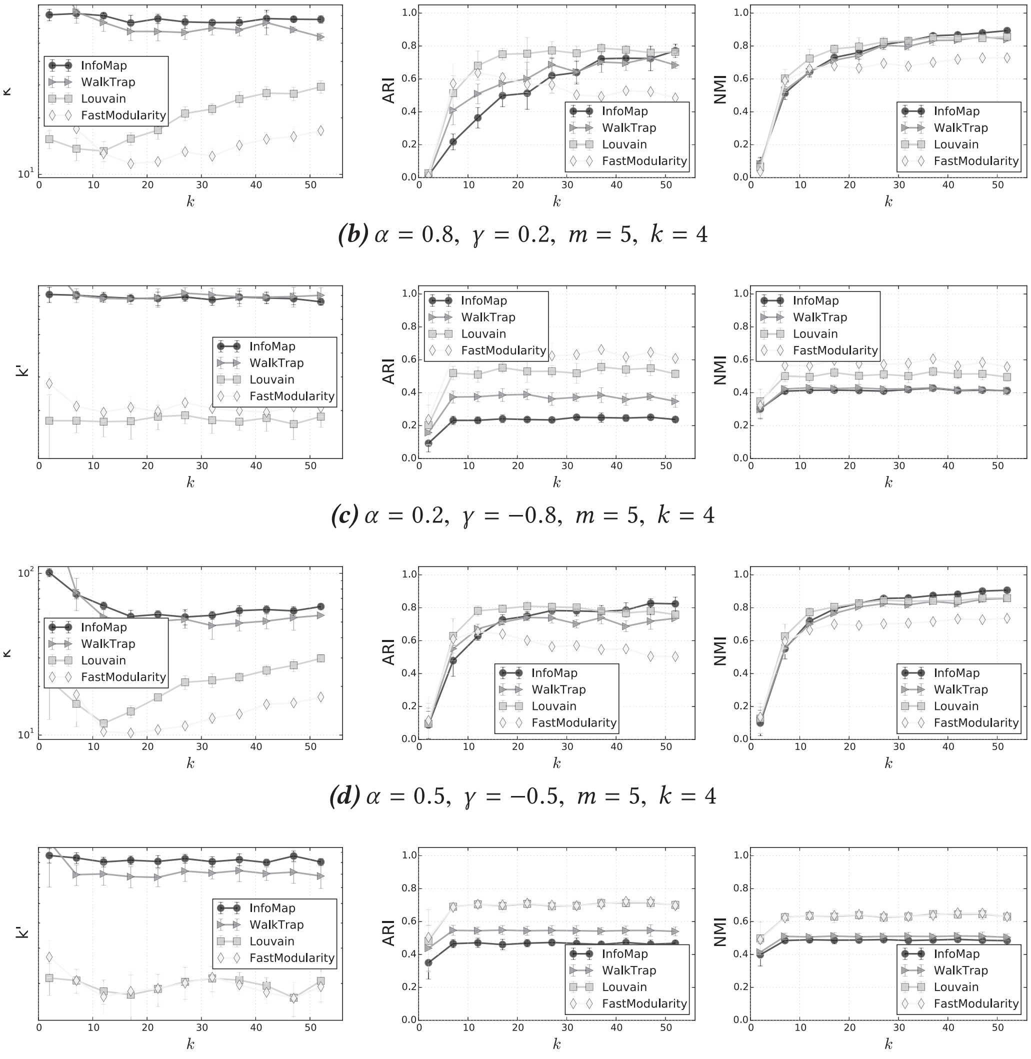


***(b)*** $\alpha = 0.8,\ \gamma = 0.2,\ m = 7,\ k = 20$

***(c)*** $\alpha = 0.2,\ \gamma = -0.8,\ m = 7,\ k = 20$

***(d)*** $\alpha = 0.5,\ \gamma = -0.5,\ m = 7,\ k = 20$

***Figure C.9:*** *Performance of community mining algorithms on benchmarks with different* ***number of built-in communities****, for all the four settings. Also reporting the number of clusters found by each method.*



