Community Structure
in Temporal Multilayer Networks,
and its Application to Financial Correlation Networks

Marya Bazzi
Somerville College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2015

Acknowledgements

I would like to thank the EPSRC (CASE studentship BK/10/41) and HSBC bank for funding this work. I am very grateful to Stacy Williams, Mark McDonald, and Dan Fenn for their financial insight and many very useful discussions. I would particularly like to thank Mason Porter and Sam Howison for their continuous support and guidance, without which this work would not have been possible.

Abstract

Many real-world applications in the social, biological, and physical sciences involve large systems of entities that interact together in some way. The number of components in these systems can be extremely large, so some simplification is typically needed for tractable analysis. A common representation of interacting entities is a network. In its simplest form, a network consists of a set of nodes that represent entities and a set of edges between pairs of nodes that represent interactions between those entities. In this thesis, we investigate clustering techniques for time-dependent networks.

An important mesoscale feature in networks is communities. Most community detection methods are designed for time-independent networks. A recent framework for representing temporal networks is multilayer networks. In this thesis, we focus primarily on community-detection in temporal networks represented as multilayer networks. We investigate three main topics: a community detection method known as multilayer modularity maximization, the development of a benchmark for community detection in temporal networks, and the application of multilayer modularity maximization to temporal financial asset-correlation networks. We first investigate theoretical and computational issues in multilayer modularity maximization. We introduce a diagnostic to measure *persistence* of community structure in a multilayer network partition and we show how communities one obtains with multilayer modularity maximization reflect a trade-off between time-independent community structure within layers and temporal persistence between layers. We discuss computational issues that can arise when solving this method in practice and we suggest ways to mitigate them. We then propose a benchmark for community detection in temporal networks and carry out various numerical experiments to compare the performance of different methods and computational heuristics on our benchmark. We end with an application of multilayer modularity maximization to temporal financial correlation networks.

Papers

Most of the work in this thesis is either in a manuscript that has been accepted for publication or is part of a manuscript that is in preparation. Details are given below.

[1] M. Bazzi, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison, *Community detection in temporal multilayer networks, with an application to correlation networks.* To appear in Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal, arXiv1501.00040, 2015.

[2] M. Bazzi, L. G. S. Jeub, M. A. Porter, A. Arenas, and S. D. Howison, *Community-detection benchmark for multilayer networks*, in preparation, 2015.

[3] M. Bazzi, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison, *Community structure in temporal financial asset-correlation networks*, in preparation, 2015.

Statement of Originality

The research in this thesis is a result of collaboration between myself and my coauthors on the listed papers. My collaborators have helped develop the ideas described in this thesis, but I have performed all of the analysis leading to the results that I present.

Contents

List of Figures

List of Tables

N	number of nodes
$\vert\mathcal{T}\vert$	number of layers
i_s	node i in layer s
$\boldsymbol{A}_s$	adjacency matrix of observed network in layer s (p. 10)
$\boldsymbol{P}_s$	adjacency matrix of null network in layer s (p. 12)
k_{i_s}	strength of node i in layer s (p. 12)
$\mathcal{N}_s := \{1_s, \dots, N_s\}$	set of nodes in layer s (p. 30)
$\mathcal{C}$	set of all $N\vert\mathcal{T}\vert$-node hard partitions (p. 29)
$C \in \mathcal{C}$	arbitrary partition C in the set of partitions $\mathcal{C}$
$C_k \in C$	arbitrary set C_k of nodes in a partition C
c_{i_s}	set assignment of node i_s in C (p. 30)
$C_k\vert_s := C_k \cap \mathcal{N}_s$	set of nodes in C_k that are in layer s (p. 30)
$C\vert_s := \{C_k\vert_s, C_k \in C\}$	N-node partition induced on layer s (p. 30)
$\mathrm{Pers}(C) := \sum_{s=1}^{\vert\mathcal{T}\vert-1} \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_{s+1}})$	persistence (p. 63)
$\mathrm{Pers}(C) / \left[N(\vert\mathcal{T}\vert - 1)\right]$	normalized persistence
$\mathrm{Pers}(C)\vert_{s,r} := \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_r})$	pairwise persistence(p. 63)
$\left(\mathrm{Pers}(C)\vert_{s,r}\right) / N$	normalized pairwise persistence
$\gamma \geq 0$	resolution parameter (p. 15)
$\omega \geq 0$	inter-layer coupling parameter (p. 28)

Table 1: Notation

$\boldsymbol{B}_s$	single-layer modularity matrix (p. 14)
$\boldsymbol{\mathcal{B}}$	multilayer modularity matrix (p. 29)
$Q(C\mid\boldsymbol{B}) := \sum_{i,j=1}^{N} B_{ij}\delta(c_i, c_j)$	single-layer modularity (p. 13)
$\mathcal{C}_{\text{local}}(\boldsymbol{\gamma})$	set of local optima for different γ values (p. 36)
$\hat{\boldsymbol{A}} := \big(\sum_{C\in\mathcal{C}_{\text{local}}} \delta(c_i, c_j)\big)/\lvert\mathcal{C}_{\text{local}}(\boldsymbol{\gamma})\rvert$	multiscale association matrix (p. 36)
$Q(C\mid\boldsymbol{B_1}, \ldots, \boldsymbol{B}_{\lvert\boldsymbol{\mathcal{T}}\rvert}; \omega) := \sum_{s=1}^{\lvert\mathcal{T}\rvert} \sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s}, c_{j_s}) + 2\omega\text{Pers}(C)$	multilayer modularity (p. 29)
$Q(C\mid\boldsymbol{B_1}, \ldots, \boldsymbol{B}_{\lvert\boldsymbol{\mathcal{T}}\rvert}; 0) = \sum_{s=1}^{\lvert\mathcal{T}\rvert} \sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s}, c_{j_s})$	intra-layer modularity (p. 63)
$\mathcal{C}_{\max}(\omega)$	set of optimal multilayer partitions for (2.24) (p. 68)
$C^{\omega}_{\max}$	arbitrary optimal partition in $\mathcal{C}_{\max}(\omega)$ (p. 68)
$\mathbb{P}^s_0$	null distribution of layer s (p. 94)
$\mathcal{C}_{\text{bench}}(\mathbb{P}^s_0, p)$	set of benchmark multilayer partitions (p. 94)
$nVI(C, C') := \frac{1}{\lvert\mathcal{T}\rvert} \sum_{s=1}^{\lvert\mathcal{T}\rvert} SnVI(C\rvert_s, C'\rvert_s)$	multilayer normalized variation of information (p. 40)
T	uniformly discretized time window (p. 47)
$\lvert T\rvert$	time window length (p. 47)
δT	time window step size (p. 47)
$z_i = \{z_i(t), t \in T\}$	logarithmic returns of asset i (p. 43)
$\hat{z}_i(t) := (z_i(t) - \langle z_i\rangle)/\sigma(z_i)$	standardized logarithmic return (p. 46)
$\text{corr}(z_i, z_j)$	Pearson correlation between asset i and asset j (p. 44)

Table 1 (Cont.): Notation

Chapter 1

Introduction

1.1 Networks and financial systems

Many real-world applications involve large systems of entities that interact together in some way. Many of these systems are known as *complex systems*. Although there is no precise definition of a complex system, these are often described as systems made up of many parts, and in which the behaviours of individual parts are highly variable and strongly dependent on the behaviours of other parts [11,180]. The idea that "the whole is more than the sum of its parts" is a key feature of complex systems: the subparts can behave differently in isolation than they do when part of a larger system, and it is thus necessary to study them together to gain insight into the behaviour of the system as a whole [183]. An intuitive example is a social system, where even if we knew everything possible about the behaviour of each individual in a group, we would not be able to infer the behaviour of the individuals as a group (e.g., development and evolution of societal rules). Other examples of complex systems include the spread of disease or information, the human brain, bird flocks, intracellular systems, and traders in a stock market [11]. This list is far from exhaustive and the diversity of examples suggests that the notion of a complex system has more to do with how something is taking place rather than what is taking place [153].

The number of components in a complex system can be extremely large, so some simplification is typically needed for tractable analysis. For example, one can coarse-grain the system by choosing a subset of components and interactions that capture the phenomenon of interest. Given a question, one tries to keep the components and interactions that are most relevant to it. As with any simplification process, reducing the number of interactions in a complex system may throw away information, but may make more evident patterns of interaction and useful information.

A common way of representing a simplified description of a complex system of interacting entities is a *network*. Networks provide useful representations of complex systems across many disciplines [147]. In its simplest form, a network is simply a graph: it consists of a set of *nodes* that represent entities and a set of *edges* between pairs of nodes that represent interactions between those entities. One can consider *weighted graphs* (in which each edge has an associated value that quantifies the strength of the interaction of interest) or *unweighted graphs* (weighted graphs with binary edge weights). Common types of networks include social networks (which arise via offline and/or online interactions), information networks (e.g., hyperlinks between webpages in the World Wide Web), infrastructure networks (e.g., transportation routes between cities), and biological networks (e.g., metabolic interactions between cells or proteins, food webs, etc.).

In this thesis, we focus on networks generated from correlations between financial asset price time series as real-world examples [23, 38]. In this approach, nodes in the network are financial products, and edge weights are statistical estimates of correlations between two time series during a time interval [23, 63, 65, 134, 154]. We use correlations as a simple quantification of dependency structures within markets. Such structures are important because financial systems are composed of various interdependent components — such as banks, consumers, firms, and investors — that react to a market by adjusting their buying and selling decisions, prices, and forecasts, which in turn causes the market to change. Time plays an important role in the feedback loop between financial components and the aggregate they form, the market. The price of an asset is one manifestation of these interactions and the fluctuations of prices in time should to some extent reflect the interactions, feedback, and adaptation of market participants. The study of correlations between price time series is therefore important to market practitioners. It can help to better understand systemic risk and the ways in which different markets interact [138]. It is also important for credit derivatives [201], where modeling correlation structure plays a crucial role, although this is not a focus of the thesis. The practical motivation for this study was provided by HSBC bank. Members from the HSBC bank FX Quantitative Strategy Group and Asset Allocation Strategy Group have investigated the issue of identifying higher-order structure in correlation matrices that is not captured by standard stock market classification (e.g., using principal component analysis [65, 101, 200]). We propose to approach this problem using techniques from network science, and to investigate whether such techniques can provide additional insight.

1.2 Community structure

1.2.1 Time-independent networks

Given a network representation of a system, it can be useful to apply a coarse-graining technique in order to investigate features that lie between features at the "microscale" (e.g., nodes and pairwise interactions) and the "macroscale" (e.g., total edge weight and degree distribution) [148, 163]. One thereby studies "mesoscale" features such as core–periphery structure and community structure. Loosely speaking, a *community* (or *cluster*) in a network is a set of nodes that are "more densely" connected to each other than they are to nodes in the rest of the network [68, 163]. Giving a precise definition of "densely connected" is, of course, necessary to have a method for community detection. It is important to recognize at the outset that although the intuition is clear in a general sense, it seems impossible to provide a single definition of community. The definition is subjective and may depend, in particular, on the application in question [68, 163]. What one calls a "community" ultimately depends on the method and algorithm that one uses to detect them. We elaborate on this point further in Chapter 2. We restrict ourselves to *hard partitions*, in which each node is assigned to exactly one community, and we use the term "partition" to mean "hard partition". It is also important, but beyond the scope of this thesis, to consider "soft partitions", in which communities can overlap [68, 100, 156, 163].

Analysis of community structure has been useful in a wide range of applications; many of which are described in [68, 73, 148, 163]. In social networks, communities can reveal groups of people with common interests, places of residence, or other similarities [150, 191]. In biological systems, communities can reveal functional groups that are responsible for synthesizing or regulating an important chemical product [81, 127]. In financial systems, despite the diversity of markets, financial products, and geographical locations, financial assets can exhibit strong time-dependent correlations, both within and between asset classes. It is a key concern for market practitioners (e.g., for portfolio diversification) to estimate the strengths of these correlations and to identify sets of assets that are highly correlated [134, 200].

1.2.2 Temporal networks

Most methods for detecting communities are designed for time-independent (or *static*) networks. However, in many applications, entities and/or interactions between entities evolve in time. In such applications, one can use the formalism of *temporal* *networks*, where nodes and/or their edge weights vary in time [89,90]. This is important for numerous applications — including person-to-person communication [202], one-to-many information dissemination (e.g., Twitter networks [76] and Facebook networks [210]), cell biology [90], neuroscience [22], ecology [90], finance [63–65,154], and more.

Two main approaches have been adopted to detect communities in time-dependent networks. The first entails constructing a static network by aggregating snapshots of the evolving network at different points in time into a single network (e.g., by taking the mean or total edge weight for each edge across all time points, which can be problematic if the set of nodes varies in time). One can then use standard network techniques. The second approach entails using static community-detection techniques on each element of a time-ordered sequence of networks at different times or on each element of a time-ordered sequence of network aggregations (computed as above) over different time intervals (which can be either overlapping or nonoverlapping) and then tracking the communities across the sequence [15,63,64,91,132,156].

A third approach consists of embedding a time-ordered sequence of networks in a larger network [140,161] (related ideas are also available in other contexts [56,186]). Each element of the sequence is a network *layer*, and nodes at different time points are joined by *inter-layer* edges. This approach was introduced in [140] and the resulting network is a type of *multilayer network* [31,109]. The main difference between this approach and the previous approach is that the presence of nonzero inter-layer edges introduces a dependence between communities identified in one layer and connectivity patterns in other layers. Thus far, most computations that have used a multilayer representation of temporal networks have assumed that inter-layer connections are "diagonal" (i.e., they exist only between copies of the same node) and "ordinal" (i.e., they exist only between consecutive layers) [109]. Diagonal coupling is a natural model of the persistence of node identity in time, while ordinal coupling preserves the temporal ordering.

The authors of [140] proposed a generalization of *modularity maximization*, a popular clustering method for time-independent networks, to multilayer networks. Modularity is a function that measures the "quality" of a network partition into disjoint sets of nodes by computing the difference between the total edge weight in sets in the observed network and the total expected edge weight in the same sets in a "null network" generated from some "null model" [68,163]. Modularity maximization consists of maximizing the modularity quality function over the space of network partitions. (In practice, given the combinatorial complexity of this maximization problem, one uses some computational heuristic and finds a local maximum [77].) Intuitively, the null model controls for connectivity patterns that one anticipates finding in a network, and one uses modularity maximization to identify connectivity patterns in an observed network that are stronger than what is anticipated. We give a precise definition of the modularity function for single-layer networks in Section 2.1, where we distinguish between a "null network" and a "null model" in modularity maximization. In Section 4.1, we discuss the role of null networks in modularity maximization and use correlation networks as illustrative examples.

To date, almost no theory has explained how a multilayer partition obtained with zero inter-layer coupling (which reduces to single-layer modularity maximization on each layer independently) differs from a multilayer partition obtained with nonzero inter-layer coupling. In Section 4.2.2, we prove several properties of an optimal solution for the multilayer maximization problem to better understand how such partitions differ and how one can try to exploit this difference in practice. We stress that our results only depend on the form of the maximization problem and are not restricted to the modularity quality function. (We give examples of other measures one could use in Chapter 2.)

For many community-detection methods, one cannot detect optimal communities in polynomial time [37, 68] and one therefore often needs to resort to computational heuristics [29,40,118]. We describe computational issues that can arise when using the popular (locally greedy) Louvain heuristic [29] to solve the multilayer maximization problem in Section 4.2.3, and we suggest ways to try and mitigate them. Furthermore, many scalable heuristics currently have few or no theoretical guarantees on how closely an identified partition resembles an optimal partition [37,68,77]. Although it is clear that efforts towards establishing theoretical grounding for community-detection heuristics need further development [72,80,95,135,167], it can be useful to benchmark the output of a heuristic against something "known" [45, 72, 120]. As community detection in multilayer networks is a recent area of research [31, 109], there is currently no "standard" (synthetic or real) multilayer benchmark that one can use to compare multilayer community-detection methods or heuristics. We propose a simple community-detection benchmark for temporal networks in Chapter 5, and we perform several numerical experiments to compare the behaviour of the methods and computational heuristics that we discuss throughout the thesis. Such tests can help to assess different methods in practice, in order to begin to understand when some methods work relatively well and when they work less well.

1.3 Organization of the thesis

This thesis is organized into six additional chapters. A continuous thread runs through it as we move from an overview of community-detection methods for time-independent and temporal networks to a discussion of a particular community-detection method for temporal networks and then to developing benchmarks for community-detection in temporal networks. In all chapters, we use toy examples and numerical experiments with financial data to illustrate our points. We end with an application of multilayer modularity maximization to temporal correlation networks.

In Chapter 2, we give an overview of community-detection methods for time-independent and temporal networks, and we introduce most of the notation, assumptions, and conventions that we adopt throughout the thesis. In Chapter 3, we describe the multilayer network representation that we use in our experiments with financial data. In Chapter 4, we investigate multilayer modularity maximization, a community-detection methods that can be applied to temporal networks. In particular, we address two main issues: (1) the choice of null network and (2) the role of inter-layer edges in multilayer modularity maximization. In Section 4.1, we investigate the effect that a null network can have on communities identified in correlation networks when the distribution of edge weights in the null network is sample-dependent. In Section 4.2, we prove several properties to better understand how a partition obtained with zero inter-layer coupling can differ from a partition obtained with nonzero inter-layer coupling. We also investigate computational issues that can arise when one uses the Louvain heuristic [29] or a spectral bipartitioning heuristic [146] to solve the multilayer maximization problem. We suggest ways to try and mitigate some of these issues. In Chapter 5, we propose a family of community-detection benchmarks for temporal networks. We perform various numerical experiments to compare the performance of several methods and computational heuristics that we discuss in Chapters 2 and 4. In Chapter 6, we perform further numerical experiments on financial data based on the results and observations of Chapters 4 and 5. Finally, in Chapter 7, we offer some conclusions and suggest some possible directions for future work.

Chapter 2

Preliminaries and Literature Review

We begin with some ideas about "clusters" (or "communities") before starting our description of the methods that we discuss in this thesis. Broadly speaking, *cluster analysis* consists of grouping entities (e.g., a set of d-dimensional data points [133] or nodes in a network [68]) into clusters. Usually, entities within the same cluster should "resemble" one another and/or entities in different clusters should "differ" from one another [84]. Cluster analysis is an old problem [12, 60] and has been addressed in many contexts (e.g., natural sciences, psychology, medicine, engineering, economics, marketing) and by researchers in many disciplines (e.g., statistics, mathematics, computer science) [84, 177, 204]. As a result, the cluster analysis literature is vast and heterogeneous [9, 68, 84, 129]. Furthermore, cluster analysis is inherently subjective. As pointed out by Backer and Jain in 1981 [16], "in cluster analysis, a group of objects is split up into a number of more or less homogeneous subgroups on the basis of an often subjectively chosen measure of similarity (i.e., chosen subjectively based on its ability to create "interesting" clusters), such that the similarity between objects within a subgroup is larger than the similarity between objects belonging to different subgroups". In particular, the definition of cluster may be strongly influenced by the application that one studies or even by the algorithm that one uses to detect them. In the present thesis, we restrict our focus to the clustering problem in networks with time-varying edge weights [68, 90].

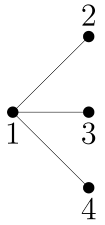
While a network in its simplest form is a graph, some methods in community detection were inherited from *graph partitioning* [9, 177] and many new methods were developed specifically for community detection [68, 148, 163]. Possible reasons for this are that networks can have properties that are not typically studied in the graph-theoretic literature for which one may wish to account for (e.g., signed edge weights [75, 189], temporality [90, 140]), one often has no *a priori* information on the number or sizes of clusters in the final partition (and several graph partitioning methods require such information) [9, 146], and one's interpretation of what constitutes a "good" partition in a network (e.g., objective functions such as "modularity" [73] and "stability" [54, 116]) can differ from graph-theoretic measures of a "good" partition (e.g., "normalized cut" [182] and "conductance" [34]). In the remainder of this section, we give a brief overview of community-detection methods for time-independent and temporal networks, and we give a more detailed description of the methods on which we focus in this thesis.

2.1 Community detection in time-independent networks

2.1.1 Overview

There are myriad community-detection methods, and the overview that we will give is by no means comprehensive [50, 68, 148, 163]. We briefly describe two common approaches to community detection and subsequently describe in detail the approach that we focus on in the present thesis. In this chapter, we also introduce most of the notation that we use in the thesis. We give a summary of the notation in Section 2.5 and include a notation table in Table 1.

Consider an N-node network $\mathcal{G}$ and let the edge weights between pairs of nodes be $\{A_{ij} | i, j \in \{1, \ldots, N\}\}$, so that $\boldsymbol{A} = (A_{ij}) \in \mathbb{R}^{N \times N}$ is the *adjacency matrix* of $\mathcal{G}$. In this thesis, we only consider symmetric adjacency matrices (and hence undirected networks), so $A_{ij} = A_{ji}$ for all i and j. We call $\mathcal{G}$ an *observed network* and give an example of an unweighted and undirected observed network with its corresponding adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{N \times N}$ in Fig. 2.1. An old and still widely used technique for detecting communities in certain types of networks is *hierarchical clustering* (e.g., social networks [68, 163], financial Pearson correlation networks [23, 134, 154]) [68, 148]. Hierarchical methods can be *divisive* (one starts with a single cluster and then iteratively splits clusters until one obtains N singleton clusters or satisfies some other stopping criterion) or *agglomerative* (one starts with N singleton clusters and then iteratively combines clusters until one obtains a single cluster). Hierarchical methods produce exactly nested partitions which one can represent as a *dendrogram* [68,134,154]. Many hierarchical methods require an adjacency matrix with pairwise edge weights given



$$\longleftrightarrow \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2.1: Example of an unweighted and undirected network (left) and its corresponding adjacency matrix (right).

by some measure of similarity (e.g., "cosine similarity" [17, 148] or a distance function derived from Pearson correlation [134,154]). Given a choice of adjacency matrix, hierarchical methods differ by the definition of cluster similarity used to iteratively combine or split clusters (e.g., "single, complete, or average linkage" for agglomerative hierarchical clustering [68, 148] and "edge betweeness" for divisive hierarchical clustering [68,73]). One reason hierarchical methods are appealing is that they have been thoroughly studied [84, 98, 177]. Limitations of hierarchical methods include their sensitivity to the cluster-similarity measure adopted, the mis-classification of nodes these methods often produce, and the fact that hierarchical structure is a restrictive and potentially unrealistic assumption in many applications [62,68,73,131,148,150].