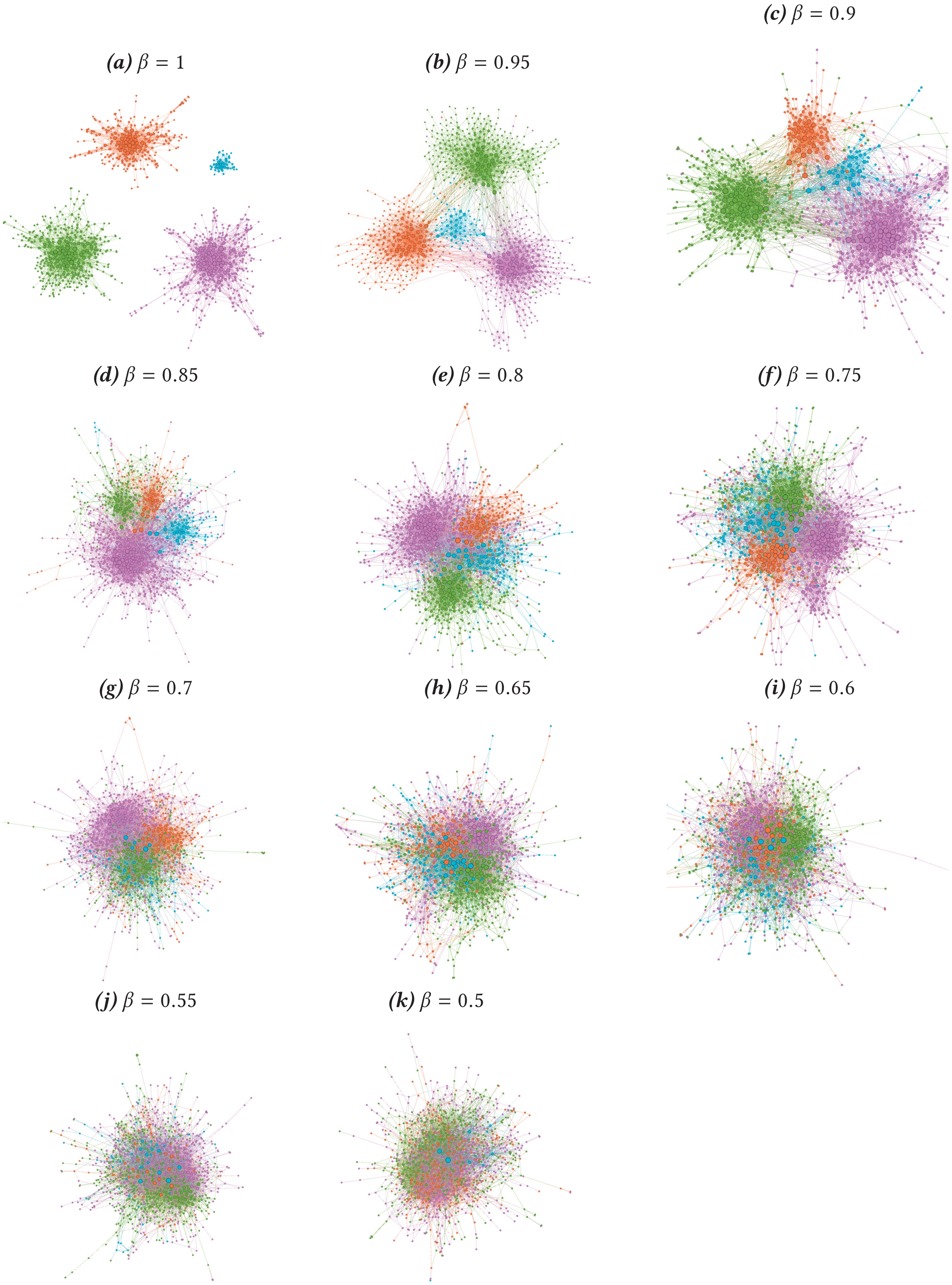
***(a)*** $\alpha = 0.5, \ \gamma = 0.5, \ m = 5, \ k = 4$

***(b)*** $\alpha = 0.8, \ \gamma = 0.2, \ m = 5, \ k = 4$

***(c)*** $\alpha = 0.2, \ \gamma = -0.8, \ m = 5, \ k = 4$

***(d)*** $\alpha = 0.5, \ \gamma = -0.5, \ m = 5, \ k = 4$

***Figure C.10:*** *Example of actual graphs generated by FARZ, used in the previous plots,* $\alpha = 0.5,\ \gamma = 0.5,\ m = 5,\ k = 4$. *Plots visualized with Gephi toolbox using ForceAtlas2 layout, where node sizes corresponds to the degree of the nodes, and the colours of nodes to their assigned communities.*