A second approach to community detection is block-modelling, a form of '*statistical inference*' for networks [88, 97, 148, 152, 185]. A *stochastic block model* is a probabilistic model that assigns a probability $p_{ij} := \mathbb{P}[A_{ij} = 1]$ to each edge in the network such that the probability of an edge between two nodes depends only on the "blocks" to which these nodes belong [198, 205]. In the context of community detection, "blocks" are communities and we call a (hard) division of nodes into blocks a *planted partition*. A block model can be used to generate a network with a planted partition (e.g., [74,88,197]) or to infer a partition from an observed network by fitting it to a block model (e.g., via a "maximum likelihood" formulation of the community-detection problem [7, 105, 185]). The simplest block model for generating an N-node network with a planted partition is to generate edges between nodes in the same community with probability p_{in} and edges between nodes in different communities with probability p_{out}, where $p_{\text{in}} > p_{\text{out}}$. We use this stochastic block model in the numerical experiments of Chapter 5. Traditionally, block models were developed for *simple* (i.e., no multi-edges and no self-edges) unweighted networks with a hard planted partition [13, 67, 88]. There exist generalizations that account for directed edges [197], "heterogeneous degree distributions" within blocks (e.g., "degree-corrected" block model [105, 151, 205]), weighted edges (e.g., weights drawn from an exponential family distribution [6]), overlapping community structure [7,124,168], etc. A block model approach to community detection has several attractive features, such as its roots in statistical inference (which can provide some theoretical guarantees on the optimal partition [28,39,87]) and its adaptability to various types of community structure (e.g., hierarchical community structure [43], overlapping community structure [7,124,168]). Some practical limitations include the need for a model of the observed network (weighted networks are often thresholded because of this [6]), the trade-off between the number of model parameters and analytic tractability [148], and the need for an *a priori* estimate of the number of communities in the final partition [87, 185, 205].

A third approach to community detection is to define an objective "function" (or "quality function") that maps every partition to a real number. One then identifies communities by optimizing this function over the set of all possible network partitions [68, 73, 148, 163]. Some of these methods mainly exploit links between community structure and network topology [73, 150] and others mainly exploit links between community structure and dynamical processes taking place on the network [54, 116, 171]. We describe three commonly used single-layer optimization methods and their multilayer counterpart in the next few sections. We try to point out connections between the methods. We stress that many more community detection methods exist [68]. (For example, one can use *local* methods, in contrast to the *global* methods described in this section [100, 157].)

2.1.2 Single-layer modularity maximization

2.1.2.1 The modularity function

Consider an undirected N-node network $\mathcal{G}$ with edge weights $\{A_{ij}|i,j \in \{1,\dots,N\}\}$ between pairs of nodes, so that $\boldsymbol{A} = (A_{ij}) \in \mathbb{R}^{N\times N}$ is the adjacency matrix of $\mathcal{G}$. The *strength* of a node i is

$$k_i = \sum_{j=1}^{N} A_{ij} = \sum_{j=1}^{N} A_{ji}\,, \tag{2.1}$$

and it is given by the i^{th} row (or column) sum of $\boldsymbol{A}$.

When studying the structure of a network, it is useful to compare what is observed with what is anticipated. We define a *null model* to be a probability distribution on the set of adjacency matrices and the adjacency matrix of a *null network* to be the expected adjacency matrix under a specified null model. In a loose sense, null models play the role of *prior models*, as they control for features that one anticipates to find in the system under investigation. One can thereby take into account known (or suspected) connectivity patterns that might obscure unknown connectivity patterns that one hopes to discover via processes like community detection. For example, in social networks, one often takes the strength of a node in a null network to be its observed strength k_i [146, 148, 163]. We discuss the use of this null network for financial-asset correlation networks in Section 4.1. In spatial networks that represent the spread of a disease or information between different locations, some authors have used null networks in which edge weights between two locations scale inversely with the distance between them [61, 175].

As we discussed in Chapter 1, one can use modularity maximization to partition a network into sets of nodes called "communities" that have a larger total internal edge weight than the expected total internal edge weight in the same sets in a null network, generated from some null model [68, 148, 150, 163]. *Modularity maximization* consists of finding a partition that maximizes this difference [68, 163]. As we mentioned in Chapter 1, in practice, one uses some computational heuristic and finds a local maximum [77]. In the present thesis, we use the term *modularity* for an arbitrary choice of null network (i.e., we do not restrict ourselves to the standard "Newman–Girvan" choice, which we define in Section 2.1.2.3) and we ignore any normalization constant that depends on the choice of null network but does not affect the solution of the modularity-maximization problem for a given null network. Modularity thus acts as a "*quality function*" $Q : \mathcal{C} \to \mathbb{R}$, where the set $\mathcal{C}$ is the set of all possible N-node network partitions.

Suppose that we have a partition C of a network into K disjoint sets of nodes $\{C_1, \ldots, C_K\}$, where $K \leq N$. We can then define a map $c(\cdot)$ from the set of nodes $\{1, \ldots, N\}$ to the set of integers $\{1, \ldots, K\}$ such that $c(i) = c(j) = k$ if and only if nodes i and j lie in C_k. We use the term *global maximum* to refer to a solution of the modularity maximization problem and the term *local maximum* to refer to a solution that one obtains with a computational heuristic and from which one cannot escape. That is, a local maximum is a partition $C \in \mathcal{C}$ such that $\mathbb{P}[S_{t+1} = C | S_t = C] = 1$, where S_t is a modularity-increasing sampling process on the space of partitions. We call $c(i)$ the *set assignment* (or *community assignment* when C is a global or local maximum) of node i in partition C. The value of modularity for a given partition C is then

$$Q(C|\boldsymbol{A};\boldsymbol{P}) := \sum_{i,j=1}^{N} (A_{ij} - P_{ij})\delta(c_i, c_j)\,, \tag{2.2}$$

where $\boldsymbol{P} = (P_{ij}) \in \mathbb{R}^{N \times N}$ is the adjacency matrix of the null network, c_i is shorthand notation for $c(i)$, and $\delta(c_i, c_j)$ is the Kronecker delta function. We state the modularity-maximization problem as follows:

$$\max_{C \in \mathcal{C}} \sum_{i,j=1}^{N} (A_{ij} - P_{ij}) \delta(c_i, c_j)\,, \tag{2.3}$$

which we can also write as $\max_{C \in \mathcal{C}} Q(C|\boldsymbol{B})$ or $\max_{C \in \mathcal{C}} \sum_{i,j}^{N} B_{ij} \delta(c_i, c_j)$, where $\boldsymbol{B} = \boldsymbol{A} - \boldsymbol{P}$ is the so-called *modularity matrix* [146]. It is clear from (2.3) that pairwise contributions to modularity are only counted when two nodes are assigned to the same set. These contributions are positive (respectively, negative) when the observed edge weight A_{ij} between nodes i and j is larger (respectively, smaller) than the expected edge weight P_{ij} between them. If $A_{ij} < P_{ij}$ for all i and j, then the optimal solution is N singleton communities. Conversely, if $A_{ij} > P_{ij}$ for all i and j, then the optimal solution is a single N-node community. To obtain a partition of a network with a high value of modularity, one hopes to have many edges within sets that satisfy $A_{ij} > P_{ij}$ and few edges within sets that satisfy $A_{ij} < P_{ij}$. As is evident from equation (2.3), what one regards as "densely connected" in this setting depends fundamentally on the choice of null network.

It can be useful to write the modularity-maximization problem using the trace of matrices [146]. As before, we consider a partition C of a network into K sets of nodes $\{C_1, \ldots, C_K\}$. We define the *assignment matrix* $\boldsymbol{S} \in \{0,1\}^{N \times K}$ as

$$S_{ij} = \delta(c_i, j)\,, \tag{2.4}$$

where $j \in \{1, \ldots, K\}$ and $c_i = j$ means that node i lies in C_j [9] . The columns of $\boldsymbol{S}$ are orthogonal and the j^{th} column sum of $\boldsymbol{S}$ gives the number of nodes in C_j. This yields

$$\sum_{i,j=1}^{N} B_{ij} \delta(c_i, c_j) = \sum_{i,j=1}^{N} \sum_{k=1}^{K} S_{ik} B_{ij} S_{jk} = \mathrm{Tr}(\boldsymbol{S}^T \boldsymbol{B} \boldsymbol{S})\,,$$

where $\boldsymbol{S}^T \boldsymbol{B} \boldsymbol{S} \in \mathbb{R}^{K \times K}$, the $(i,i)^{\text{th}}$ term of $\boldsymbol{S}^T \boldsymbol{B} \boldsymbol{S}$ is twice the sum of modularity-matrix entries between pairs of nodes in C_i. (The $(i,j)^{\text{th}}$ off-diagonal term is the sum of modularity-matrix entries between a node in C_i and a node in C_j.) One can then restate the modularity-maximization problem in (2.3) as

$$\max_{S \in \mathcal{S}} \mathrm{Tr}(\boldsymbol{S}^T \boldsymbol{B} \boldsymbol{S})\,, \tag{2.5}$$

where $\mathcal{S}$ is the set of all assignment matrices in $\{0,1\}^{N \times K}$ (where $K \leq N$).

Modularity maximization is one of myriad community-detection methods [68], and it has many limitations (e.g., a resolution limit on the size of communities [69], a huge number of nearly degenerate local maxima [77], and statistical significance issues associated with an optimal partition [107, 121, 135]). Nevertheless, it is a popular method (which has been used successfully in numerous applications [68, 163]), and the ability to specify explicitly what one anticipates is a useful (and under-exploited) feature for users working on different applications [20, 21, 61, 131, 175]. In Section 4.1, we make some observations on one's choice of null network when using the modularity quality function.

2.1.2.2 Multiscale modularity

Many networks include community structure at multiple scales [68, 163], and some systems even have a hierarchical community structure of "parts-within-parts" [183]. In the latter situation, although there are dense interactions within communities of some size (e.g., friendship ties between students in the same school), there are even denser interactions in subsets of nodes that lie inside of these communities (e.g., friendship ties between students in the same school and in the same class year). Some variants of the modularity function have been proposed to detect communities at different scales. A popular choice is to scale the null network using a *resolution parameter* $\gamma \geq 0$ to yield a *multiscale modularity-maximization* problem [167]:

$$\max_{C \in \mathcal{C}} \sum_{i,j=1}^{N} (A_{ij} - \gamma P_{ij}) \delta(c_i, c_j) \,. \tag{2.6}$$

In some sense, the value of the parameter γ determines the importance that one assigns to the null network relative to the observed network. The corresponding modularity matrix and modularity function evaluated at a partition C are $\boldsymbol{B} = \boldsymbol{A} - \gamma \boldsymbol{P}$ and $Q(C|\boldsymbol{A}; \boldsymbol{P}; \gamma) = \sum_{i,j=1}^{N} (A_{ij} - \gamma P_{ij}) \delta(c_i, c_j)$. The special case $\gamma = 1$ yields the modularity matrix and modularity function in the modularity-maximization problem (2.3). Although the multiscale modularity-maximization problem in (2.6) was initially introduced in [167] using an *ad hoc* approach, it has a dynamical interpretation [114, 116] that we will discuss in Section 2.1.3.

In most applications of community detection, the adjacency matrix of the observed and null networks have nonnegative entries. In these cases, the solution to (2.6) when $0 \leq \gamma \leq \gamma^- = \min_{i \neq j, P_{ij} \neq 0} (A_{ij}/P_{ij})$ is a single community regardless of any structure, however clear, in the observed network, because then

$$B_{ij} = A_{ij} - \gamma P_{ij} \geq 0 \quad \text{for all} \quad i, j \in \{1, \dots, N\} \,.$$

(We exclude diagonal terms because a node is always in its own community.) However, the solution to (2.6) when $\gamma > \gamma^+ = \max_{i \neq j, P_{ij} \neq 0} (A_{ij}/P_{ij})$ is N singleton communities because

$$B_{ij} = A_{ij} - \gamma P_{ij} < 0 \quad \text{for all} \quad i, j \in \{1, \ldots, N\} .$$

Partitions at these boundary values of γ correspond to the coarsest and finest possible partitions of a network, and varying the resolution parameter between these bounds makes it possible to examine a network's community structure at intermediate scales.

For an observed and/or null network with signed edge weights, the intuition behind the effect of varying γ in (2.6) on a globally optimal solution is not straightforward. A single community and N singleton communities do not need to be optimal partitions for any value of $\gamma \geq 0$. In particular, B_{ij} has the same sign as A_{ij} for sufficiently small values of γ, and B_{ij} has the opposite sign to P_{ij} for sufficiently large values of γ. We discuss this issue further in Section 4.1, where we explore the effect of varying the resolution parameter on an optimal partition for an observed and null network with signed edge weights.

It is important to differentiate between a "resolution limit" on the smallest community size that is imposed by a community-detection method [69] and inherently multiscale community structure in a network [68, 163, 183]. The former is a limitation of a method and the latter is a feature of the application at hand.[1] While it is desirable that a community-detection method identifies partitions at multiple scales (if present), some community-detection methods possess a lower bound on the smallest detectable community size for a given observed network (i.e., a "resolution limit") [69, 77]. For example, consider a network composed of $\tilde{K}$ cliques and $\tilde{K}$ edges between cliques, such that cliques are arranged into a ring and adjacent cliques are connected by an edge. In such a network, a "natural" division into communities would be one in which each clique constitutes a community. One can show that there is a number $\tilde{K}$ of cliques (that depends on the number of nodes and edges in the network) above which the modularity function with a Newman–Girvan null network (which we describe in Section 2.1.2.3) yields a higher value of modularity to a partition in which pairs of adjacent cliques are merged [69, 77].[2] For a large enough number of cliques, the size of an optimal community is thus larger than that of a clique. One can easily generalize this observation to l adjacent cliques, $2 < l \leq \tilde{K}$,

[1] We note that none of the computational heuristics considered in this thesis impose an a priori constraint on the size or number of communities in the output partition.

[2] The observations in [69, 77] on modularity's resolution limit depend on one's choice of null network. A discussion on the effect of a uniform null network on the resolution limit can be found in [190].

and the manifestation of modularity's resolution limit is thus that intuitively modular structures (e.g., cliques) can be hidden within very large sets in an optimal partition. For the formulation of multiscale modularity in (2.6), the resolution limit described in [69, 77] applies to any fixed value of γ. By varying γ in (2.6), one can identify communities that are smaller than the limit for any particular γ value. In this sense, multiscale formulations of modularity help "mitigate" the resolution limit, though there remain issues [14, 77, 114]. In this thesis, we do not address the issue of how to identify communities at different scales, though we note in passing that the literature includes variants of multiscale modularity. For example, see [14, 79] for an approach where one varies the self-edge (and thus the strength) of each node in the observed network. We make observations on null networks in Section 4.1, and we illustrate how our observations can manifest in practice using the formulation of multiscale modularity in (2.6). Our observations hold independently of the formulation of multiscale modularity that one adopts, but the precise manifestation can be different for different variants of multiscale modularity.

2.1.2.3 Null models and null networks

In this section, we describe three null networks. In the computational experiments of Section 4.1, we will make several observations on the interpretation of communities that we obtain from Pearson correlation matrices using each of these null networks.

Newman-Girvan (NG) null network A popular choice of null network for networks with positive edge weights is the Newman-Girvan (NG) null network, whose adjacency-matrix entries are $P_{ij} = k_i k_j/(2m)$, where k_i are the observed node strengths [144, 150]. This yields the equivalent maximization problems

$$\max_{C \in \mathcal{C}} \sum_{i,j=1}^{N} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \Leftrightarrow \max_{\boldsymbol{S} \in \mathcal{S}} \operatorname{Tr}\left[\boldsymbol{S}^T \left(\boldsymbol{A} - \frac{\boldsymbol{k}\boldsymbol{k}^T}{2m} \right) \boldsymbol{S} \right], \tag{2.7}$$

where $\boldsymbol{k} = \boldsymbol{A}\mathbf{1}$ is the $N \times 1$ vector of node strengths (see (2.1)) and $2m = \mathbf{1}^T \boldsymbol{A}\mathbf{1}$ is the total edge weight of the observed network. This null network can be derived from a variety of null models. One way to generate an unweighted network with expected adjacency matrix $\boldsymbol{k}\boldsymbol{k}^T/(2m)$ is to generate each of its edges and self-edges with probability $k_i k_j/(2m)$ (provided $k_i k_j \leq 2m$ for all i, j). That is, the presence and absence of edges and self-edges is a Bernoulli random variable with probability $k_i k_j/(2m)$ [35, 36]. More generally, any probability distribution on the set of adjacency matrices that satisfies $\mathbb{E}\big[\sum_{j=1}^{N} W_{ij}\big] = k_i$ (i.e., the expected strength equals the observed strength, see for e.g., [41]) and $\mathbb{E}[W_{ij}] = f(k_i)f(k_j)$ for some real-valued function f has an expected adjacency matrix of $\mathbb{E}[\boldsymbol{W}] = \boldsymbol{k}\boldsymbol{k}^T/(2m)$.[3] The adjacency matrix of the NG null network is symmetric and positive semidefinite.

We briefly mention a way of deriving a variant of the NG null network from a model on time-series data (in contrast to a model on a network). The *partial correlation* $\mathrm{corr}(a, b \,|\, c)$ between two sets of observations a and b while controlling for a third set of observations c is the *Pearson correlation* between the residuals that result from the linear regression of a with c and b with c, and it is given by

$$\mathrm{corr}(a, b \,|\, c) = \frac{\mathrm{corr}(a, b) - \mathrm{corr}(a, c)\mathrm{corr}(b, c)}{\sqrt{1 - \mathrm{corr}^2(a, c)}\sqrt{1 - \mathrm{corr}^2(b, c)}}\,, \tag{2.8}$$

where $\mathrm{corr}(a, b)$ is the Pearson correlation between a and b [111]. Suppose that the data used to construct the observed network is a set of time series $\{z_i | i \in \{1, \dots, N\}\}$, where $z_i = \{z_i(t) | t \in T\}$ and T is a discrete set of time points. The authors of [131] pointed out that when $A_{ij} = \mathrm{corr}(z_i, z_j)$, it follows that $k_i = \mathrm{cov}(\hat{z}_i, \hat{z}_{\mathrm{tot}})$ and thus that

$$\frac{k_i k_j}{2m} = \mathrm{corr}(\hat{z}_i, \hat{z}_{\mathrm{tot}})\mathrm{corr}(\hat{z}_j, \hat{z}_{\mathrm{tot}})\,, \tag{2.9}$$

where $\hat{z}_i(t) = (z_i(t) - \langle z_i \rangle)/\sigma(z_i)$ is a standardized time series and $\hat{z}_{\mathrm{tot}}(t) = \sum_{i=1}^{N} \hat{z}_i(t)$ is the sum of the standardized time series.[4] Taking $a = \hat{z}_i$, $b = \hat{z}_j$, and $c = \hat{z}_{\mathrm{tot}}$, equation (2.8) implies that if $\mathrm{corr}(\hat{z}_i, \hat{z}_j \,|\, \hat{z}_{\mathrm{tot}}) = 0$ then $\mathrm{corr}(\hat{z}_i, \hat{z}_j) = k_i k_j/(2m)$. That is, Pearson correlation coefficients between pairs of time series that satisfy $\mathrm{corr}(\hat{z}_i, \hat{z}_j \,|\, \hat{z}_{\mathrm{tot}}) = 0$ are precisely the adjacency-matrix entries of the NG null network. An example set of time series in which pairs of distinct time series satisfy this condition is one in which each standardized time series depends linearly on the mean time series and residuals are mutually uncorrelated (i.e., $\hat{z}_i = \alpha_i \hat{z}_{\mathrm{tot}}/N + \beta_i + \epsilon_i$ for some $\alpha_i, \beta_i \in \mathbb{R}$ and $\mathrm{corr}(\epsilon_i, \epsilon_j) = 0$ for $i \neq j$).

Generalization of Newman-Girvan null network to signed networks (NGS)

In [75], Gómez *et al.* proposed a generalization of the NG null network to signed networks. They separated $\boldsymbol{A}$ into its positive and negative edge weights:

$$\boldsymbol{A} = \boldsymbol{A}^+ - \boldsymbol{A}^-\,,$$

[3] The linearity of the expectation and the assumptions $\mathbb{E}\big[\sum_{j=1}^{N} W_{ij}\big] = k_i$ and $\mathbb{E}[W_{ij}] = f(k_i)f(k_j)$ imply that $f(k_i) = k_i/\sum_{j=1}^{N} f(k_j)$ and $\sum_{j=1}^{N} f(k_j) = \sqrt{2m}$. Combining these equations gives the desired result.