***(a)*** $\beta = 1$

***(b)*** $\beta = 0.95$

***(c)*** $\beta = 0.9$

***(d)*** $\beta = 0.85$

***(e)*** $\beta = 0.8$

***(f)*** $\beta = 0.75$

***(g)*** $\beta = 0.7$

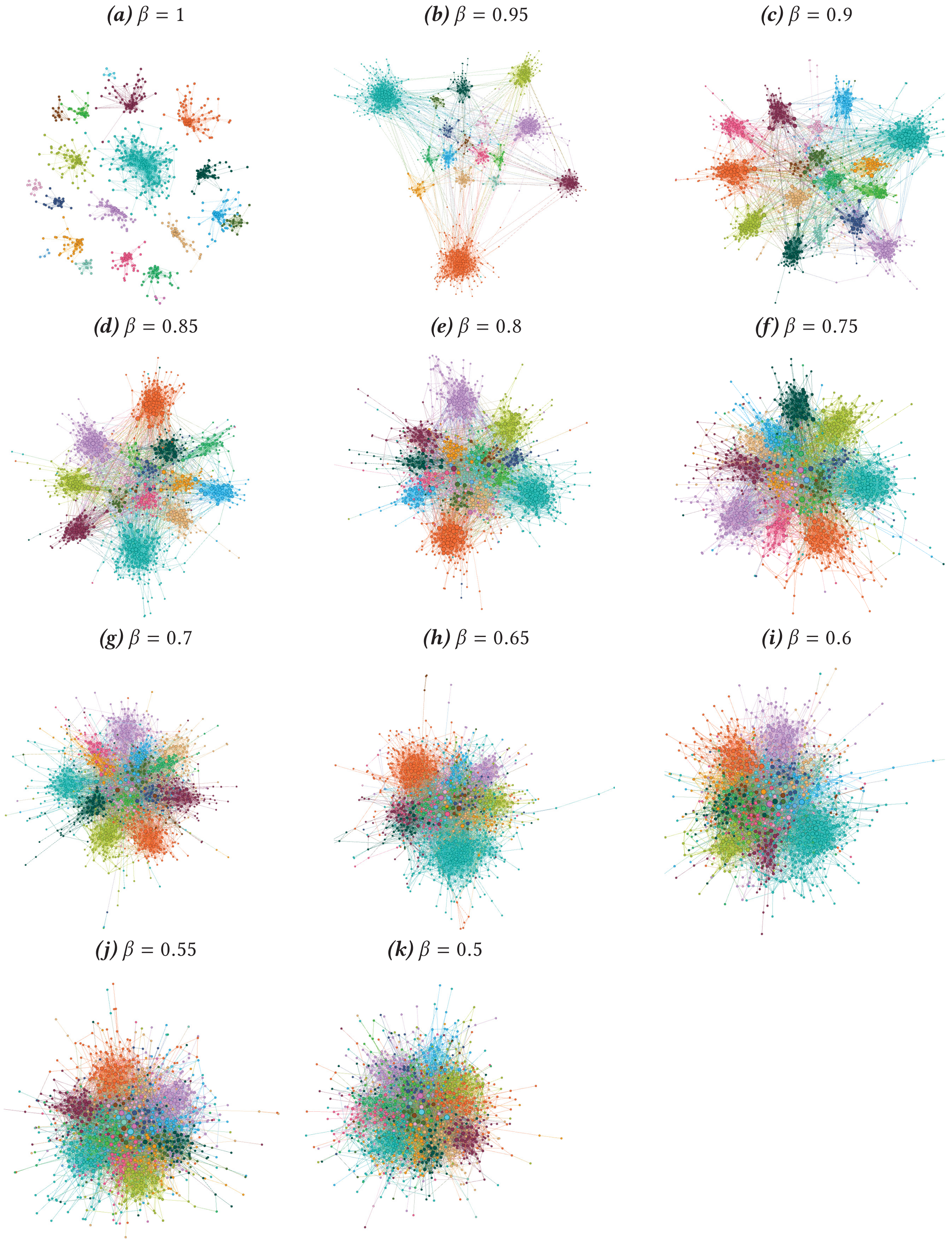
***(h)*** $\beta = 0.65$

***(i)*** $\beta = 0.6$

***(j)*** $\beta = 0.55$

***(k)*** $\beta = 0.5$

***Figure C.11:*** *Example of actual graphs generated by FARZ, used in the previous plots,* $\alpha = 0.5,\ \gamma = 0.5,\ m = 7,\ k = 20.$