[4] The equality (2.9) holds for Pearson correlation networks, which are signed networks with edge weights in $[-1, 1]$. The strength of a node i is given by the i^{th} (signed) column or row sum of the correlation matrix.

where $\boldsymbol{A}^+$ denotes the positive part of $\boldsymbol{A}$ and $-\boldsymbol{A}^-$ denotes its negative part. That is,

$$A^+_{ij} = \begin{cases} A_{ij} & \text{if } A_{ij} > 0\,, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad A^-_{ij} = \begin{cases} |A_{ij}| & \text{if } A_{ij} < 0\,, \\ 0 & \text{otherwise.} \end{cases}$$

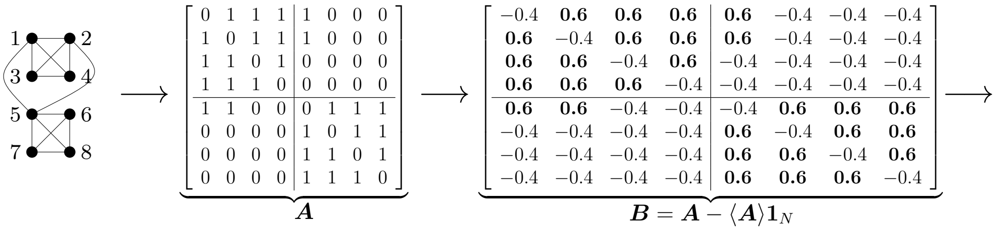
Their generalization of the NG null network to signed networks (NGS) is $P_{ij} = k^+_i k^+_j/(2m^+) - k^-_i k^-_j/(2m^-)$. This yields the maximization problem

$$\max_{C \in \mathcal{C}} \sum_{i,j=1}^{N} \left[\left(A^+_{ij} - \frac{k^+_i k^+_j}{2m^+} \right) - \left(A^-_{ij} - \frac{k^-_i k^-_j}{2m^-} \right) \right] \delta(c_i, c_j)\,, \tag{2.10}$$

where k^+_i and $2m^+$ (respectively, k^-_i and $2m^-$) are the strengths and total edge weight in $\boldsymbol{A}^+$ (respectively, $\boldsymbol{A}^-$). The intuition behind this generalization is to use an NG null network on both unsigned matrices $\boldsymbol{A}^+$ and $\boldsymbol{A}^-$ but to count contributions to modularity from negative edge weights (i.e., the second group of terms in (2.10)) in an opposite way to those from positive edge weights (i.e., the first group of terms in (2.10)). Negative edge weights that exceed their expected edge weight are penalized (i.e., they decrease modularity) and those that do not are rewarded (i.e., they increase modularity). One can generate a network with edge weights 0, 1, or -1 and expected edge weights $k^+_i k^+_j/(2m^+) - k^-_i k^-_j/(2m^-)$ by generating one network with expected edge weights $W^+_{ij} = k^+_i k^+_j/(2m^+)$ and a second network with expected edge weights $W^-_{ij} = k^-_i k^-_j/(2m^-)$ using the procedure described for the NG null network in the previous section. One then defines a network whose edge weights are given by the difference between the edge weights of these two networks. More generally, any probability distribution on the set of signed adjacency matrices $\{\boldsymbol{W} \in \mathbb{R}^{N \times N}\}$ with the same properties as those for the NG null network for $\boldsymbol{W}^+$ and $\boldsymbol{W}^-$ (where $\boldsymbol{W} = \boldsymbol{W}^+ - \boldsymbol{W}^-$ defined as above) will have expected edge weights of $W_{ij} = k^+_i k^+_j/(2m^+) - k^-_i k^-_j/(2m^-)$ for all $i, j \in \{1, \ldots, N\}$ (by linearity of the expectation).

Uniform (U) null network A third null network that we consider is a uniform (U) null network, with adjacency-matrix entries $P_{ij} = \langle k \rangle^2/(2m)$, where $\langle k \rangle := \left(\sum_{i=1}^N k_i\right)/N$ denotes the mean strength in a network. We thereby obtain the equivalent maximization problems

$$\max_{C \in \mathcal{C}} \sum_{i,j=1}^{N} \left(A_{ij} - \frac{\langle k \rangle^2}{2m} \right) \delta(c_i, c_j) \Leftrightarrow \max_{\boldsymbol{S} \in \mathcal{S}} \operatorname{Tr}\left[\boldsymbol{S}^T \left(\boldsymbol{A} - \frac{\langle k \rangle^2}{2m} \mathbf{1}_N \right) \boldsymbol{S} \right], \tag{2.11}$$



$$\underbrace{\left[\begin{array}{cccc|cccc} 0&1&1&1&1&0&0&0\\ 1&0&1&1&1&0&0&0\\ 1&1&0&1&0&0&0&0\\ 1&1&1&0&0&0&0&0\\ \hline 1&1&0&0&0&1&1&1\\ 0&0&0&0&1&0&1&1\\ 0&0&0&0&1&1&0&1\\ 0&0&0&0&1&1&1&0 \end{array}\right]}_{\boldsymbol{A}} \longrightarrow \underbrace{\left[\begin{array}{cccc|cccc} -0.4&\mathbf{0.6}&\mathbf{0.6}&\mathbf{0.6}&\mathbf{0.6}&-0.4&-0.4&-0.4\\ \mathbf{0.6}&-0.4&\mathbf{0.6}&\mathbf{0.6}&\mathbf{0.6}&-0.4&-0.4&-0.4\\ \mathbf{0.6}&\mathbf{0.6}&-0.4&\mathbf{0.6}&-0.4&-0.4&-0.4&-0.4\\ \mathbf{0.6}&\mathbf{0.6}&\mathbf{0.6}&-0.4&-0.4&-0.4&-0.4&-0.4\\ \hline \mathbf{0.6}&\mathbf{0.6}&-0.4&-0.4&-0.4&\mathbf{0.6}&\mathbf{0.6}&\mathbf{0.6}\\ -0.4&-0.4&-0.4&-0.4&\mathbf{0.6}&-0.4&\mathbf{0.6}&\mathbf{0.6}\\ -0.4&-0.4&-0.4&-0.4&\mathbf{0.6}&\mathbf{0.6}&-0.4&\mathbf{0.6}\\ -0.4&-0.4&-0.4&-0.4&\mathbf{0.6}&\mathbf{0.6}&\mathbf{0.6}&-0.4 \end{array}\right]}_{\boldsymbol{B} = \boldsymbol{A} - \langle \boldsymbol{A} \rangle \mathbf{1}_N}$$

Figure 2.2: Example of modularity maximization with a uniform null network. We show (first panel) the observed network, (second panel) the adjacency matrix $\boldsymbol{A}$ of the observed network, (third panel) the modularity matrix $\boldsymbol{B} = \boldsymbol{A} - \langle \boldsymbol{A} \rangle \mathbf{1}_N$ with a uniform null network rounded to the nearest decimal, and (fourth panel) the partition that optimizes modularity for this modularity matrix. Colors in the last panel represent community assignments.

where $\boldsymbol{A}$ is an unsigned adjacency matrix and $\mathbf{1}_N$ is an $N \times N$ matrix in which every entry is 1.[5] The expected edge weight in (2.11) is constant and satisfies

$$\frac{\langle k \rangle^2}{2m} = \frac{\left(\sum_{i=1}^N k_i/N\right)^2}{\sum_{i=1}^N k_i} = \frac{2m}{N^2} = \langle \boldsymbol{A} \rangle \,,$$

where $\langle \boldsymbol{A} \rangle$ denotes the mean value of the adjacency matrix.[6] One way to generate an unweighted network with adjacency matrix $\langle \boldsymbol{A} \rangle \mathbf{1}_N$ is to generate each edge with probability $\langle \boldsymbol{A} \rangle$ (provided $\langle \boldsymbol{A} \rangle \leq 1$). That is, the presence and absence of an edge (including self-edges) are independent and identically distributed (i.i.d.) Bernoulli random variables with probability $\langle \boldsymbol{A} \rangle$. More generally, any probability distribution on the set of adjacency matrices that satisfies $\mathbb{E}\left[\sum_{i,j=1}^N W_{ij}\right] = 2m$ and $\mathbb{E}[W_{ij}] = \mathbb{E}[W_{i'j'}]$ for all i, j, i', j' has an expected adjacency matrix of $\mathbb{E}[\boldsymbol{W}] = \langle \boldsymbol{A} \rangle \mathbf{1}_N$. The adjacency matrix of the U null network is symmetric and positive semidefinite.

2.1.3 Single-layer stability maximization

We now define a pair of quality functions that are based on the intuition that communities can be associated with "bottlenecks" of dynamical processes taking place on the observed network [52, 116].

[5] For a network in which all nodes have the same strength, the uniform and Newman-Girvan null networks are equivalent because $k_i = k_j$ for all $i, j \Leftrightarrow k_i = 2m/N = \langle k \rangle$ for all i. This was pointed out for an application to foreign exchange markets in [63].

[6] Although we use the uniform null network on unsigned adjacency matrices in this thesis, the expected edge weight in the uniform null network is always nonnegative for correlation matrices, as positive semidefiniteness guarantees that $\langle \boldsymbol{A} \rangle = (\mathbf{1}^T \boldsymbol{A} \mathbf{1})/(N^2) \geq 0$.

2.1.3.1 Discrete-time stability

Consider a discrete-time Markov process $(X_t), t \in \mathbb{N}$, on an observed network with adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, $A_{ij} \geq 0$. The probability density of a discrete-time Markov process satisfies [8, 130]

$$\boldsymbol{p}^{t+1} = \boldsymbol{p}^t \boldsymbol{M} \,, \tag{2.12}$$

where the vector $\boldsymbol{p}^t \in [0,1]^{1 \times N}$ is the probability density of a random walker at each node (i.e., $p_i^t := \mathbb{P}[X_t = i]$ for each i) and $\boldsymbol{M} \in [0,1]^{N \times N}$ is the transition matrix of a random walker (i.e., $M_{ij} := \mathbb{P}[X_{t+1} = j | X_t = i] = A_{ij}/k_i$). The stationary distribution of (2.12) is unique (provided the network is connected) and given by $\boldsymbol{\pi} = \boldsymbol{k}^T/(2m)$ [130]. *Discrete-time stability* is a quality function defined by [53, 54]

$$r(\boldsymbol{S}, t) = \mathrm{Tr}\left[\boldsymbol{S}^T \left(\boldsymbol{\Pi} M^t - \boldsymbol{\pi}^T \boldsymbol{\pi}\right) \boldsymbol{S}\right] \,,$$

where $\Pi_{ij} = \delta(i,j)\pi_i$. Equivalently, the discrete-time stability is

$$r(C, t) = \sum_{i,j=1}^{N} \left[\pi_i (M^t)_{ij} - \pi_i \pi_j\right] \delta(c_i, c_j) \,. \tag{2.13}$$

Taking $\boldsymbol{p}^0 = \boldsymbol{\pi}$, the first term in the square brackets on the right-hand side of (2.13) is $\mathbb{P}\big[(X_0 = i) \cap (X_t = j)\big]$ and the second term in the square brackets is $\lim_{t \to \infty} \mathbb{P}\big[(X_0 = i) \cap (X_t = j)\big]$ (provided the system is ergodic). The intuition behind the discrete-time stability quality function is that a good partition at a given time before reaching stationarity corresponds to one in which the time that a random walker spends within communities is large compared with the time that it spends transiting between communities. The resulting maximization problem is $\max_{\boldsymbol{S} \in \mathcal{S}} r(\boldsymbol{S}, t)$, or equivalently $\max_{C \in \mathcal{C}} r(C, t)$. By taking $t = 1$, one obtains the modularity-maximization problem in (2.7) with the NG null network $P_{ij} = k_i k_j/(2m)$.

2.1.3.2 Continuous-time stability

We now consider the continuous counterpart of discrete-time stability. Let $(X(t))$, $t \in \mathbb{R}^+$, be a continuous-time Markov process on an observed network with adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}, A_{ij} \geq 0$. The probability density of a continuous-time Markov process with exponentially distributed waiting times at each node parametrized by $\lambda(i)$ satisfies

$$\dot{\boldsymbol{p}} = \boldsymbol{p} \boldsymbol{\Lambda} \boldsymbol{M} - \boldsymbol{p} \boldsymbol{\Lambda} \,, \tag{2.14}$$

where the vector $\boldsymbol{p}(t) \in [0,1]^{1\times N}$ is the probability density of a random walker at each node [i.e., $p_i(t) := \mathbb{P}[X(t) = i]$ for each i], $\boldsymbol{\Lambda}$ is a diagonal matrix with the rate $\lambda(i)$ on its i^{th} diagonal entry, and $\boldsymbol{M} \in [0,1]^{N\times N}$ is the transition matrix of a random walker (i.e., $M_{ij} = A_{ij}/k_i$) [8,116]. The discrete-time random walker jumps from one node to the next at unit time intervals, whereas the continuous-time random walker waiting time is an exponentially-distributed random variable. However, whenever a jump occurs, the transition probabilities between nodes are identical for the two processes [53]. The solution to equation (2.14) is $\boldsymbol{p}(t) = \boldsymbol{p}_0 e^{\boldsymbol{\Lambda}(\boldsymbol{M}-\boldsymbol{I})t}$ and (provided the network is connected) its stationary distribution is unique and given by $\boldsymbol{\pi} = c\boldsymbol{k}^T\boldsymbol{\Lambda}^{-1}/(2m)$, where c is a normalization constant to ensure $\sum_{i=1}^{N} \pi_i = 1$. The *continuous-time stability* of a partition is a quality function defined by [53, 114–116]

$$r(\boldsymbol{S}, t) = \mathrm{Tr}\left[\boldsymbol{S}^T\left(\boldsymbol{\Pi} e^{\boldsymbol{\Lambda}(\boldsymbol{M}-\boldsymbol{I})t} - \boldsymbol{\pi}^T\boldsymbol{\pi}\right)\boldsymbol{S}\right] ,$$

where $\Pi_{ij} = \delta(i,j)\pi_i$. Equivalently, continuous-time stability is

$$r(C,t) = \sum_{i,j=1}^{N}\left[\pi_i\left(e^{\boldsymbol{\Lambda}(\boldsymbol{M}-\boldsymbol{I})t}\right)_{ij} - \pi_i\pi_j\right]\delta(c_i,c_j)\,. \tag{2.15}$$

Taking $\boldsymbol{p}_0 = \boldsymbol{\pi}$, the first term in the square brackets on the right-hand side of (2.15) is $\mathbb{P}\big[(X(0) = i) \cap (X(t) = j)\big]$, and the second term in the square brackets is $\lim_{t\to\infty}\mathbb{P}[(X(0) = i) \cap (X(t) = j)]$ (provided the system is ergodic). As with the discrete-time case, the intuition behind the continuous-time stability quality function is that a good partition at a given time before reaching stationarity corresponds to one in which the time that a random walker spends within communities is large compared with the time that it spends transiting between communities. The resulting maximization problem is $\max_{\boldsymbol{S}\in\mathcal{S}} r(\boldsymbol{S}, t)$, or equivalently $\max_{C\in\mathcal{C}} r(C,t)$.

The multiscale modularity-maximization problem,

$$\max_{C\in\mathcal{C}}\sum_{i,j=1}^{N}(A_{ij} - \gamma P_{ij})\delta(c_i,c_j)\,,$$

in (2.6) was initially introduced in [167] using an *ad hoc* approach. Interestingly, one can derive this formulation of the maximization problem for sufficiently large values of γ from the continuous-time stability quality function in (2.15). By linearizing $e^{\boldsymbol{\Lambda}(\boldsymbol{M}-\boldsymbol{I})t}$ at $t = 0$ and taking $\boldsymbol{\Lambda} = \boldsymbol{I}$, one obtains the multiscale modularity-maximization problem in (2.6) at short timescales with $\gamma = 1/t$ and $P_{ij} = k_ik_j/(2m)$. This approach provides a dynamical interpretation of the resolution parameter γ as the inverse (after linearization) of the time used to explore a network by a random walker. One can derive the multiscale formulation of modularity for the U null network (i.e., $P_{ij} = \langle \boldsymbol{A} \rangle$) from continuous-time stability in precisely the same way as it is derived for the NG null network, except that one needs to consider exponentially-distributed waiting times at each node with rates proportional to node strength (i.e., $\Lambda_{ij} = \delta(i,j)k_i/\langle k \rangle$) [116].

The authors of [140] derived a variant of the multiscale formulation of modularity for the NGS null network (i.e., $P_{ij} = k_i^+k_j^+/(2m^+) - k_i^-k_j^-/(2m^-)$) at short time scales by building on the approach described in the previous paragraph for deriving the NG and U null networks from continuous-time stability.[7] They considered the function

$$\hat{r}(C,t) = \sum_{i,j=1}^{N} \left(\pi_i \big[\delta_{ij} + t\Lambda_{ii}(M_{ij} - \delta_{ij})\big] - \pi_i \rho_{i|j} \right) \delta(c_i, c_j) \,, \tag{2.16}$$

where the term in square brackets on the right-hand side of (2.16) is a linearization of the exponential term in (2.15), $\boldsymbol{M}$ and π_i are the transition matrix and stationary distribution defined in (2.15) on a network with adjacency matrix $|\boldsymbol{A}| := \boldsymbol{A}^+ + \boldsymbol{A}^-$, and $\rho_{i|j}$ is the probability of jumping from node i to node j at stationarity in one step conditional on the network structure [140]. If the network is non-bipartite, unsigned, and undirected, then $\rho_{i|j}$ reduces to the stationary probability π_j.

2.1.4 Single-layer map minimization

The final single-layer optimization approach that we describe is the *map equation*. The map equation was introduced by Rosvall *et al.* in [171]. It uses a discrete-time random walk and ideas from coding theory to coarse-grain a network into communities [47, 170, 171]. A key concept in the map equation is Shannon entropy [126, 181].

Consider a random variable X with discrete finite support $x \in \mathcal{X}$ and probability distribution $\boldsymbol{p} \in [0,1]^{|\mathcal{X}|\times 1}$. The *Shannon entropy* of X is [126, 181]

$$H(\boldsymbol{p}) = -\sum_{i=1}^{|\mathcal{X}|} p_i \log_2(p_i) \,. \tag{2.17}$$

Entropy quantifies the "unevenness" of a probability distribution [126]. It represents the "average information" required to specify an outcome of a random variable given

[7] In particular, they derived the multiscale formulation of modularity obtained using a Potts-model approach from [189]. This multiscale formulation results in one resolution parameter γ_1 for the term $(k_i^+k_j^+)/(2m^+)$ and a second resolution parameter γ_2 for the term $(k_i^-k_j^-)/(2m^-)$ in (2.10) (see [132] for an application of this multiscale formulation to United Nations General Assembly voting networks). Without an application-driven justification for how to choose these parameters, this increases the parameter space substantially, so we only consider the case $\gamma_1 = \gamma_2$ in this thesis.

its probability distribution. It achieves its minimal value 0 for a constant random variable (i.e., $\mathbb{P}[X = \tilde{x}] = 1$ and $\mathbb{P}[X \neq \tilde{x}] = 0$) and its maximal value $\log_2(|\mathcal{X}|)$ for a uniform distribution.

Now, consider a discrete-time Markov process (X_t), $t \in \mathbb{N}$, on an observed network with probability density

$$\boldsymbol{p}^{t+1} = \boldsymbol{p}^t \boldsymbol{M}\,,$$

where $\boldsymbol{p}^t \in [0,1]^{1\times N}$ is the probability density of a random walker at each node and $\boldsymbol{M} \in [0,1]^{N\times N}$ (with $M_{ij} = A_{ij}/k_i$) is the transition matrix of a random walker. Denote the stationary distribution of the Markov process by $\boldsymbol{\pi}$ and assume that $\boldsymbol{p}^0 = \boldsymbol{\pi}$. Consider a *binary encoding* of the dynamic (X_t) that maps each node to a *binary codeword*, which is a sequence of 1s and 0s. For example, one could assign binary codewords of equal length to each node, or one could compress the description length of the encoding by assigning shorter binary codewords to frequently-visited nodes and longer binary codewords to rarely-visited nodes (this approach is known as the "Huffman code" [94, 171]). Shannon showed that the expected per step average length of an optimal binary code for the random walker dynamic is bounded below by the *entropy rate*

$$\lim_{t\to\infty} \frac{1}{t} H(X_1, \ldots, X_t) = \sum_{i=1}^{N} \pi_i H(\boldsymbol{m}_i)\,, \tag{2.18}$$

where $H(X_1, \ldots, X_t)$ is the entropy of the joint distribution of $X_1, \ldots, X_t$ and $\boldsymbol{m}_i$ is the i^{th} row of the transition matrix $\boldsymbol{M}$ (we adopt the convention that $0 \log_2 0 = 0$) [33, 126, 181].

The idea behind the map equation is to incorporate the structure of a network into the coding scheme. That is, instead of using a unique codeword for each node, one uses a unique codeword for each community, a unique codeword for each node within a community (these are *repeated* across different communities, much like street names are repeated across different US cities in a map [171]), and a unique codeword to indicate that a random walker has exited a community (one "exit codeword" per community) [171, 172].[8] Note that this coding scheme allows for a community to have the same binary codeword as a node and the authors use an exit codeword to signal when the next codeword refers to a community and not to a node.

Consider an N-node partition $C \in \mathcal{C}$, with $|C| = K$ and C_k an arbitrary set in C. The authors of [172] use the term *module codebook* to describe a random walker's

[8] Assigning a unique codeword to every node in a network is a special case of this coding scheme that corresponds to a partition into a single community.

movement within a set (the exit probability is contained in the module codebook) and the term *index codebook* to describe a random walker's movement between sets. Denote by p_{out}^k the probability that a random walker in stationary distribution leaves a set C_k

$$p_{\text{out}}^k := \mathbb{P}[(X_t \in C_k) \cap (X_{t+1} \notin C_k)] = \sum_{i \in C_k} \sum_{j \notin C_k} \pi_i M_{ij}\,,$$

and by p_{in}^k the probability that a random walker in stationarity distribution leaves a set C_k or is in set C_k

$$p_{\text{in}}^k := p_{\text{out}}^k + \mathbb{P}[X_t \in C_k] = p_{\text{out}}^k + \sum_{i \in C_k} \pi_i\,.$$

The term p_{out}^k is used in the index codebook and the term p_{in}^k is used in the module codebook. One can think of the first term in p_{in}^k as the probability that a random walker is at an "exit node", and the second term in p_{in}^k is the probability that a random walker is at a node in a community C_k (both terms are computed at stationarity). The map equation is a lower bound on the expected per step average length of an optimal binary code for the random walker dynamic using this coding scheme and it is given by [33, 171, 172]

$$L(C) = H(\mathcal{P}_{\text{out}}) \sum_{k=1}^{K} p_{\text{out}}^k + \sum_{k=1}^{K} p_{\text{in}}^k H(\mathcal{P}_{\text{in}}^k)\,, \tag{2.19}$$

a weighted combination of the Shannon entropies

$$\begin{aligned} H(\mathcal{P}_{\text{out}}) &= -\sum_{k=1}^{K} \left(\frac{p_{\text{out}}^k}{\sum_{k=1}^{K} p_{\text{out}}^k} \right) \log_2 \left(\frac{p_{\text{out}}^k}{\sum_{k=1}^{K} p_{\text{out}}^k} \right), \\ H(\mathcal{P}_{\text{in}}^k) &= -\frac{p_{\text{out}}^k}{p_{\text{in}}^k} \log_2 \left(\frac{p_{\text{out}}^k}{p_{\text{in}}^k} \right) - \sum_{i \in C_k} \frac{\pi_i}{p_{\text{in}}^k} \log_2 \left(\frac{\pi_i}{p_{\text{in}}^k} \right), \end{aligned}$$

where $\mathcal{P}_{\text{out}} \in [0,1]^{K \times 1}$ is the probability distribution with k^{th} entry $p_{\text{out}}^k / (\sum_{k=1}^{K} p_{\text{out}}^k)$ and $\mathcal{P}_{\text{in}}^k \in [0,1]^{(1+|C_k|) \times 1}$ is the probability distribution given by $\big[p_{\text{out}}^k / p_{\text{in}}^k, \pi_{i_1}/p_{\text{in}}^k, \ldots,$ $\pi_{i_{|C_k|}}/p_{\text{in}}^k\big]$, where $i_1, \ldots, i_{|C_k|}$ are the nodes in C_k.

The intuition behind the map equation is that a random walker tends to get trapped within communities and one can thereby shorten a binary encoding of a random walker's dynamic by capitalizing on a network's community structure and assigning short codewords to nodes inside of each community. This motivates the minimization problem $\min_{C \in \mathcal{C}} L(C)$. There exist various extensions of the map equation (e.g., hierarchical structure [173], overlapping community structure [59], multiscale reformulation [179]), discussions of its "resolution-limit" [106], and discussions of some of its limitations in [179]. In the present thesis, we only use the map equation in Section 5.3.2 where we perform preliminary experiments to compare the performance of multilayer map minimization (which we describe in Section 2.2.3) to multilayer modularity maximization (which we describe in Section 2.2.2) on our temporal multilayer benchmark.

2.2 Community detection in temporal networks

2.2.1 Overview

Most community-detection methods are designed for time-independent networks. However, in many applications, entities and/or interactions between entities evolve in time. In such applications, one can use the formalism of *temporal networks*, where nodes and/or their edge weights vary in time [89, 90].

Two main approaches have been adopted to detect communities in time-dependent networks. The first entails constructing a time-independent network by aggregating snapshots of the evolving network at different points in time into a single network (e.g., by taking the mean or total edge weight for each edge across all time points, which can be problematic if the set of nodes varies in time and which also makes restrictive assumptions on the interaction dynamics between entities [86]). One can then use standard network techniques. The second approach entails using time-independent community-detection techniques on each element of a time-ordered sequence of networks at different times or on each element of a time-ordered sequence of network aggregations[9] (computed as above) over different time intervals (which can be either overlapping or nonoverlapping) and then tracking the communities across the sequence [63, 64, 91, 132, 156].

A third approach consists of embedding a time-ordered sequence of networks in a larger network [26,56,140]. Each element of the sequence is a network *layer*, and nodes at different time points are joined by *inter-layer* edges. This approach was introduced in [140] and the resulting network is a type of *multilayer network* [31, 109]. The main difference between this approach and the previous approach is that the presence of nonzero inter-layer edges introduces a dependence between communities identified in one layer and connectivity patterns in other layers.

[9] One needs to distinguish between this kind of aggregation and the averaging of a set of time series over a moving window to construct a correlation matrix, which one can then interpret as a fixed-time snapshot of a time-evolving network. Although both involve averaging over a time window, the former situation entails averaging a network, and the latter situation entails averaging over a collection of time series (one for each node) with no directly observable edge weights.

There are also community detection techniques for identifying clusters in a *single* snapshot while accounting for connectivity patterns in *other* snapshots in some way (e.g., by defining the transition matrix of a random walker in a snapshot as the right-multiplication [to respect the arrow of time] of transition matrices up to and including the snapshot [161] or by using a partition into communities of a previous snapshot as an initial partition when detecting communities for the current snapshot algorithmically [15,187]). Various examples of such techniques can be found in [5,85]. There has also been a recent growth in the literature of stochastic block model generalizations from single-layer networks to a time-ordered sequence (or more generally, a correlated sequence) of single-layer networks [83,158,160,193,203]. As with single-layer networks, these methods can be used to generate a sequence of networks with some planted structure or to infer structure from an observed sequence of networks by fitting it to a stochastic block model. We discuss an instance of the former in Chapter 5 but do not discuss the latter in this thesis.

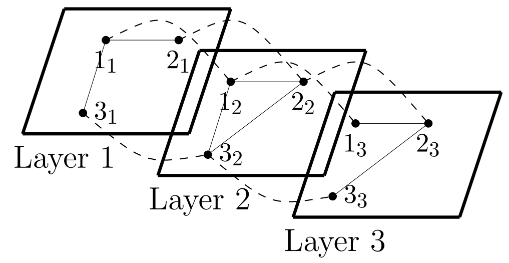
Throughout the thesis, we focus primarily on a generalization of the modularity quality function to multilayer networks [140] and we also carry out preliminary numerical experiments with a generalization of the map equation to multilayer networks [56] in Section 5.3.2. Both generalizations yield a partition of a multilayer network (as opposed to a partition of a single-layer network), and they are the focus of the next two sections.

2.2.2 Multilayer modularity maximization

2.2.2.1 Multilayer representation of temporal networks

We restrict our attention to temporal networks in which only edges vary in time. (Thus, each node is present in all layers.) We use the notation $\boldsymbol{A}_s$ for the adjacency matrix of a network layer in a sequence of adjacency matrices $\mathcal{T} = \{\boldsymbol{A}_{\mathbf{1}}, \dots, \boldsymbol{A}_{|\mathcal{T}|}\}$, and we denote node i in layer s by i_s. We use the term *multilayer network* for a network defined on the set of nodes $\{1_1, \dots, N_1; 1_2, \dots, N_2; \dots; 1_{|\mathcal{T}|}, \dots, N_{|\mathcal{T}|}\}$ [109].

Thus far, computations that have used a multilayer framework for temporal networks have almost always assumed (1) that inter-layer connections exist only between nodes that correspond to the same entity (i.e., between nodes i_s and i_r for some i and $s \neq r$) and (2) that the network layers are "ordinal" (i.e., inter-layer edges exist only between consecutive layers) [22,109,140,141,166]. It is also typically assumed that (3) inter-layer connections are uniform (i.e., inter-layer edges have the same weight). In a recent review article on multilayer networks [109], condition (1) was called "diagonal"



$$\longleftrightarrow \left[\begin{array}{ccc|ccc|ccc}
0 & 1 & 1 & \omega & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & \omega & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & \omega & 0 & 0 & 0 \\
\hline
\omega & 0 & 0 & 0 & 1 & 1 & \omega & 0 & 0 \\
0 & \omega & 0 & 1 & 0 & 1 & 0 & \omega & 0 \\
0 & 0 & \omega & 1 & 1 & 0 & 0 & 0 & \omega \\
\hline
0 & 0 & 0 & \omega & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & \omega & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & \omega & 0 & 1 & 0
\end{array}\right]$$

Figure 2.3: Example of (left) a multilayer network with unweighted intra-layer connections (solid lines) and uniformly weighted inter-layer connections (dashed curves) and (right) its corresponding adjacency matrix. (The adjacency matrix that corresponds to a multilayer network is sometimes called a "supra-adjacency matrix" in the network-science literature [109].)

coupling, and condition (2) implies that a network is "layer-coupled". We refer to the type of coupling defined by (1), (2), and (3) as *ordinal diagonal* and *uniform inter-layer coupling* and we denote the value of the inter-layer edge weight by $\omega \in \mathbb{R}$. We show a simple illustration of a multilayer network with ordinal diagonal and uniform inter-layer coupling in Fig. 2.3. One can consider more general inter-layer connections (e.g., nonuniform ones). Although we restrict our attention to uniform coupling in our theoretical and computational discussions, we give an example of a nonuniform choice of inter-layer coupling in Section 4.2. Results similar to those of Section 4.2.2 also apply in this more general case.

2.2.2.2 The multilayer modularity function

The authors of [140] generalized the single-layer multiscale modularity-maximization problem

$$\max_{C \in \mathcal{C}} \sum_{i,j=1}^{N} (A_{ij} - \gamma P_{ij}) \delta(c_i, c_j) \,,$$

in (2.6) to a multilayer network using a similar approach as the one used to derive the NGS null network from a continuous-time stochastic Markov process on the observed network in (2.16). That is, the authors of [140] linearize the exponential term in the continuous-time stability quality function

$$r(C,t) = \sum_{i,j} \left[\pi_i \left(e^{\boldsymbol{\Lambda}(\boldsymbol{M}-\boldsymbol{I})t} \right)_{ij} - \pi_i \pi_j \right] \delta(c_i, c_j) \,, \tag{2.20}$$

and modify the normalization constant (i.e., the right-hand side term in square brackets in equation (2.20)) for the network of interest. For simplicity, we express intra-layer and inter-layer connections in an $N|\mathcal{T}|$-node multilayer network using a single $N|\mathcal{T}| \times N|\mathcal{T}|$ matrix. Each node i_s in layer s has the unique index $i' := i + (s-1)N$, and we use $\boldsymbol{\mathcal{A}}$ to denote the multilayer adjacency matrix, which has entries $\mathcal{A}_{i'j'} = A_{ijs}\delta(s,r) + \omega\delta(|s-r|,1)$ when the inter-layer coupling is ordinal, diagonal, and uniform. The generalization in [140] consists of applying the following function to the $N|\mathcal{T}|$-node multilayer network:

$$\hat{r}(C,t) = \sum_{i,j=1}^{N|\mathcal{T}|} \left(\pi_i \big[\delta_{ij} + t\Lambda_{ii}(\mathcal{M}_{ij} - \delta_{ij})\big] - \pi_i \rho_{i|j}\right)\delta(c_i, c_j)\,, \tag{2.21}$$

where C is now a multilayer partition (i.e., a partition of an $N|\mathcal{T}|$-node multilayer network), $\boldsymbol{\Lambda}$ is the $N|\mathcal{T}| \times N|\mathcal{T}|$ diagonal matrix with the rates of the exponentially distributed waiting times at each node of each layer on its diagonal, $\boldsymbol{\mathcal{M}}$ (with entries $\mathcal{M}_{ij} := \mathcal{A}_{ij}/\sum_{k=1}^{N|\mathcal{T}|}\mathcal{A}_{ik}$) is the $N|\mathcal{T}| \times N|\mathcal{T}|$ transition matrix for the $N|\mathcal{T}|$-node multilayer network with adjacency matrix $\boldsymbol{\mathcal{A}}$, π_i is the corresponding stationary distribution (with the strength of a node and the total edge weight now computed from the multilayer adjacency matrix $\boldsymbol{\mathcal{A}}$), and $\rho_{i|j}$ is the probability of jumping from node i to node j at stationarity in one step conditional on the structure of the network within and between layers. Movement within layers is conditional on nodes' intra-layer strength (i.e., sum of edge weights incident to a node with endpoints in the same layer) and inter-layer strength (i.e., sum of edge weights incident to a node with endpoints in a different layer). Movement between layers is conditional on nodes' intra-layer strength, inter-layer strength, and the specific "sparsity pattern"[10] of inter-layer edges. The authors' choice of $\rho_{i|j}$ motivates the following *multilayer modularity-maximization problem*:

$$\max_{C\in\mathcal{C}} \sum_{i,j=1}^{N|\mathcal{T}|} \mathcal{B}_{ij}\delta(c_i, c_j)\,, \tag{2.22}$$

which we can also write as $\max_{C\in\mathcal{C}} Q(C|\boldsymbol{\mathcal{B}})$ or $\max_{C\in\mathcal{C}} Q(C|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; \omega)$, where $\mathcal{C}$ is the set of all $N|\mathcal{T}|$-node partitions, $\boldsymbol{\mathcal{B}}$ is the *multilayer modularity matrix*

$$\boldsymbol{\mathcal{B}} = \begin{bmatrix} \boldsymbol{B}_1 & \omega\boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \omega\boldsymbol{I} & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \ddots & \ddots & \ddots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \omega\boldsymbol{I} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \omega\boldsymbol{I} & \boldsymbol{B}_{|\mathcal{T}|} \end{bmatrix}, \tag{2.23}$$

[10] The *sparsity pattern* of a matrix $\boldsymbol{X}$ is a matrix $\boldsymbol{Y}$ with entries $Y_{ij} = 1$ when $X_{ij} = 0$ and $Y_{ij} = 0$ when $X_{ij} = 0$.

and $\boldsymbol{B}_s$ is a single-layer modularity matrix computed on layer s. (For example, $\boldsymbol{B}_s = \boldsymbol{A}_s - \langle \boldsymbol{A}_s \rangle \boldsymbol{1}_N$ if one uses the U null network and sets $\gamma = 1$.) We rewrite the multilayer modularity-maximization problem in [140] as

$$\max_{C \in \mathcal{C}} \left[\sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs} \delta(c_{i_s}, c_{j_s}) + 2\omega \sum_{s=1}^{|\mathcal{T}|-1} \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_{s+1}}) \right] , \tag{2.24}$$

where c_{i_s} is the integer community assignment of node i in layer s and B_{ijs} denotes the $(i,j)^{\text{th}}$ entry of $\boldsymbol{B}_s$. Equation (2.24) clearly separates intra-layer contributions (left term) from inter-layer contributions (right term) to the multilayer quality function.

To date, almost no theory has explained how a multilayer partition obtained with zero inter-layer coupling (which reduces to single-layer modularity maximization on each layer independently) differs from a multilayer partition obtained with nonzero inter-layer coupling. In Section 4.2, we try to gain some insight into how to interpret a globally optimal multilayer partition by proving several properties that it satisfies. The results that we show hold for *any* choice of matrices $\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}$, so (for example) they also apply when one uses the stability quality function in (2.15) or (2.13) on each layer instead of the modularity quality function. For ease of writing (and because modularity is the quality function that we use in most of our computational experiments of Chapters 4, 5, and 6), we will continue to refer to the maximization problem (2.24) as a multilayer modularity maximization problem.

It is clear from (2.24) that placing nodes from different layers into the same set, which we refer to as an inter-layer *merge*, always decreases the value of the multilayer quality function when $\omega < 0$, so we only consider $\omega \geq 0$ in the present thesis. Furthermore, although merging pairs of nodes with $\mathcal{B}_{ij} = 0$ does not change the value of multilayer modularity, we assume in our theoretical discussion of Section 4.2 that the set $\mathcal{C}$ of $N|\mathcal{T}|$-node partitions does not contain partitions into sets with multiple connected components in the $N|\mathcal{T}|$-node weighted graph with adjacency matrix $\mathcal{B}$.

We end this section with some notation that will be useful for comparing single-layer partitions to a multilayer partition. We denote by $\mathcal{N}_s := \{1_s, \ldots, N_s\}$ the set of nodes in layer s. The restriction of a set of nodes

$$C_k \subseteq \{1_1, \ldots, N_1; 1_2, \ldots, N_2; \ldots ; 1_{|\mathcal{T}|}, \ldots, N_{|\mathcal{T}|}\}$$

to a layer s is $C_k|_s := C_k \cap \mathcal{N}_s$, and we define the *partition induced by a multilayer partition* $C \in \mathcal{C}$ on layer s by

$$C|_s := \{C_k|_s, C_k \in C\} .$$

We will use this notation repeatedly in Section 4.2.

2.2.3 Multilayer map minimization

DeDomenico *et al.* proposed a generalization of the map equation to multilayer networks in [56]. Their generalization consists of making two changes to the single-layer map equation: (1) they modified the transition probabilities in the discrete-time random walk and (2) they modified the coding scheme in the module codebook in a way that accounts for multilayer structure. We describe these changes below.

Consider a discrete-time Markov process (X_t), $t \in \mathbb{N}$, on the $N|\mathcal{T}|$-node multilayer network. The transition probabilities of (X_t) are defined in [56] by

$$\mathbb{P}[X^{t+1} = j_l | X^t = i_s] := (1-r)\delta_{sl} \frac{A_{ijl}}{\sum_{k=1}^{N} A_{ikl}} + r \frac{A_{ijl}}{\sum_{m=1}^{|\mathcal{T}|} \sum_{k=1}^{N} A_{ikm}}\,, \tag{2.25}$$

where $s, l \in \{1, \dots, |\mathcal{T}|\}$, $r \in [0,1]$ is termed the *relax-rate*, and $\boldsymbol{A}_s$ is the adjacency matrix of the s^{th} layer with $(i,j)^{\text{th}}$ entry A_{ijs}. Equation (2.25) gives the probability that a random walker at node i_s jumps to a node in layer s (first term on the right-hand side of (2.25)) or to a node in layer l (second term on the right-hand side of (2.25)). The random walker can move along a weighted intra-layer edge connected to i_s in layer s (first term on the right-hand side of (2.25)) or along a weighted intra-layer edge connected to i_l in layer l (second term on the right-hand side of (2.25)). One consequence of equation (2.25) is that a random walker cannot transition from i_s to i_l, $s \neq l$, in a single jump (unless the single-layer networks contain self-loops).

One can also write the transition probabilities in (2.25) as

$$\boldsymbol{\mathcal{M}} = (1-r)\begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{M}_{|\mathcal{T}|} \end{bmatrix} + r \begin{bmatrix} \hat{\boldsymbol{M}}_1 & \hat{\boldsymbol{M}}_2 & \dots & \hat{\boldsymbol{M}}_{|\mathcal{T}|} \\ \hat{\boldsymbol{M}}_1 & \hat{\boldsymbol{M}}_2 & \dots & \hat{\boldsymbol{M}}_{|\mathcal{T}|} \\ \vdots & \vdots & & \vdots \\ \hat{\boldsymbol{M}}_1 & \hat{\boldsymbol{M}}_2 & \dots & \hat{\boldsymbol{M}}_{|\mathcal{T}|} \end{bmatrix}, \tag{2.26}$$

where $\boldsymbol{\mathcal{M}} \in [0,1]^{N|\mathcal{T}| \times N|\mathcal{T}|}$ is the transition matrix of (X_t), $\boldsymbol{M}_s \in [0,1]^{N \times N}$ has entries $M_{ijs} = A_{ijs}/\left(\sum_{k=1}^{N} A_{iks}\right)$, and $\hat{\boldsymbol{M}}_s \in [0,1]^{N \times N}$ has entries $\hat{M}_{ijs} = A_{ijs}/\left(\sum_{l=1}^{|\mathcal{T}|} \sum_{k=1}^{N} A_{ikl}\right)$. When $r = 0$, the transition matrix $\boldsymbol{\mathcal{M}}$ reduces to a block-diagonal matrix with single-layer transition matrices on its diagonal blocks. As one increases r, the random walker can "move more freely on the fully aggregated network" [56]. To our knowledge, theoretical properties of the multilayer map equation as one varies the relax-rate r remain unexplored.

To incorporate multilayer structure into the coding scheme, the authors of [56] make the following change to the module codebook. In addition to having a unique codeword for each node in a set of a partition, the authors assign *the same* codeword to copies of the same node in a set. That is, a set of nodes $\{i_{s_1}, \dots, i_{s_k}\}$ that belong to the same set in a partition receive the same codeword, where $\{s_1, \dots, s_k\} \subseteq \{1, \dots, \mathcal{T}\}$. The authors then define the multilayer map equation in a similar way to how they define the single-layer map equation. We denote by $N(C_k)$ the set of nodes in $\{1, \dots, N\}$ with at least one copy in C_k (i.e., $\{i \in \{1, \dots, N\} : i_s \in C_k \text{ for some } s\}$). The multilayer map equation is given by

$$L(C) = H(\mathcal{P}_{\text{out}}) \sum_{k=1}^{K} p_{\text{out}}^k + \sum_{k=1}^{K} p_{\text{in}}^k H(\mathcal{P}_{\text{in}}^k) \,, \tag{2.27}$$

a weighted combination of the Shannon entropies

$$\begin{aligned} H(\mathcal{P}_{\text{out}}) &= -\sum_{k=1}^{K} \left(\frac{p_{\text{out}}^k}{\sum_{k=1}^{K} p_{\text{out}}^k} \right) \log_2 \left(\frac{p_{\text{out}}^k}{\sum_{k=1}^{K} p_{\text{out}}^k} \right) , \\ H(\mathcal{P}_{\text{in}}^k) &= -\frac{p_{\text{out}}^k}{p_{\text{in}}^k} \log_2 \left(\frac{p_{\text{out}}^k}{p_{\text{in}}^k} \right) - \sum_{i \in N(C_k)} \frac{\left[\sum_{i_s \in C_k} \pi_{i_s} \right]}{p_{\text{in}}^k} \log_2 \left(\frac{\left[\sum_{i_s \in C_k} \pi_{i_s} \right]}{p_{\text{in}}^k} \right) , \end{aligned}$$

where C is now an $N|\mathcal{T}|$-node partition, π_i is the stationary distribution of node $i \in \{1, \dots, N|\mathcal{T}|\}$, $\pi_{i_s} = \pi_{i'}$ for $i' = i + (s-1)N$, p_{out}^k is given by

$$p_{\text{out}}^k := \sum_{i \in C_k} \sum_{j \notin C_k} \pi_i \mathcal{M}_{ij} \,,$$

and p_{in}^k is given by

$$p_{\text{in}}^k := p_{\text{out}}^k + \sum_{i \in C_k} \pi_i \,.$$

We only use multilayer map minimization in the numerical experiments of Section 5.3.2. We note that this generalization was developed for a multilayer network where different layers represent different *types* of connections between a set of nodes (e.g., interactions on Facebook and interactions on Twitter) rather than a specific type of connection at different points in time. To try and account for this, we consider two cases in our numerical experiments: (1) the case where a random walker can "relax" to all layers (i.e., the transition matrix is given by (2.26)) and (2) the case where a random walker can only "relax" to an adjacent layer (i.e., the transition matrix is given by the diagonal blocks and the first off-diagonal blocks in (2.26) — all other blocks are set to zero and the normalization constant is adjusted accordingly). We use publicly available code from [1] for our experiments and in particular, we do not discuss heuristics for solving the multilayer map minimization problem in this thesis.