***(a)*** $\beta = 1$

***(b)*** $\beta = 0.95$

***(c)*** $\beta = 0.9$

***(d)*** $\beta = 0.85$

***(e)*** $\beta = 0.8$

***(f)*** $\beta = 0.75$

***(g)*** $\beta = 0.7$

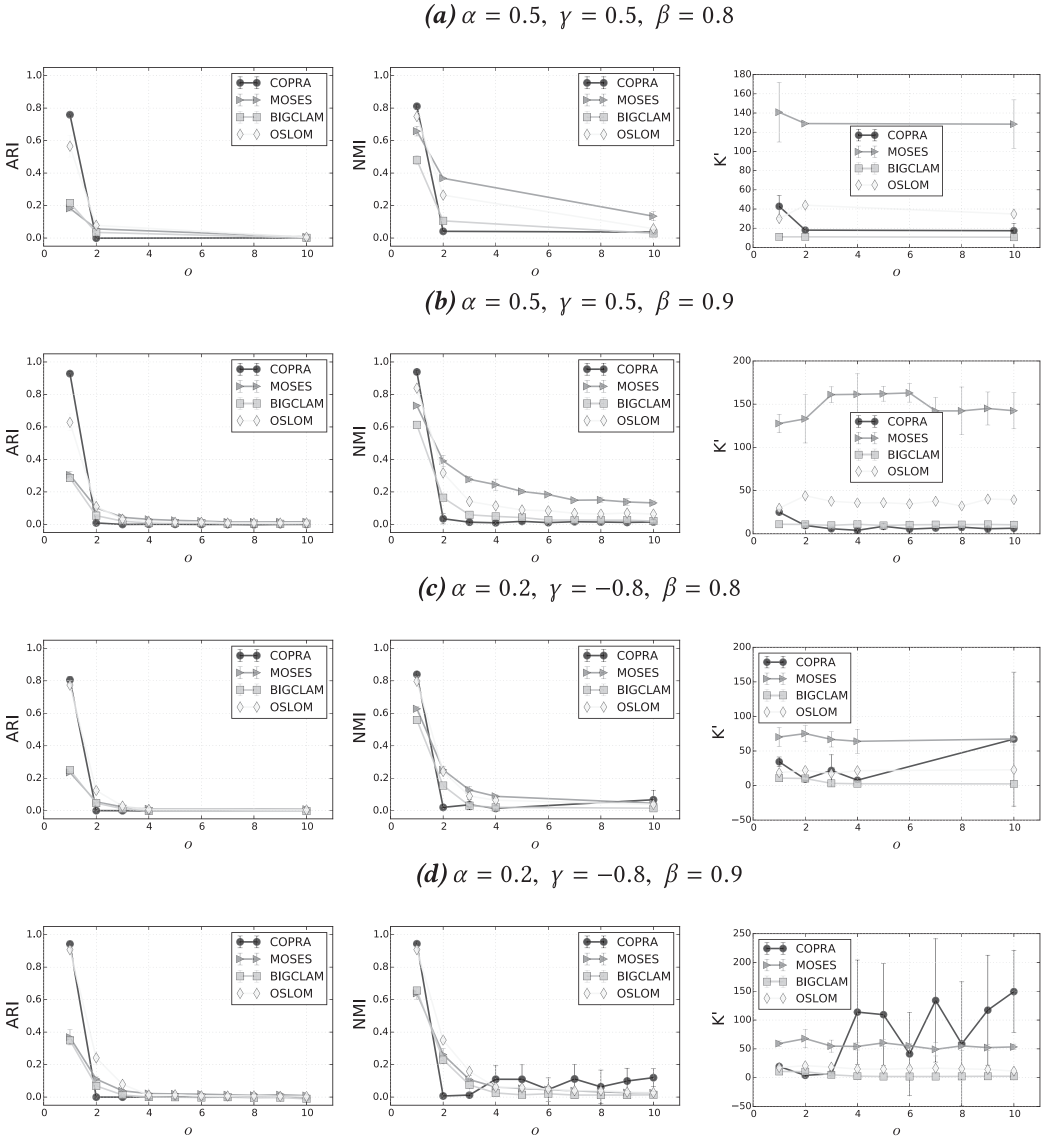
***(h)*** $\beta = 0.65$

***(i)*** $\beta = 0.6$

***(j)*** $\beta = 0.55$

***(k)*** $\beta = 0.5$

***Figure C.12:*** *Comparing performance of community mining algorithms on benchmarks with* ***overlapping communities****, plotted as a function of the number of communities each node can belong to. All methods perform poorly, for when nodes are all overlapping.*



***(a)*** $\alpha = 0.5,\ \gamma = 0.5,\ \beta = 0.8$

***(b)*** $\alpha = 0.5,\ \gamma = 0.5,\ \beta = 0.9$

***(c)*** $\alpha = 0.2,\ \gamma = -0.8,\ \beta = 0.8$

***(d)*** $\alpha = 0.2,\ \gamma = -0.8,\ \beta = 0.9$