2.3 Computational heuristics

2.3.1 Overview

We describe a few heuristics that one can use for solving the single-layer modularity maximization problem first stated in (2.3), namely

$$\max_{C \in \mathcal{C}} \sum_{i,j=1}^{N} B_{ij}\delta(c_i, c_j)\,,$$

where $\mathcal{C}$ is the the set of N-node partitions and $\boldsymbol{B} \in \mathbb{R}^{N \times N}$ is the single-layer modularity matrix,[11] and we focus on two heuristics that trivially extend to the multilayer modularity maximization problem stated in (2.22), namely

$$\max_{C \in \mathcal{C}} \sum_{i,j=1}^{N|\mathcal{T}|} \mathcal{B}_{ij}\delta(c_i, c_j)\,,$$

where $\mathcal{C}$ is the set of $N|\mathcal{T}|$-node partitions and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{N|\mathcal{T}| \times N|\mathcal{T}|}$ is the multilayer modularity matrix given in (2.23) by

$$\boldsymbol{\mathcal{B}} = \begin{bmatrix} \boldsymbol{B}_1 & \omega \boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \omega \boldsymbol{I} & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \ddots & \ddots & \ddots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \omega \boldsymbol{I} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \omega \boldsymbol{I} & \boldsymbol{B}_{|\mathcal{T}|} \end{bmatrix}.$$

For a given modularity matrix $\boldsymbol{B}$, a solution to the modularity-maximization problem is guaranteed to exist in any network with a finite number of nodes. However, the number of possible partitions in an N-node network, given by the Bell number [24], grows at least exponentially with N so an exhaustive search of the space of partitions is infeasible. Modularity maximization was shown in [37] to be an NP-hard problem (at least for the null networks which we consider in this paper), so solving it requires the use of computational heuristics, which we refer to as "heuristics" in the rest of the thesis. Various heuristics have been suggested for solving the modularity-maximization problem. We briefly describe four common approaches in this section.

A first approach is *greedy* heuristics, which perform locally optimal moves at each update until a (usually) local optimum is reached. One of the first greedy heuristics was suggested by Newman in [145]. It is an agglomerative hierarchical procedure that

[11] Most of the heuristics that we mention in this section were developed for the NG null network in (2.7) and not for an arbitrary null network.

iteratively merges sets of nodes based on the merge's contribution to the modularity function. The complexity of Newman's implementation is $O((m_0+N)N)$, where m_0 is the number of edges with nonzero edge weights in the network [68]. Clauset *et al.* [44] suggested a faster implementation of Newman's algorithm by making use of more efficient data structures. The complexity of Clauset's implementation is $O(m_0 d \log N)$ for an arbitrary network, where d is the "depth" of the dendrogram [44, 68]. Clauset's heuristic has performed poorly on some benchmark tests [118] and the authors of [195] observed that it often runs at its worst time complexity. A third locally greedy heuristic is the *Louvain* heuristic [29], a local modularity-increasing sampling process on the space of partitions. The Louvain heuristic is a popular choice in practice [118]. It is claimed that most of its computational time is usually spent in the first iteration of its first phase and so its complexity tends to be $O(m_0)$ [40, 68]. We describe this heuristic in detail in Section 2.3.2

A second type of heuristic first introduced in [108] and later extended to the modularity quality function by Guimerà *et al.* [81] is *simulated annealing.* An important difference between this heuristic and greedy heuristics is that simulated annealing allows modularity-*decreasing* moves. This can help prevent a heuristic from getting trapped in local optima (a common problem with greedy heuristics [9, 30]). Simulated annealing is an iterative procedure that depends on a *temperature* parameter $\hat{T}$. Every move at each iteration of simulated annealing is accepted with probability 1 if it increases the quality function and with probability $\exp(\Delta Q/\hat{T})$ if it decreases the quality function, where ΔQ is the difference between the current modularity value of the network and the modularity value of the network if one performs the update. Simulated annealing can come very close to an optimal solution, but it is slow (its precise complexity depends on the parameters chosen for the optimization) [68].

A third type of heuristic first introduced in [32] and later applied to modularity maximization in [57] is *extremal optimization.* The aim of the authors in [32] was to achieve an accuracy comparable to that of simulated annealing, but with a substantial reduction in computational time [68]. The heuristic starts at a random bipartition of the network (i.e., a partition into two sets) and iteratively bipartitions the network by performing locally optimal moves (using a "fitness" measure to quantify a node's contribution to modularity) [40, 68]. Its complexity is $O(N^2 \log N)$.

A fourth approach is *spectral heuristics.* Broadly speaking, spectral heuristics reformulate the discrete optimization problem as a continuous optimization problem by embedding the modularity-maximization problem in $\mathbb{R}^{N\times K}$. One can then use the spectrum of the modularity matrix (or the spectrum of a related matrix [199]) to solve the optimization problem in $\mathbb{R}^{N\times K}$ and approximate an assignment matrix in $\{0,1\}^{N\times K}$ from the obtained solution [146, 169, 196, 209]. More heuristics based on a relaxation of the modularity-maximization problem to a subset of $\mathbb{R}^{N\times K}$ can be found in [4,55,92]. In this thesis we discuss some features of a *spectral bipartitioning heuristic* suggested by Newman in [146]. This heuristic uses the sign of the leading eigenvector of $\boldsymbol{B}$ to bipartition a network and iterates this procedure on each set of the bipartition until no further improvement is possible. Its complexity is $O((N+m_0)N)$ [40, 146].

The next two sections give a description of the Louvain heuristic and the spectral bipartitioning heuristic. The main reasons we investigate these heuristics is because they scale well with the size of a network (which is an important consideration in a multilayer network, in which the number of nodes is $N|\mathcal{T}|$), they trivially extend to the multilayer modularity-maximization problem, and they can be used for an *arbitrary* choice of null network. Given a modularity matrix, we note that a heuristic yields a partition of a network whether or not the underlying network "truly" contains community structure. How to assess the statistical significance of an identified partition is an actively studied (and unresolved) research question [21, 80, 95, 104, 122, 135, 167]. We do not address this issue in this thesis, and the computational issues that we discuss in Section 4.2.3 hold whether or not a network "truly" contains community structure.

2.3.2 The Louvain computational heuristic

The Louvain heuristic consists of two phases, which are repeated iteratively. Initially, each node in the network constitutes a set, which gives an initial partition that consists of N singletons. During phase 1, one considers the nodes one by one (in some order), and one places each node in a set (including its own) that results in the largest increase of modularity. This phase is repeated until one reaches a local maximum (i.e., until one obtains a partition in which the move of a single node cannot increase modularity). Phase 2 consists of constructing a reduced network $\mathcal{G}'$ from the sets of nodes in $\mathcal{G}$ that one obtains after the convergence of phase 1. We denote the sets in $\mathcal{G}$ at the end of phase 1 by $\{\hat{C}_1,\dots,\hat{C}_{\hat{N}}\}$ (where $\hat{N}\leq N$) and the set assignment of node i in this partition by $\hat{c}_i$. Each set $\hat{C}_k$ in $\mathcal{G}$ constitutes a node k in $\mathcal{G}'$, and the reduced modularity matrix of $\mathcal{G}'$ is

$$\boldsymbol{B}'=\hat{\boldsymbol{S}}^{\boldsymbol{T}}\boldsymbol{B}\hat{\boldsymbol{S}}\,,$$

where $\hat{\boldsymbol{S}}$ is the assignment matrix of $\{\hat{C}_1,\dots,\hat{C}_{\hat{N}}\}$. This ensures that the all-singleton partition in $\mathcal{G}'$ has the same value of modularity as the partition of $\mathcal{G}$ that we identified at the end of phase 1. One then repeats phase 1 on the reduced network and continues iterating until the heuristic converges (i.e., until phase 2 induces no further changes). We refer to a partition obtained after convergence of the heuristic as an *output partition*. In practice, one can solve the multilayer modularity-maximization problem with the Louvain heuristic by using the multilayer modularity matrix $\boldsymbol{\mathcal{B}}$ instead of the single-layer modularity matrix $\boldsymbol{B}$ as an input (the number of nodes in the first iteration of phase 1 becomes $N|\mathcal{T}|$ instead of N).

Because we use a nondeterministic implementation of the Louvain heuristic—in particular, the node order is randomized at the start of each iteration of phase 1—the network partitions that we obtain for a fixed modularity matrix can differ across runs.[12] To account for this, one can compute the frequency of co-classification of nodes into communities for a given modularity matrix $\boldsymbol{B}$ across multiple runs of the heuristic instead of using the output partition of a single run. (See [119] for an application of such an approach to "consensus clustering" and [174] for an application of such an approach to hierarchical clustering.) We use the term *association matrix* for a matrix that stores the mean number of times that two nodes are placed in the same community across multiple runs of a heuristic, and we use the term *co-classification index* of nodes i and j to designate the $(i,j)^{\text{th}}$ entry of an association matrix. We call *multiscale community structure* a set $\mathcal{C}_{\text{local}}(\boldsymbol{\gamma})$ of local optima that we obtain for a set of (not necessarily all distinct) resolution-parameter values $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_l\}$, where $\gamma^- = \gamma_1 \leq \ldots \leq \gamma_l = \gamma^+$ (with γ^+ and γ^- defined in Section 2.1.2.2) and $\boldsymbol{B} = \boldsymbol{A} - \gamma_i \boldsymbol{P}$, $i = 1, \ldots, l$. We use the term *multiscale association matrix* for an association matrix $\hat{\boldsymbol{A}} \in [0,1]^{N \times N}$ that stores the co-classification index of all pairs of nodes for partitions in this set:

$$\hat{A}_{ij} = \frac{\sum_{C \in \mathcal{C}_{\text{local}}(\boldsymbol{\gamma})} \delta(c_i, c_j)}{|\mathcal{C}_{\text{local}}(\boldsymbol{\gamma})|} \,. \tag{2.28}$$

The summation is over the set $\mathcal{C}_{\text{local}}(\boldsymbol{\gamma})$ of output partitions obtained with a given computational heuristic using different values of γ. For each partition $C \in \mathcal{C}_{\text{local}}(\boldsymbol{\gamma})$, nodes i and j either are (i.e., $\delta(c_i, c_j) = 1$) or are not (i.e., $\delta(c_i, c_j) = 0$) in the same community. It follows that $\hat{A}_{ij}$ is the mean number of partitions in $\mathcal{C}_{\text{local}}(\boldsymbol{\gamma})$

[12] The implementation [2,102] of the heuristic that we use in this thesis is a generalized version of the implementation in [29]. It is independent of the null network—so it takes the modularity matrix as an input to allow an arbitrary choice of null network—and it randomizes the node order at the start of each iteration of the heuristic's first phase to increase the search space of the heuristic. When one chooses the same null network that was assumed in [29] and uses a node order fixed to $\{1, \ldots, N\}$ at each iteration of phase 1 (the value of N can change after each iteration of the heuristic's second phase), then the implementation in [29] and the implementation in [102] return the same output.

for which nodes i and j are in the same community. The value of $|\mathcal{C}_{\text{local}}(\boldsymbol{\gamma})|$ is the number of distinct values of γ that we consider multiplied by the number of run of the Louvain algorithm performed for each value of γ. We use the matrix $\hat{\boldsymbol{A}}$ repeatedly in our computational experiments of Section 4.1. We vary the resolution parameter in the interval $[0, \gamma^+]$ instead of $[\gamma^-, \gamma^+]$ in numerical experiments with signed observed networks because globally optimal partitions can be different in the interval $[0, \gamma^-]$.[13] In Section 4.2, we point out two issues that the Louvain heuristic (independently of how it is implemented) faces with temporal multilayer networks.

2.3.3 Spectral bipartitioning heuristic

The heuristic we refer to as *spectral bipartioning* was introduced in the context of modularity maximization by Newman in 2006 [146]. Its main idea is simple: one computes the leading eigenvector of $\boldsymbol{B}$ and one uses the sign of its entries to divide the nodes into two sets. One then iterates this procedure (in some way) on each set until no further improvement is possible. The roots of this heuristic lie in traditional spectral bipartioning [82].

Consider a network $\mathcal{G}$ with an unsigned adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, $A_{ij} \geq 0$. Let $C \in \mathcal{C}$ be a bipartition of $\mathcal{G}$ and $\boldsymbol{S}$ its associated assignment matrix with i^{th} column $\boldsymbol{s}_i$, $i \in \{1, 2\}$. Since $\boldsymbol{s}_1 = \mathbf{1} - \boldsymbol{s}_2$ in the case of a bipartition, only one vector is needed and we represent the bipartition by $\boldsymbol{s} = \boldsymbol{s}_1 - \boldsymbol{s}_2 \in \{-1, 1\}^{N \times 1}$. A *cut* of $\mathcal{G}$ is a bipartition of its nodes and the *cut-size* of a bipartition is the sum of (nonnegative) edge weights with endpoints in different clusters. That is,

$$\textit{cut-size}(C) := \frac{1}{4} \sum_{i,j=1}^{N} A_{ij}(1 - s_i s_j) = \frac{1}{4} \boldsymbol{s}^T (\boldsymbol{D} - \boldsymbol{A}) \boldsymbol{s} \,, \tag{2.29}$$

where the term $1 - s_i s_j$ equals 2 when i and j are in different sets and 0 otherwise, $\boldsymbol{D}$ is a diagonal matrix with node-strengths on its diagonal (i.e., $D_{ij} = \delta(i, j) k_i$), and $\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{A} \in \mathbb{R}^{N \times N}$ is termed the *graph Laplacian* of $\boldsymbol{A}$ [42]. The matrix $\boldsymbol{L}$ is symmetric and thus has N orthonormal eigenvectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ that form a basis for $\mathbb{R}^N$. A common approach to bipartitioning a graph is to solve the *min-cut bipartioning problem* $\min_{C \in \mathcal{C}}$ *cut-size*(C) [9, 45, 177].

Suppose that one relaxes the min-cut problem from $\boldsymbol{s} \in \{-1, 1\}^{N \times 1}$ to $\boldsymbol{s} \in [-1, 1]^{N \times 1}$ (so that $\boldsymbol{s}$ may take values between -1 and 1), subject to the constraint $||\boldsymbol{s}||_2 = 1$. It is clear from the definition of cut-size that a single cluster is a trivial

[13] For $\gamma \geq \gamma^+$, one can show that all modularity contributions no longer change signs: these are negative (respectively, positive) between pairs of nodes with $P_{ij} \geq 0$ (respectively, $P_{ij} \leq 0$).

solution to the min-cut bipartitioning problem when $\boldsymbol{A}$ has positive entries.[14] By writing $\boldsymbol{s}$ as a linear combination of $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$, Hall showed in [82] that $\boldsymbol{s} = \boldsymbol{v}_2$ gives the optimal nontrivial solution to the minimization problem

$$\begin{aligned}&\min_{\boldsymbol{s}\in[-1,1]^{N\times 1}} \sum_{i,j=1}^{N} A_{ij}(1 - s_i s_j)\\ &\text{subject to} \quad ||\boldsymbol{s}||_2 = 1\,.\end{aligned}$$

Hall's result provides the optimal non-discrete solution for the min-cut bipartioning problem. An important consequence of his observation is that it suggests a heuristic for finding a discrete solution that is "closest" to $\boldsymbol{v}_2$. Given cluster size constraints (so that the solution to the minimization problem is not the trivial partition), one can sort the entries of $\boldsymbol{v}_2$ to obtain a discrete solution $\boldsymbol{s}$ that is closest to $\boldsymbol{v}_2$ as measured by the L2 norm [9]. This approach for finding the bipartion that best approximates the second smallest eigenvector (called the *Fiedler vector* [66]) was first used by Barnes [18] and subsequently extended to approaches that use multiple eigenvectors and/or produce more than two clusters at each iteration in [9, 10, 27, 71].

Now, let $\boldsymbol{B} \in \mathbb{R}^{N\times N}$ be the modularity matrix of $\boldsymbol{A}$ and note that

$$\max_{\boldsymbol{s}\in\{-1,1\}^{N\times 1}} \boldsymbol{s}^T\boldsymbol{B}\boldsymbol{s} \Leftrightarrow \max_{C\in\mathcal{C}} \sum_{i,j=1}^{N} B_{ij}\delta(c_i, c_j) \Leftrightarrow \min_{\boldsymbol{s}\in\{-1,1\}^{N\times 1}} \sum_{i,j=1}^{N} B_{ij}(1 - s_i s_j)\,, \tag{2.30}$$

when C is a bipartition. To extend the spectral bipartitioning heuristic to modularity maximization, Newman applies the same steps as above to the modularity matrix $\boldsymbol{B}$ instead of the Laplacian matrix $\boldsymbol{L}$. Provided $\boldsymbol{B}$ is symmetric, it has a set of N orthonormal eigenvectors $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N\}$. The optimal nontrivial (because the entries of $\boldsymbol{B}$ are in general signed) non-discrete solution is given by the leading eigenvector $\boldsymbol{\mu}_N$, and one can sort the entries of $\boldsymbol{\mu}_N$ according to their sign in order to obtain a discrete solution $\boldsymbol{s}$ that is closest to $\boldsymbol{\mu}_N$ as measured by the L2 norm. To obtain a partition into more than two sets, one iterates this procedure on each set of the bipartition until no further improvement is possible. Newman uses a "generalized modularity matrix" for each set at each iteration to ensure that changes in modularity are computed with respect to the original network (and not with respect to the network induced on the set) [146, 169]. There are extensions of this heuristic that use more than two eigenvectors [169, 209] as well as some investigation of its properties in cases where one uses the NG null network [143, 149]. In the present thesis, we

[14] The trivial solution corresponds to the choice $\boldsymbol{s} = \boldsymbol{v}_1$, with $\boldsymbol{v}_1 = \frac{1}{\sqrt{N}}\boldsymbol{1}$.

restrict our discussions to recursive bipartitioning using the leading eigevector of the modularity matrix. One can in theory use this heuristic to solve the multilayer modularity-maximimization problem by applying the same steps to the multilayer modularity matrix $\boldsymbol{\mathcal{B}}$ instead of the single-layer modularity matrix $\boldsymbol{B}$, provided $\boldsymbol{\mathcal{B}}$ is symmetric. We discuss some issues that can arise with this heuristic when one uses ordinal, diagonal, and uniform inter-layer coupling in Section 4.2.3.2.

2.4 Measures of partition similarity

We use two measures of similarity to compare partitions in Chapter 5 of the thesis.

2.4.1 Normalized variation of information

Let $C, C' \in \mathcal{C}$ be two N-node partitions, with $|C| = K$ and $|C'| = K'$. Consider the probability distribution $\boldsymbol{p}(C) \in [0,1]^{K\times 1}$ associated with a partition C defined by

$$p_k(C) := \frac{|C_k|}{N}\,,$$

where $C_k \in C$, and the joint probability distribution $\boldsymbol{p}(C, C') \in [0,1]^{KK'\times 1}$ associated with partitions C and C' defined by

$$p_{k,l}(C, C') := \frac{|C_k \cap C_l|}{N}\,.$$

where $C_l \in C'$. The *variation of information* between C and C' is an information theoretic similarity measure defined by [137]

$$VI(\boldsymbol{p}(C), \boldsymbol{p}(C')) := H(\boldsymbol{p}(C)) + H(\boldsymbol{p}(C')) - 2I(\boldsymbol{p}(C), \boldsymbol{p}(C'))\,, \tag{2.31}$$

where

$$H(\boldsymbol{p}(C)) = -\sum_{k=1}^{|C|} p_k(C)\log(p_k(C))\,,$$

is the entropy of $\boldsymbol{p}(C)$, and

$$I(\boldsymbol{p}(C), \boldsymbol{p}(C')) := \sum_{k=1}^{|C|}\sum_{l=1}^{|C'|} p_{k,l}(C, C')\log\left(\frac{p_{k,l}(C, C')}{p_k(C)p_l(C')}\right)\,,$$

is the *mutual information* between $\boldsymbol{p}(C)$ and $\boldsymbol{p}(C')$. For ease of writing, we write $VI(C, C')$ and $I(C, C')$ instead of $VI(\boldsymbol{p}(C), \boldsymbol{p}(C'))$ and $I(\boldsymbol{p}(C), \boldsymbol{p}(C'))$, respectively. The mutual information between two partitions is always nonnegative, symmetric, and satisfies $I(C, C') = H(C) = H(C')$ if and only if $C = C'$ [137]. The variation of information between two partitions is a metric: it is nonnegative, symmetric, takes the value 0 if and only if $C = C'$, and satisfies the triangular inequality [137]. The variation of information is bounded above by $\log N$ (for example, this bound is reached when C is a single cluster and C' is N singleton clusters) and we use the following normalization in our numerical experiments [137]

$$SnVI(C, C') := \frac{VI(C, C')}{\log N} \in [0, 1] \,, \tag{2.32}$$

where C and C' are single-layer partitions. (The value of N is fixed in a given experiment). For comparing two multilayer $N|\mathcal{T}|$-node partitions $C, C' \in \mathcal{C}$ in Chapter 5 (unless otherwise specified) we use the average $SnVI$ over all layers in our numerical experiments. That is,

$$nVI(C, C') := \frac{1}{|\mathcal{T}|} \sum_{s=1}^{|\mathcal{T}|} SnVI(C|_s, C'|_s) \,, \tag{2.33}$$

where $C|_s$ is the partition induced by the multilayer partition C on layer s.

2.4.2 Jaccard coefficient

There are also much older measures of similarity that are based on counting the number of node pairs on which two partitions agree or disagree [93, 137]. For any two partitions C and C', a pair of nodes belongs to one of the four following categories:

N_{11}: number of node pairs that are in the same cluster in C and in C'

N_{00}: number of node pairs that are in different clusters in C and in C'

N_{10}: number of node pairs that are in the same cluster in C but not in C'

N_{01}: number of node pairs that are in the same cluster in C' but not in C

with $N_{11} + N_{00} + N_{10} + N_{01} = \binom{N}{2}$.

One of the oldest pair counting similarity measures is the *Rand Index*, defined by $RI(C, C') = (N_{11} + N_{00})/\binom{N}{2}$. One drawback of the Rand Index is that it does not range over the whole interval $[0, 1]$ (i.e., $\min(RI(C, C')) > 0$ for all C, C') and it has been shown to concentrate in a small interval near 1 in practice [70, 137]. Alternative formulations that try to adjust for this have been suggested [93, 139]. Another pair counting similarity measure is the *Jaccard coefficient*, defined by [25, 96]

$$J(C, C') = \frac{N_{11}}{N_{11} + N_{10} + N_{01}} \,.$$

The Jaccard coefficient has been used to compare the output of community-detection methods [78] and we use it once in Section 5.3.1 to check whether the qualitative behaviour observed in an experiment with nVI is robust to a different choice of similarity measure.

2.5 Summary of notation

We summarize the assumptions, conventions, and notation that we need from this Chapter for the rest of the thesis. A notation list with a page number indicating where the notation is first introduced is included in Table 1.

We use bold uppercase letters for matrices, bold lowercase letters for vectors, and $\langle\,.\,\rangle$ to denote the mean value of entries in a vector, entries in a matrix, or scalars in a set. We use superscript 'T' to denote the transpose of a real-valued matrix or vector and use $\mathbf{1}_N$ (respectively, $\mathbf{0}_N$) to denote an $N \times N$ matrix in which every entry is 1 (respectively, 0). We use $|\,.\,|$ to denote the absolute value of a scalar or to denote the number of elements in a set. We denote the number of layers by $|\mathcal{T}|$, the number of nodes in each layer by N, and the i^{th} node of the s^{th} layer by i_s. The matrix $\boldsymbol{A}_s \in \mathbb{R}^{N\times N}$, $s \in \{1,\ldots,|\mathcal{T}|\}$, is the adjacency matrix of an observed network in layer s, and k_{i_s} is the strength of node i in layer s. We assume all adjacency matrices are symmetric and suppress the subscript s from any quantity when dealing with a single-layer network for ease of writing.

We denote the adjacency matrix of a null network for layer s by $\boldsymbol{P}_s$, the resolution parameter by $\gamma \geq 0$, and the inter-layer coupling parameter by $\omega \geq 0$. The matrix $\boldsymbol{B}_s \in \mathbb{R}^{N\times N}$ is the single-layer modularity matrix of layer s (e.g., $\boldsymbol{B}_s = \boldsymbol{A}_s - \mathbf{1}_N\langle \boldsymbol{A}_s\rangle$ if one uses a uniform null network) and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{N|\mathcal{T}|\times N|\mathcal{T}|}$ is a multilayer modularity matrix (that depends on the single-layer modularity matrices $\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|}$ and the coupling parameter ω).

We denote the set of all multilayer partitions by $\mathcal{C}$, an arbitrary partition in $\mathcal{C}$ by C, and an arbitrary set in C by C_k. (For a single-layer network, $|\mathcal{T}| = 1$ and $\mathcal{C}$ is the set of all N-node partitions.) The integer set assignment (or community assignment when C optimizes a quality function) of node i_s in C is c_{i_s}, with $c_{i_s} \leq N|\mathcal{T}|$. We assume throughout the thesis that a multilayer partition C does not contain partitions into sets with multiple connected components in the graph with adjacency matrix $\boldsymbol{\mathcal{B}}$. We denote by $\mathcal{N}_s = \{1_s,\ldots,N_s\}$ the set of nodes in layer s, by $C_k|_s = C_k \cap \mathcal{N}_s$ the restriction of a set $C_k \in C$ to layer s, and by $C|_s = \{C_k|_s, C_k \in C\}$ the partition induced by a multilayer partition $C \in \mathcal{C}$ on layer s.

The single-layer modularity quality function is $Q(C|\boldsymbol{B}) = \sum_{i,j=1}^{N} B_{ij}\delta(c_i, c_j)$ and the multilayer quality function is $Q(C|\boldsymbol{B_1}, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; \omega) = \sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s}, c_{j_s}) + 2\omega \sum_{s=1}^{|\mathcal{T}|-1} \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_{s+1}})$ (which we sometimes also denote $Q(C|\boldsymbol{\mathcal{B}})$). We use the normalized variation of information averaged over all layers $nVI(C, C'))$ as our main measure of similarity between two multilayer partitions.

We use the term "global maximum" to refer to a solution of the (single-layer or multilayer) modularity maximization problem and the term "local maximum" to refer to a solution that one obtains with a computational heuristic. We use the term "association matrix" for a matrix that stores the mean number of times two nodes are placed in the same community across multiple runs of a heuristic and the term "multiscale association matrix" for an association matrix in which the value of γ varies across runs. We use the words "cluster" and "community" interchangeably throughout the thesis and we use the acronyms "NG", "U", and "NGS" to refer to the Newman–Girvan null network, the uniform null network, and the generalization of the Newman–Girvan null network to signed adjacency matrices, respectively.

Chapter 3

From Financial Data to Multilayer Networks

In this chapter we describe the multilayer network representation that we adopt in all experiments with financial data.

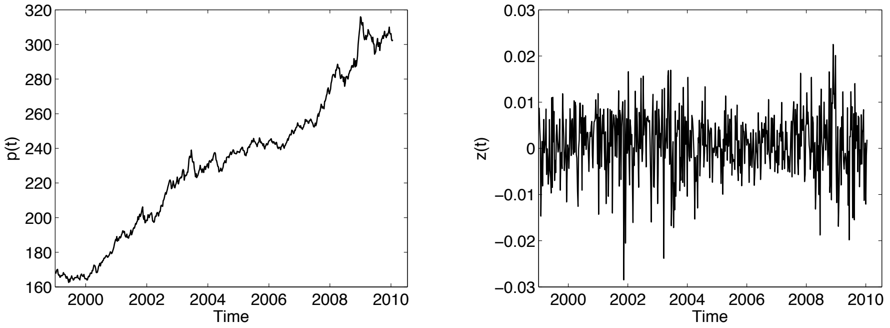
3.1 Intra-layer network representation

3.1.1 Logarithmic returns

We denote by $p_i(t)$ the price of a financial asset i at a discrete time t. We use two data sets of financial price time series $\{p_i(t_j)\}_{i=1,\ldots,N,j=1,\ldots,t_{\text{tot}}}$ throughout the thesis (one of weekly prices and one of daily prices), and we describe them in Section 3.3. In practice, rather than being concerned with the absolute price of an asset, market practitioners are usually interested in relative price change. These are called *returns* and give the potential loss or gain as a proportion of a previous price [48]. The most commonly used measure of price change is the *logarithmic return* [46, 48]

$$z_i(t_j) := \log(p_i(t_{j+1})) - \log(p_i(t_j)) = \log\left(\frac{p_i(t_{j+1})}{p_i(t_j)}\right), \quad i \in \{1, \ldots, N\},\, j \in \{1, \ldots, t_{\text{tot}}\},$$

where t_{j+1} denotes one week (respectively, day) after time t_j if prices are weekly (respectively, daily). Returns are dimensionless, i.e., they are independent of the original unit in which a price is measured. We show an example time series plot of weekly prices and logarithmic returns for a US government bond index ("USGATR" in Appendix A) in Fig. 3.1. By taking the ratio rather than the difference between consecutive prices, we measure price change as a proportion of the previous price. It is easy see why this can be desirable via a simple example. Suppose that a 20p asset and a £100 asset both increase in price by 10p. The absolute change is the same in



(a) Prices

(b) Logarithmic returns

Figure 3.1: Time series plots for a U.S. government bond index (USGATR).

both cases, but the relative change is not. Measuring price change as a proportion of the previous price reflects this difference. Furthermore, taking the logarithm of the ratio has the convenient property of being time additive. Given a time interval $[t_1, t_n]$, logarithmic returns satisfy

$$\log\left(\frac{p(t_n)}{p(t_1)}\right) = \log\left(\frac{p(t_n)}{p(t_{n-1})} \cdots \frac{p(t_2)}{p(t_1)}\right) = \sum_{k=1}^{n-1} \log\left(\frac{p(t_{k+1})}{p(t_k)}\right) ,$$

where $t_1 < t_2 \ldots < t_{n-1} < t_n$ is a discretization of the time interval $[t_1, t_n]$.[1] We show a frequency plot of logarithmic returns of assets from different asset classes in Fig. 3.2. We only consider logarithmic returns in the remainder of the thesis, and use the term "return" to mean "logarithmic return". All time intervals considered in this chapter are uniformly discretized time intervals $T = \{t_1, \ldots t_{|T|}\}$, where $t_1 < \ldots < t_{|T|}$ and $|T|$ is the number of time points in T.

3.1.2 Pearson correlation networks

We use (shifted or unshifted) sample *Pearson correlation matrices* as adjacency matrices for each layer and use a *rolling time window* to construct a time-dependent sequence of adjacency matrices. We describe both choices in the next two sections.

We first define a sample Pearson correlation matrix for a set of return time series and prove three of its basic properties: symmetry, boundedness, and positive semidefiniteness. Let T be a discrete time interval and denote by t an arbitrary time point in T. Unless otherwise specified, all quantities that we use are sample quantities (e.g.,

[1] Another type of return is the *arithmetic return* defined by $z_i^a(t) := \frac{p_i(t)-p_i(t-1)}{p_i(t-1)}$. Arithmetic returns do not have the time-additive property. However, $z_i^a(t) = \frac{p_i(t)}{p_i(t-1)} - 1$ by definition, which implies that $z_i(t) = \log(1 + z_i^a(t)) \approx z_i^a(t)$ for small $z_i^a(t)$.

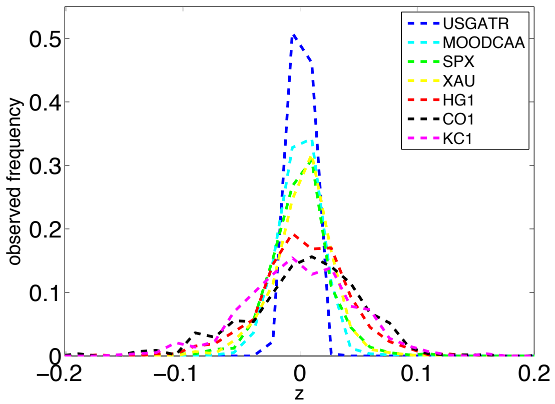


Figure 3.2: Frequency plot of logarithmic returns for financial assets from different asset classes. Asset abbreviations correspond to the following assets (or indices): US government bond (USGATR), Moody's AA rated corporate bond (MOODCAA), Standard and Poor's 500 (SPX), Gold (XAU), Copper (HG1), Crude oil (CO1), and Coffee (KC1). (Asset abbreviations are explained in Appendix A).

Pearson correlation, variance, covariance). That is, they are computed from a set of empirical observations (e.g., a set of price time series). The Pearson correlation between the return time series of assets i and j during a time interval T is defined as [111]

$$\mathrm{corr}_T(z_i, z_j) := \frac{\mathrm{cov}_T(z_i, z_j)}{\sigma_T(z_i)\sigma_T(z_j)}\,, \tag{3.1}$$

where $z_i = \{z_i(t), t \in T\}$ and

$$\begin{aligned} \langle z_i \rangle_T &:= \frac{\sum_{t\in T} z_i(t)}{|T|}\,, \\ \sigma_T(z_i) &:= \sqrt{\langle\, (z_i(t) - \langle z_i\rangle_T)^2\,\rangle_T}\,, \end{aligned}$$

are the mean and standard deviation, respectively, of z_i during the time interval T, and where

$$\mathrm{cov}_T(z_j, z_i) := \langle\, (z_i(t) - \langle z_i\rangle_T)(z_j(t) - \langle z_j\rangle_T)\,\rangle_T\,,$$

is the covariance between the returns of assets i and j during T. Using the above definitions and the linearity of the mean, we obtain the following identities:

$$\begin{aligned} \mathrm{cov}_T(z_j, z_i) &= \langle z_i z_j\rangle_T - \langle z_i\rangle_T \langle z_j\rangle_T\,, \\ \sigma_T(z_i) &= \sqrt{\mathrm{cov}_T(z_i, z_i)} = \sqrt{\langle z_i^2\rangle_T - \langle z_i\rangle_T^2}\,, \end{aligned}$$

where $\langle z_i z_j\rangle_T = \frac{1}{|T|}\sum_{t\in T} z_i(t) z_j(t)$ and $\langle z_i^2\rangle_T = \frac{1}{|T|}\sum_{t\in T} z_i^2(t)$. For ease of writing, we drop the subscript T from the above quantities in the remainder of the thesis, but we emphasize that the mean operator $\langle\,.\,\rangle$ (and consequently any quantity that depends on it) is always defined with respect to a specific time interval.

It follows immediately from the definition of $\mathrm{corr}(z_i, z_j)$ in (3.1) that

$$\mathrm{corr}(z_i, z_j) = \mathrm{corr}(z_j, z_i), \quad \text{for all } i, j \in \{1, \dots, N\}\,. \tag{3.2}$$

That is, the correlation matrix $\boldsymbol{R} = \{\mathrm{corr}(z_j, z_i)\} \in \mathbb{R}^{N\times N}$ is symmetric. It is also straightforward to see that

$$\mathrm{corr}(z_i, z_i) = 1, \quad \text{for all } i \in \{1, \dots, N\}\,. \tag{3.3}$$

Furthermore, the Cauchy-Schwarz inequality implies that

$$\left(\sum_{t\in T} \underbrace{(z_i(t) - \langle z_i\rangle)}_{u_t}\underbrace{(z_j(t) - \langle z_j\rangle)}_{v_t}\right)^2 \le \left(\sum_{t\in T}\underbrace{(z_i(t) - \langle z_i\rangle)^2}_{u_t^2}\right)\left(\sum_{t\in T}\underbrace{(z_j(t) - \langle z_j\rangle)^2}_{v_t^2}\right),$$

and so

$$\mathrm{cov}(z_i, z_j)^2 \le \sigma(z_i)^2\sigma(z_j)^2\,,$$

or equivalently,

$$-1 \le \mathrm{corr}(z_i, z_j) \le 1\,, \tag{3.4}$$

for all i and j in $\{1, \dots, N\}$.

One can express the correlation matrix $\boldsymbol{R}$ in terms of the matrix of *standardized returns* $\hat{\boldsymbol{Z}} = (\hat{z}_{ij}) \in \mathbb{R}^{N\times|T|}$, which we obtain by subtracting from each return its mean value over T and by normalizing each return by its standard deviation [65]:

$$\hat{z}_i(t) = \frac{z_i(t) - \langle z_i\rangle}{\sigma(z_i)}, \quad t \in T\,. \tag{3.5}$$

In particular,

$$\boldsymbol{R} = \frac{1}{T}\hat{\boldsymbol{Z}}\hat{\boldsymbol{Z}}^T\,, \tag{3.6}$$

because

$$\begin{aligned}\frac{1}{T}(\hat{\boldsymbol{Z}}\hat{\boldsymbol{Z}}^T)_{ij} &= \frac{1}{T}\sum_{t\in T}\hat{z}_i(t)\hat{z}_j(t) \qquad (3.7)\\ &= \frac{1}{T}\sum_{t\in T}\left(\frac{z_i(t) - \langle z_i\rangle}{\sigma(z_i)}\right)\left(\frac{z_j(t) - \langle z_j\rangle}{\sigma(z_j)}\right)\\ &= \frac{\langle\,(z_i(t) - \langle z_i\rangle)(z_j(t) - \langle z_j\rangle)\,\rangle}{\sigma(z_i)\sigma(z_j)}\\ &= \frac{\mathrm{cov}(z_i, z_j)}{\sigma(z_i)\sigma(z_j)} = \mathrm{corr}(z_i, z_j)\,.\end{aligned}$$

Using $\langle \hat{z}_i \rangle = 0$ and $\sigma(\hat{z}_i) = 1$, it follows from (3.7) that $\mathrm{corr}(z_i, z_j) = \frac{1}{T}(\hat{\boldsymbol{Z}}\hat{\boldsymbol{Z}}^T)_{ij} = \mathrm{cov}(\hat{z}_i, \hat{z}_j) = \mathrm{corr}(\hat{z}_i, \hat{z}_j)$. That is, the correlation matrix of returns is equal to the covariance matrix and correlation matrix of standardized returns. The fact that $\boldsymbol{R}$ is positive semidefinite follows directly from (3.6), because for any vector $\boldsymbol{x} \in \mathbb{R}^N$,

$$\boldsymbol{x}^T\boldsymbol{R}\boldsymbol{x} = \boldsymbol{x}^T(\hat{\boldsymbol{Z}}\hat{\boldsymbol{Z}}^T)\boldsymbol{x} = (\boldsymbol{x}^T\hat{\boldsymbol{Z}})(\hat{\boldsymbol{Z}}^T\boldsymbol{x}) = (\boldsymbol{x}^T\hat{\boldsymbol{Z}})(\boldsymbol{x}^T\hat{\boldsymbol{Z}})^T = \sum_{t\in T}\Bigg(\underbrace{\sum_{i=1}^{N} x_i\hat{\boldsymbol{z}}_i(t)}_{\hat{\boldsymbol{Z}}^T\boldsymbol{x}}\Bigg)^2 \geq 0\,,$$

where equality holds if and only if $\hat{\boldsymbol{Z}}^T\boldsymbol{x} = \boldsymbol{0}$.

3.1.3 Rolling time window

We use the term *time window* for a set of discrete time points and divide each time series into $|\mathcal{T}|$ *overlapping* time windows $\{T_1, \ldots, T_{|\mathcal{T}|}\}$. The length $|T|$ of each time window and the amount of *overlap* between consecutive time windows $|T| - \delta t$ (or the *step size* δt between consecutive time windows) are each uniform [62, 65, 153, 154].[2] Both integers $|T|$ and δt refer to a number of weeks (respectively, number of days) if the data set consists of weekly returns (respectively, daily returns). For a given data set, we study the sequence of matrices

$$\mathcal{T} = \left\{\boldsymbol{A}_s \in [-1, 1]^{N\times N} | s \in \{1, \ldots, |\mathcal{T}|\}\right\}\,,$$

where $\boldsymbol{A}_s = \boldsymbol{R}_{T_s}$ for a given choice of $|T|$ and δt. The number of time windows $|\mathcal{T}|$ is entirely determined by $|T|$, δt, and the total number of time points t_{tot} in a data set. The number of layers increases if one decreases the length of a time window (for a given amount of overlap) or if one increases the overlap between time windows (for a given window length).

We stress that the precise way that one chooses to compute a measure of similarity between pairs of time series and the subsequent choices that one makes (e.g., uniform or nonuniform window length, and overlap or no overlap if one uses a rolling time window) clearly affect the values of the similarity measure. There are myriad ways to define and to compute similarity measures—the appropriateness of a choice depends on factors such as application domain, time-series resolution, stationarity or non-stationarity of time series, and so on—and this is an active and contentious area of

[2] The amount of overlap determines the number of data points that one adds and removes from each time window. It thus determines the number of data points that can alter the connectivity patterns in each subsequent correlation matrix (i.e., in each subsequent layer).

research [51, 123, 164, 165, 178, 184, 194, 208]. Constructing a similarity matrix from a set of time series and investigating community structure in a given similarity matrix are separate problems, and we are concerned with the latter in the present thesis. Accordingly, in all of our experiments, we use Pearson correlation coefficients for our measure of similarity. We compute them using a rolling time window with a uniform window length and a uniform amount of overlap. We note that the results of Section 4.2 and Chapter 5 are independent of one's choice of adjacency matrix for each layer.

3.2 Inter-layer network representation

3.2.1 Ordinal, diagonal, and uniform inter-layer coupling

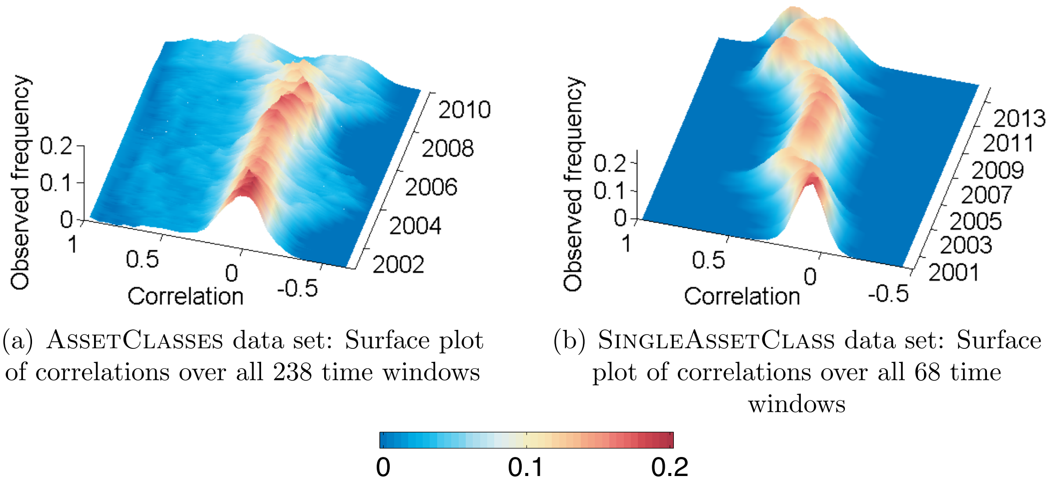
We use the multilayer representation of temporal networks described in Section 2.2.2.1 to define inter-layer connections. In particular, we assume (1) that inter-layer edges only exist between nodes that correspond to the same entity (i.e., between nodes i_s and i_r for some i and $s \neq r$), (2) that inter-layer edges are ordinal (i.e., inter-layer edges only exist between consecutive layers) and (3) inter-layer edge weights are uniform (i.e., inter-layer edges have the same weight). We denote the value of inter-layer coupling by $\omega \geq 0$. The multilayer adjacency matrix that we use in experiments with financial data is then given by

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} \boldsymbol{A}_1 & \omega \boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \omega \boldsymbol{I} & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \ddots & \ddots & \ddots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \omega \boldsymbol{I} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \omega \boldsymbol{I} & \boldsymbol{A}_{|\mathcal{T}|} \end{bmatrix}, \tag{3.8}$$

where $\boldsymbol{A}_s$ is the Pearson correlation matrix computed during the s^{th} time window. We give examples of nonuniform inter-layer coupling in Section 4.2.2 and explain how results similar to those of 4.2.2 also apply in this more general case.

3.3 Data sets

We use two data sets of financial time series in our computational experiments. We describe them in Sections 3.3.1 and 3.3.2.



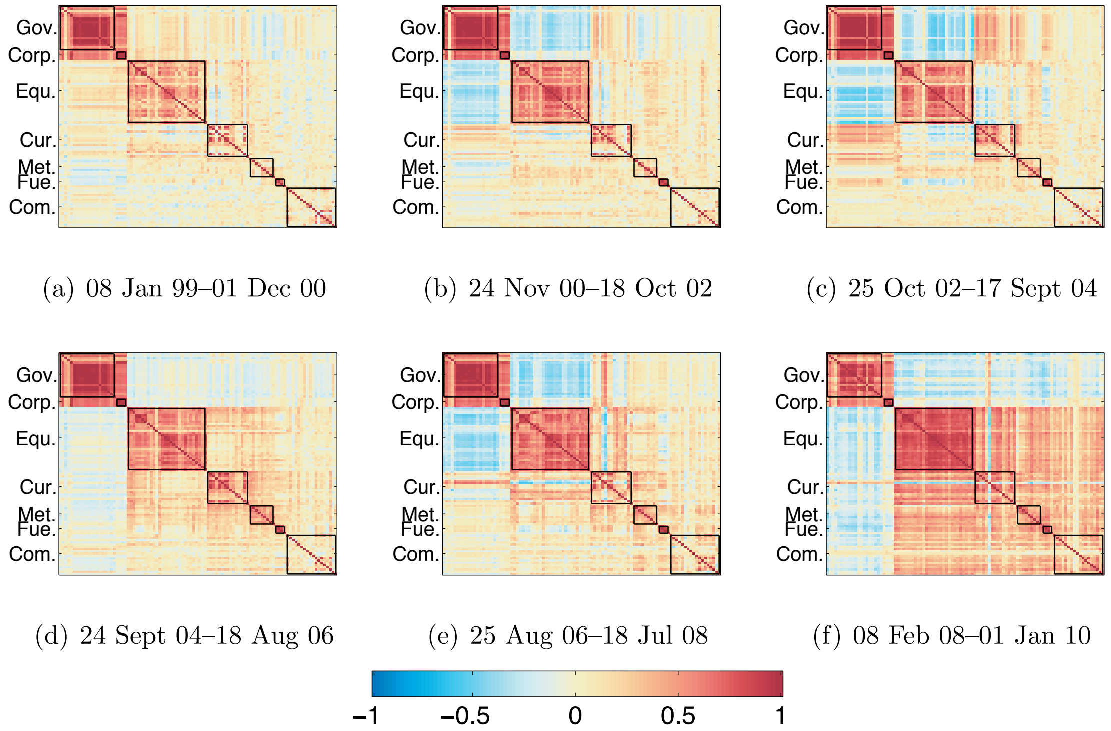
(a) AssetClasses data set: Surface plot of correlations over all 238 time windows

(b) SingleAssetClass data set: Surface plot of correlations over all 68 time windows

Figure 3.3: Surface plots of the correlations over all time windows for (a)AssetClasses data set and (b) SingleAssetClass data set. The colors in each panel scale with the value of the observed frequency.

3.3.1 AssetClasses data set

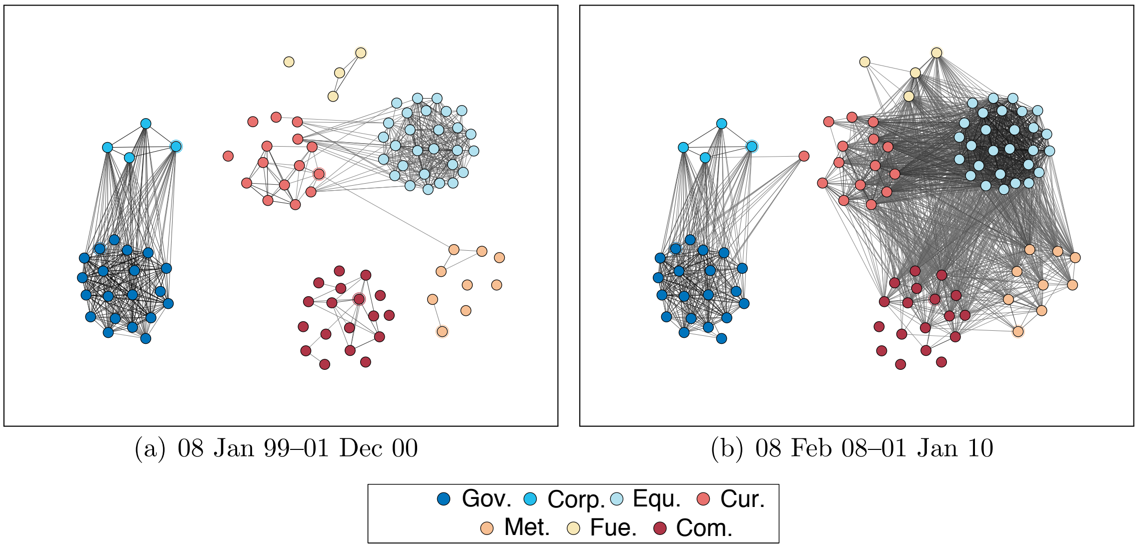
The first is a multi-asset class data set and consists of weekly price time series for $N = 98$ financial assets during the time period 01 Jan 99–01 Jan 10 (resulting in 574 prices for each asset). The assets are divided into seven asset classes: 20 government bond indices (Gov.), 4 corporate bond indices (Corp.), 28 equity indices (Equ.), 15 currencies (Cur.), 9 metals (Met.), 4 fuel commodities (Fue.), and 18 commodities (Com.). This data set covers a range of markets and geographical regions and it was studied using principal component analysis [101] in [62, 65]. The authors of [65] use weekly prices to mitigate effects resulting from the non-synchronicity of prices reported by markets from different time zones. They take the weekly price of an asset to be the last price posted each week [62]. We include a detailed description of the financial assets in this data set in Appendix A. We fix $(|T|, \delta t) = (100, 2)$ for this data set, which amounts to roughly two years of data in each time window and a difference of two data points between consecutive time windows. This produces a multilayer network with $|\mathcal{T}| = 238$ layers and $N = 98$ nodes in each layer (i.e., an $N|\mathcal{T}|$-node multilayer network with $N|\mathcal{T}| = 23324$). We include some tests of correlation robustness to time window length and time window overlap in Appendix B. We call this data set AssetClasses data set and show a surface plot of the observed frequency of correlations in each layer in Fig. 3.3(a). We show example correlation matrices and example networks of AssetClasses in Fig. 3.4 and Fig. 3.5.



(a) 08 Jan 99–01 Dec 00 (b) 24 Nov 00–18 Oct 02 (c) 25 Oct 02–17 Sept 04

(d) 24 Sept 04–18 Aug 06 (e) 25 Aug 06–18 Jul 08 (f) 08 Feb 08–01 Jan 10

Figure 3.4: Correlation matrices for AssetClasses data set computed during different time windows. Colors scale with the value of the correlations from deep blue (high negative correlation) to deep red (high positive correlation).



(a) 08 Jan 99–01 Dec 00 (b) 08 Feb 08–01 Jan 10

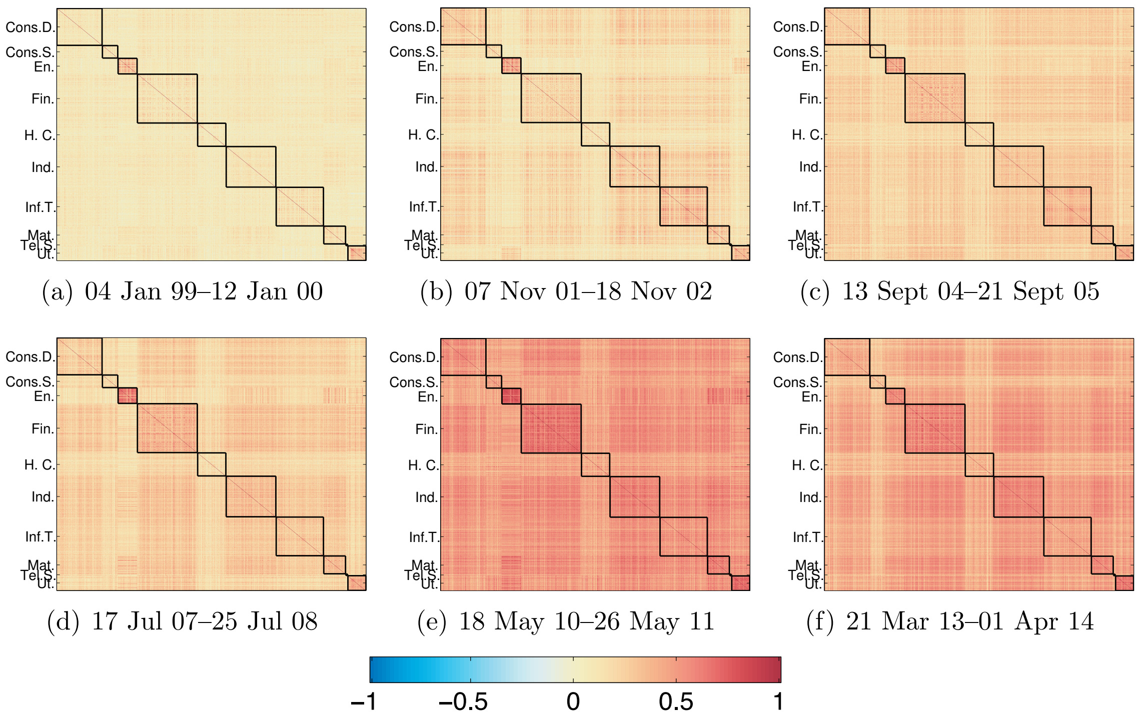
Figure 3.5: Thresholded network visualizations ($\mathrm{corr}(z_i, z_j) \geq 0.35$) for AssetClasses data set plotted using an implementation [3] of the Kamada-Kawai layout algorithm [103]. Darker edges correspond to higher edge weights.

3.3.2 SingleAssetClass data set

The second data set is a single-asset class data set and consists of daily price time series (excluding weekends) for $N = 910$ financial assets from the Standard & Poor's (S&P) 1500 during the time period 01 Jan 99–17 Jul 2014 (resulting in 3909 prices for each asset).[3] The assets in this data set are all equity assets and they are divided into ten sectors: 134 consumer discretionary (Cons. D.), 46 consumer staples (Cons. S.), 57 energy (En.), 177 financials (Fin.), 84 health care (H.C.), 147 industrials (Ind.), 140 information technology (Inf. T.), 65 materials (Mat.), 6 telecommunication services (Tel. S.), and 54 utilities (Ut.). We fix $(|T|, \delta t) = (260, 55)$ for this data set. This choice amounts to roughly one year of data in each time window and a difference of 55 data points (i.e., roughly two and a half months excluding weekends) between consecutive time windows. This produces a multilayer network with $|\mathcal{T}| = 68$ layers and $N = 910$ nodes in each layer (i.e., an $N|\mathcal{T}|$-node multilayer network with $N|\mathcal{T}| = 61880$). We include some tests of correlation robustness to time window length and time window overlap in Appendix B. We call this data set SingleAssetClass data set and show a surface plot of the observed frequency of correlations in each layer in Fig. 3.3(b). We show example correlation matrices and example networks of SingleAssetClass in Fig. 3.6 and Fig. 3.7, respectively.

We use the data sets AssetClasses and SingleAssetClass as illustrative examples in Chapter 4, where we investigate the community-detection method multilayer modularity maximization, and we perform further numerical experiments on these data sets in Chapter 6.

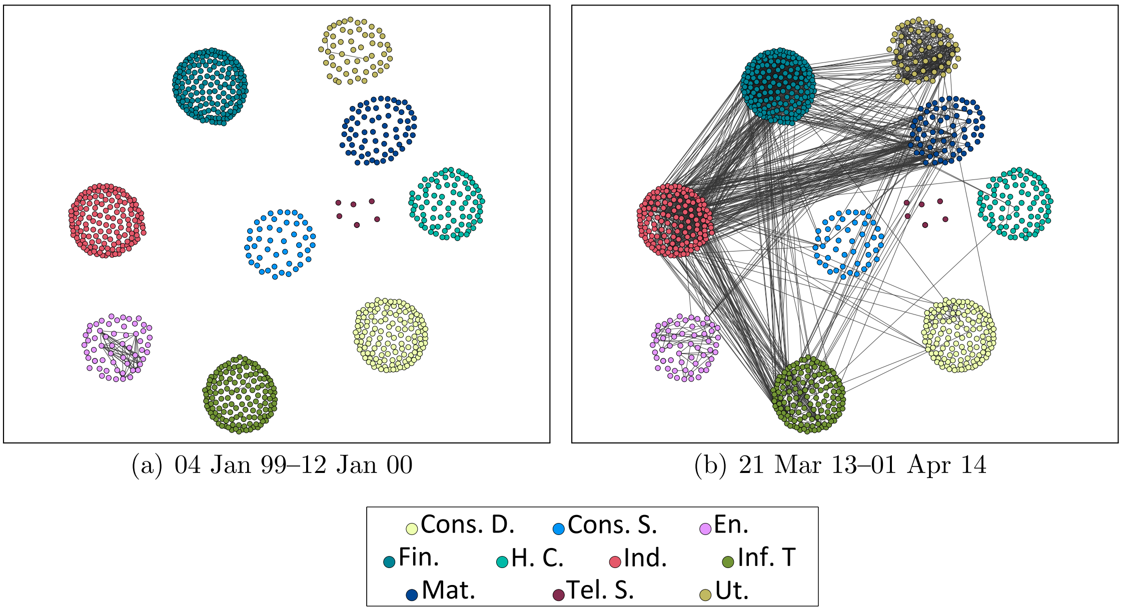
[3] We consider fewer than 1500 nodes because we only include nodes for which data is available at all time points to avoid issues associated with choices of data-cleaning techniques. We also exclude dates at which the price is missing for all assets (this amounts to 146 data points from an initial 4055 data points, giving 3909 final data points.)



(a) 04 Jan 99–12 Jan 00 (b) 07 Nov 01–18 Nov 02 (c) 13 Sept 04–21 Sept 05

(d) 17 Jul 07–25 Jul 08 (e) 18 May 10–26 May 11 (f) 21 Mar 13–01 Apr 14

Figure 3.6: Correlation matrices for SingleAssetClass data set computed during different time windows. Colors scale with the value of the correlations from deep blue (high negative correlation) to deep red (high positive correlation).



(a) 04 Jan 99–12 Jan 00 (b) 21 Mar 13–01 Apr 14

Figure 3.7: Thresholded network visualizations ($\text{corr}(z_i, z_j) \geq 0.7$) for SingleAssetClass data set plotted using an implementation [3] of the Kamada-Kawai layout algorithm [103]. Darker edges correspond to higher edge weights.

Chapter 4

Multilayer Modularity Maximization

We address two main issues in this chapter: the choice of null network and the role of inter-layer coupling in multilayer modularity maximization.

4.1 Interpretation of community structure in correlation networks with different null networks

It is clear from the structure of the multilayer modularity matrix $\boldsymbol{\mathcal{B}}$ in equation (2.23), namely

$$\boldsymbol{\mathcal{B}} = \begin{bmatrix} \boldsymbol{B}_1 & \omega\boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \omega\boldsymbol{I} & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \ddots & \ddots & \ddots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \omega\boldsymbol{I} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \omega\boldsymbol{I} & \boldsymbol{B}_{|\mathcal{T}|} \end{bmatrix},$$

that the choice of quality function within layers (i.e., diagonal blocks in the multilayer modularity matrix) and the choice of coupling between layers (i.e., off-diagonal blocks) for a given quality function affect the solution of the multilayer maximization problem in (2.24), namely

$$\max_{C \in \mathcal{C}} \sum_{i,j=1}^{N|\mathcal{T}|} \mathcal{B}_{ij}\delta(c_i, c_j)\,,$$

or equivalently,

$$\max_{C \in \mathcal{C}} \left[\sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s}, c_{j_s}) + 2\omega \sum_{s=1}^{|\mathcal{T}|-1} \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_{s+1}}) \right].$$

In this section, we make some observations on the choice of null network for correlation networks when using the modularity quality function. To do this, we consider the multilayer modularity-maximization problem with zero inter-layer coupling (i.e., $\omega = 0$), which is equivalent to performing single-layer modularity maximization on each layer independently.

4.1.1 Toy examples

We describe two simple toy networks to illustrate some features of the NG null network (2.7) (i.e., $P_{ij} = (k_i k_j)/(2m)$), and the NGS null network (2.10) (i.e., $P_{ij} = (k_i^+ k_j^+)/(2m^+) - (k_i^- k_j^-)/(2m^-)$) that can be misleading for asset correlation networks.

4.1.1.1 Newman–Girvan and uniform null networks

Assume that the nodes in a network are divided into K nonoverlapping categories (e.g., asset classes) such that all intra-category edge weights have a constant value $a > 0$ and all inter-category edge weights have a constant value b, with $0 \leq b < a$. Let κ_i denote the category of node i, and rewrite the strength of node i as[1]

$$k_i = |\kappa_i|a + (N - |\kappa_i|)b = |\kappa_i|(a-b) + Nb\,.$$

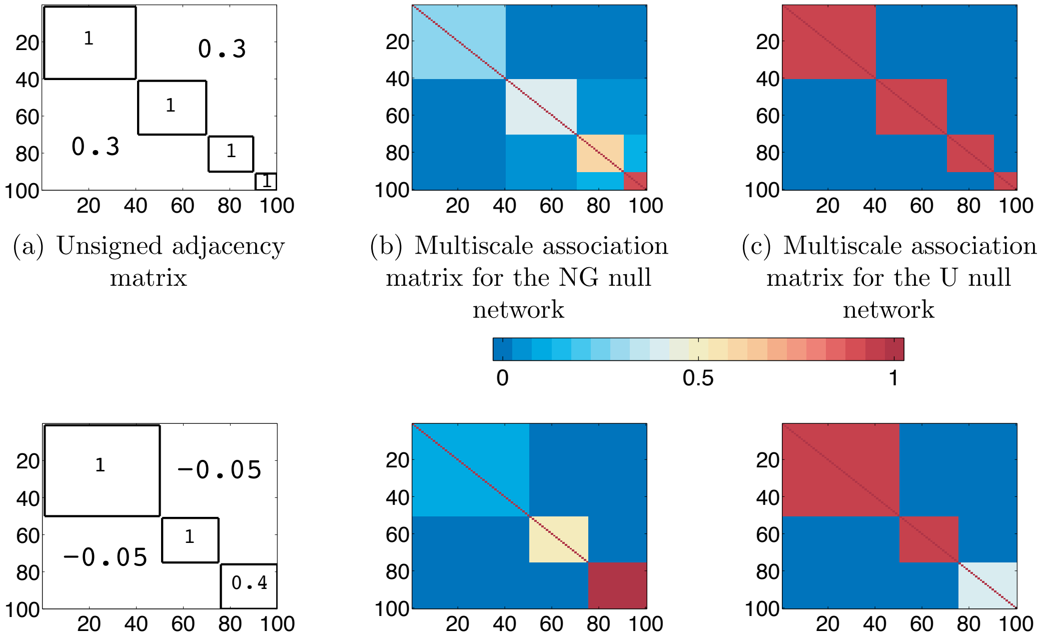
The strength of a node in this network scales linearly with the number of nodes in its category. Suppose that we have two categories κ_1, κ_2 that do not contain the same number of nodes. Taking $|\kappa_1| > |\kappa_2|$ without loss of generality, it follows that

$$P_{i,j\in\kappa_1} = \frac{1}{2m}\bigg[|\kappa_1|(a-b) + Nb\bigg]^2 > \frac{1}{2m}\bigg[|\kappa_2|(a-b) + Nb\bigg]^2 = P_{i,j\in\kappa_2}\,, \tag{4.1}$$

where $P_{i,j\in\kappa_i}$ is the expected edge weight between pairs of nodes in κ_i in the NG null network. That is, pairs of nodes in an NG null network that belong to larger categories have a larger expected edge weight than pairs of nodes that belong to smaller categories.

To see how equation (4.1) can lead to misleading results, we perform a simple experiment. Consider the toy network in Fig. 4.1(a) that contains 100 nodes divided into four categories of sizes 40, 30, 20, and 10. We set intra-category edge weights to

[1] Although we assume in this discussion that nodes have self-loops of weight a, the inequalities between expected edge weights in the next two sections (and thus their implications) hold independently of this assumption. For example, if one sets self-loops to zero in Section 4.1.1.1, one would subtract a from the strength of each node and $P_{i,j\in\kappa_1} > P_{i,j\in\kappa_2}$ in Equ. (4.1) still holds.



(a) Unsigned adjacency matrix

(b) Multiscale association matrix for the NG null network

(c) Multiscale association matrix for the U null network

(d) Signed adjacency matrix

(e) Multiscale association matrix for the NGS null network

(f) Multiscale association matrix for the U null network

Figure 4.1: (a) Toy unsigned block matrix with constant diagonal and off-diagonal blocks that take the value indicated in the block. (b) Multiscale association matrix of (a) that gives the frequency of co-classification of nodes across resolution-parameter values using an NG null network. (c) Multiscale association matrix of (a) that uses a U null network. (d) Toy signed block matrix with constant diagonal and off-diagonal blocks that take the value indicated in the block. (e) Multiscale association matrix of (d) that uses an NGS null network. (f) Multiscale association matrix of (d) that uses a U null network. For the NG and U (respectively, NGS) null networks, our sample of resolution-parameter values is the set $\{\gamma^-, \dots, \gamma^+\}$ (respectively, $\{0, \dots, \gamma^+\}$) with a discretization step of 10^{-3} between each pair of consecutive values.

1 and inter-category edge weights to 0.3 (i.e., $a = 1$ and $b = 0.3$ in equation (4.1)). In Fig. 4.1(b) (respectively, Fig. 4.1(c)), we show the multiscale association matrix defined in (2.28) using an NG null network (respectively, a U null network). Each multiscale association matrix indicates the number of times two nodes are placed in the same community across resolution-parameter values ranging from γ^- (a single community) to γ^+ (N singleton communities). Colors scale with the frequency of co-classification of pairs of nodes into the same community across resolution-parameter values. Because the nodes are ordered by category, diagonal blocks in Fig. 4.1(b,c) indicate the co-classification index of nodes in the same category, and off-diagonal blocks indicate the co-classification index of nodes in different categories. We observe in Fig. 4.1(b) that larger categories are identified as a community across a smaller

range of resolution-parameter values than smaller categories when using an NG null network. In particular, category κ is identified as a single community when $\gamma < a/P_{i,j\in\kappa}$ (with $a/P_{i,j\in\kappa_1} < a/P_{i,j\in\kappa_2}$ when $|\kappa_1| > |\kappa_2|$ by equation (4.1)). When $\gamma \geq a/P_{i,j\in\kappa}$, category κ is identified as $|\kappa|$ singleton communities. However, we observe in Fig. 4.1(c) that all four categories are identified as a single community across the same range of resolution-parameter values when using the U null network. In particular, category κ is identified as a single community when $\gamma < a/\langle \boldsymbol{A}\rangle$ and as $|\kappa|$ singleton communities when $\gamma \geq a/\langle \boldsymbol{A}\rangle$.

A common interpretation of multiscale modularity maximization is that the communities that one obtains for larger values of γ reveal "smaller" sets of "more densely" connected nodes in the observed network (e.g., hierarchical structure) [54, 114, 168]. Although all diagonal blocks in Fig. 4.1(a) have the same internal connectivity, different ones are identified as communities for different values of γ when using the NG null network—as γ increases, nodes in the largest category split into singletons first, followed by those in the second largest category, etc. One would need to be cautious in using multiscale community structure to gain information about connectivity patterns in the observed network in this example.

4.1.1.2 Newman–Girvan signed and uniform null networks

A key difference between an NG null network (2.7) and an NGS null network (2.10) is that the expected edge weight between two nodes must be positive in the former but can be negative in the latter. Consider a signed variant of the example in Section 4.1.1.1 in which intra-category edge weights equal a constant $a > 0$ and inter-category edge weights equal a constant $b < 0$. The strengths of node i in the κ^{th} category are

$$k_i^+ = |\kappa|a \quad \text{and} \quad k_i^- = (N - |\kappa|)b\,.$$

We consider two categories κ_1, κ_2 with different numbers of nodes. Taking $|\kappa_1| > |\kappa_2|$ without loss of generality, it follows that

$$\begin{aligned} P_{i,j\in\kappa_1} &= \frac{1}{2m^+}\bigg(|\kappa_1|a\bigg)^2 - \frac{1}{2m^-}\bigg[(N-|\kappa_1|)b\bigg]^2 \\ &> \frac{1}{2m^+}\bigg(|\kappa_2|a\bigg)^2 - \frac{1}{2m^-}\bigg[(N-|\kappa_2|)b\bigg]^2 = P_{i,j\in\kappa_2}\,, \end{aligned}$$

where $P_{i,j\in\kappa_i}$ is the expected edge weight between pairs of nodes in κ_i in the NGS null network. As was the case for an NG null network, pairs of nodes in an NGS null network that belong to larger categories have a larger expected edge weight than pairs of nodes that belong to smaller categories.

However, the fact that the expected edge weight can be negative can further complicate interpretations of multiscale community structure. A category κ for which $P_{i,j\in\kappa} < 0$ and $P_{i\in\kappa,j\notin\kappa} \geq 0$ is identified as a community when $-A_{ij} < -\gamma P_{ij}$ for all $i,j \in \kappa$ (this inequality must hold for sufficiently large γ because $P_{i,j\in\kappa} < 0$) and does not split further for larger values of γ. This poses a particular problem in the interpretation of multiscale community structure obtained with the NGS null network because nodes with negative expected edge weights do not need to be "densely connected" in the observed network to contribute positively to modularity. In fact, if one relaxes the assumption of uniform edge weights across categories, one can ensure that nodes in the category with *lowest* intra-category edge weight will *never* split. This is counterintuitive to standard interpretations of multiscale community structure [114].

In Fig. 4.1(d,e), we illustrate the above feature of the NGS null network using a simple example. The toy network in Fig. 4.1(d) contains 100 nodes divided into three categories: one of size 50 and two of size 25. The category of size 50 and one category of size 25 have an intra-category edge weight of 1 between each pair of nodes. The other category of size 25 has an intra-category edge weight of 0.4 between each pair of nodes. All inter-category edges have weights of -0.05. (We choose these values so that the intra-category expected edge weight is negative for the third category but positive for the first two and so that inter-category expected edge weights are positive.) We observe in Fig. 4.1(e) that the first and second categories split into singletons for sufficiently large γ, that the smaller of the two categories splits into singletons for a larger value of the resolution parameter, and that the third category never splits. Repeating the same experiment with the U null network in Fig. 4.1(f) (after a linear shift of the adjacency matrix to the interval $[0,1]$, i.e., $A_{ij} \mapsto \frac{1}{2}(A_{ij}+1)$ for all i,j), we observe that the co-classification index of nodes reflects the value of the edge weight between them. It is highest for pairs of nodes in the first and second category, and it is lowest for pairs of nodes in the third category.

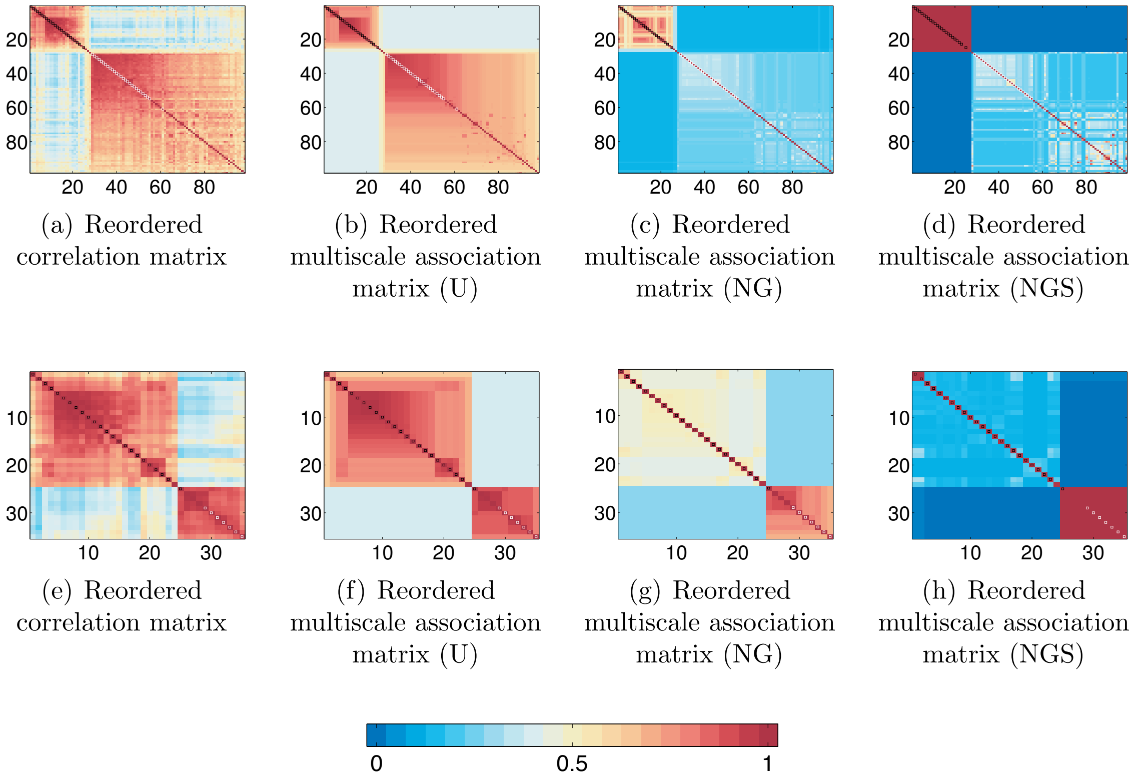
4.1.2 Multiscale community structure in asset correlation networks

We perform the same experiments as in Fig. 4.1 on the correlation matrices of the data sets AssetClasses and SingleAssetClass of Section 3.3. Our resolution-parameter sample is the set $\{\gamma^-,\ldots,\gamma^+\}$ (respectively, $\{0,\ldots,\gamma^+\}$) for the U and NG (respectively, NGS) null networks with a discretization step of the order of 10^{-3}.

We store the co-classification index of pairs of nodes averaged over all resolution-parameter values in the sample. We use the U and NG null networks for a correlation matrix that is linearly shifted to the interval $[0, 1]$. For each null network, we thereby produce $|\mathcal{T}|$ multiscale association matrices with entries between 0 and 1 that indicate how often pairs of nodes are in the same community across resolution-parameter values. The multiscale association matrix can shed light on which sets of assets are more highly correlated to each other than expected under the null model, and the extent to which they are (indicated by values closer to 1). In the case where the distribution of entries in the modularity matrix is the same as that of the correlation matrix (e.g., if one uses a uniform null network), larger values in the multiscale association matrix should correspond to larger values of observed correlation. In the case where the distribution of entries in the modularity matrix is different than that of the correlation matrix, the relationship between the multiscale association matrix and correlation matrix will depend on one's choice of null network (i.e., on how these distributions differ). We elaborate further on this point in the discussion below.

We show the multiscale association matrices for a specific layer of the AssetClasses data set in Fig. 4.2. The matrix in Fig. 4.2(a) corresponds to the correlation matrix during the interval 08 Feb 08–01 Dec 10. In accord with the results in [65], this matrix reflects the increase in correlation between financial assets that took place after the Lehman bankruptcy in 2008 compared to correlation matrices that we compute from earlier time periods (see, e.g., Fig. 3.4 and the surface plot of Fig. 3.3(a).) The matrices in Fig. 4.2(b,c,d) correspond, respectively, to the multiscale association matrix for the U, NG, and NGS null networks. We reorder all matrices (identically) using a node ordering based on the partitions that we obtain with the U null network that emphasizes block-diagonal structure in the correlation matrix. We observe that the co-classification indices in the multiscale association matrix of Fig. 4.2(b) are a better reflection of the strength of correlation between assets in Fig. 4.2(a) than the multiscale association matrices in Fig. 4.2(c,d).

As indicated by the darker shades of red in the upper left corner in Fig. 4.2(c,d), we also observe that the government and corporate bond assets (which we represent with black squares on the diagonal) are in the same community for a larger range of resolution-parameter values than the range for which equity assets are in the same community. In fact, when we use an NGS null network, the expected weight between two government or corporate bonds is negative (it is roughly -0.1), and these assets are in the same community for arbitrarily large values of the resolution parameter. (In other words, they do not split into smaller communities for large γ). On the other



(a) Reordered correlation matrix

(b) Reordered multiscale association matrix (U)

(c) Reordered multiscale association matrix (NG)

(d) Reordered multiscale association matrix (NGS)

(e) Reordered correlation matrix

(f) Reordered multiscale association matrix (U)

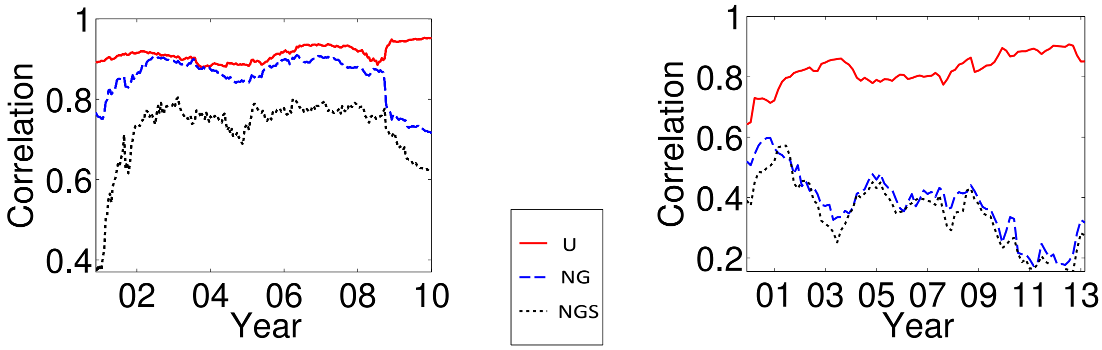
(g) Reordered multiscale association matrix (NG)

(h) Reordered multiscale association matrix (NGS)

Figure 4.2: Multiscale association matrix for the U, NG, and NGS null networks for the entire correlation matrix and a subset of the correlation in the last layer of AssetClasses data set. In panel (a), we show the entire matrix; in panels (b,c,d), we show the multiscale association matrix that we obtain from this matrix using each of the three null networks. In panel (e), we show the first 35 × 35 block of the correlation matrix from panel (a); and in panels (f,g,h), we show the multiscale association matrix that we obtain from this subset of the correlation matrix using each of the three null networks. The colors scale with the entries of the multiscale association and the entries of the correlation matrix. Black squares on the diagonals correspond to government and corporate bond assets, and white squares correspond to equity assets.

hand, the expected weight between two equities is positive, and although equities are comparably strongly correlated to each other in the observed network as bonds (see black and white squares in Fig. 4.2(a)), they split for large enough values of the resolution parameter. One would need to be cautious in using the multiscale association matrices in Fig. 4.2(c,d) to gain insight about the connectivity between assets in Fig. 4.2(a).

When studying correlation matrices of a data set with multiple asset classes, one may wish to vary the size of the asset classes included in the data (e.g., by varying the ratio of equity and bond assets). We show how doing this can lead to further misleading conclusions. By repeating the same experiment using only a subset of



(a) Correlation between multiscale association matrix and adjacency matrix (AssetClasses)

(b) Correlation between multiscale association matrix and adjacency matrix (SingleAssetClass)

Figure 4.3: Correlation between the adjacency matrix and the multiscale association matrix for the U (solid curve), NG (dashed curve), and NGS (dotted curve) null networks over all time layers for (a) AssetClasses data set and (b) SingleAssetClass data set. We compute the "total Pearson correlation" coefficients between entries in the upper diagonal blocks in each matrix (to avoid double counting, as the matrices are symmetric), and we exclude diagonal entries (which, by construction, are equal to 1 in both matrices). To compute the total Pearson correlation, we vectorize the upper diagonal block of each matrix and compute the Pearson correlation between vectors in the standard way.

the correlation matrix (the first 35 nodes), we consider an example where we have inverted the relative sizes of the bond asset class and the equity asset class. As indicated by the darker shades of red in the lower right corner in Fig. 4.2(g,h), equity assets now have a higher co-classification index than government and corporate bond assets when using the NG or NGS null networks. If one uses the co-classification index in the multiscale association matrices of Fig. 4.2(c,d) (respectively, Fig. 4.2(g,h)) to gain information about the observed correlation between equity and bond assets in Fig. 4.2(a) (respectively, Fig. 4.2(e)), one may draw different conclusions despite the fact that these have not changed. However, the multiscale association matrix with a U null network in Fig. 4.2(f) reflects the observed correlation between equity and bond assets in Fig. 4.2(e).[2]

To quantify the sense in which a multiscale association matrix of one null network

[2] The authors of [190] showed that a globally optimal partition for a null network called the "constant Potts model" (CPM), in which the edge weights are given by a constant that is *independent* of the network, is "sample-independent". Their result can be generalized as follows for the U null network (in which expected edge weights are constant but are not independent of the observed network). Suppose that $C_{\max}$ is a partition that maximizes $Q(C|\boldsymbol{A};\boldsymbol{P};\gamma_1)$ and consider the subgraph induced by the network on a set of communities $C_1,\ldots,C_l \in C_{\max}$. Then $\{C_1 \cup C_2 \ldots \cup C_l\}$ maximizes $Q(C|\hat{\boldsymbol{A}};\boldsymbol{P};\gamma_2)$, where $\hat{\boldsymbol{A}}$ is the adjacency matrix of the induced subgraph and $\gamma_2 = \gamma_1\langle\boldsymbol{A}\rangle/\langle\hat{\boldsymbol{A}}\rangle$. For the CPM null network, the same result holds with $\gamma_1 = \gamma_2$.

"reflects" the values in a correlation matrix, we compute the Pearson correlation between the upper triangular part of each multiscale association matrix and its corresponding adjacency matrix across all time layers of both data sets for the U, NG, and NGS null networks. We show these correlation plots in Fig. 4.3. Observe that the correlation between the adjacency and multiscale association matrix in Fig. 4.3(a,b) is highest in each layer for the U null network and lowest in (almost) each layer for the NGS null network.

The above observation can be explained as follows. Recall from (2.5) that we can write the modularity-maximization problem as $\max_{\boldsymbol{S}\in\mathcal{S}} \mathrm{Tr}(\boldsymbol{S^T B S})$, where $\mathcal{S}$ is the set of assignment matrices. When one uses a U null network, the entries of the modularity matrix are the entries of the adjacency matrix shifted by a constant $\gamma\langle\boldsymbol{A}\rangle$, and the modularity quality function reduces to

$$\max_{\boldsymbol{S}\in\mathcal{S}} \left[\mathrm{Tr}(\boldsymbol{S^T A S}) - \gamma\langle\boldsymbol{A}\rangle||c(\boldsymbol{S})||_2\right] , \tag{4.2}$$

where $||c(\boldsymbol{S})||_2 = ||\mathrm{Tr}(\boldsymbol{S^T 1}_N\boldsymbol{S})||_2$ is the 2-norm of the vector of set sizes in $\boldsymbol{S}$ (i.e., $c(\boldsymbol{S})$ is the vector whose k^{th} entry is $\sum_{i=1}^{N} S_{ik}$). It follows that modularity maximization with a U null network is equivalent to a block-diagonalization of the adjacency matrix $\boldsymbol{A}$ (the first term in (4.2)) with a penalty on the size of communities (the second term). In this case, one anticipates large values in the multiscale association matrix to indicate large correlation values in the observed network. As one increases the resolution parameter, one favors smaller sets of nodes with stronger internal connectivity. Note that one could also apply equation (4.2) on adjusted adjacency matrices $\boldsymbol{A}' = \boldsymbol{A} - \tilde{\boldsymbol{A}}$. For example, one can let $\tilde{\boldsymbol{A}}$ be a matrix that controls for random fluctuations in a correlation matrix $\boldsymbol{A}$ (e.g., the "random component" $\boldsymbol{C}^r$ in [131]).

For a general null network, equation (4.2) takes the form

$$\max_{\boldsymbol{S}\in\mathcal{S}} \left[\mathrm{Tr}(\boldsymbol{S^T A S}) - \mathrm{Tr}(\boldsymbol{S^T}(\gamma\boldsymbol{P})\boldsymbol{S})\right] ,$$

where $\boldsymbol{P}$ is the adjacency matrix of the null network. That is, modularity maximization finds block-diagonal structure in $\boldsymbol{A}$ (first term) that is not in $\gamma\boldsymbol{P}$ (second term). It is common to avoid using the U null network in applications because "it is not a good representation of most real-world networks" [146]. The extent to which one wants a null network to be a good representation of an observed network depends on the features which one wants to take as given. We argue that whether an NG null network is more appropriate than a U null network for a given situation depends at least in part on one's interpretation of node strength for that situation. As we discussed in Section 2.1.2.3, the strength of a node in correlation matrices is given by the covariance between its standardized time series and the mean time series. When using the NG null network, it thus follows that pairwise differences $B_{ij} - B_{i'j'}$ in the modularity quality function depend on $\text{corr}(\hat{z}_i, \hat{z}_j), \text{corr}(\hat{z}_{i'}, \hat{z}_{j'})$, and $\text{corr}(\hat{z}_k, \hat{z}_{\text{tot}})$, where $k \in \{i, j, i^{'}, j^{'}\}$, the quantity $\hat{z}_i$ is the standardized time series of asset i defined in subsection 2.1.2.3, and $\hat{z}_{\text{tot}} = \sum_{i=1}^{N} \hat{z}_i$. When using the U null network, pairwise differences in the modularity quality function depend only on the observed edge weights $\text{corr}(\hat{z}_i, \hat{z}_j)$ and $\text{corr}(\hat{z}_{i'}, \hat{z}_{j'})$. The term $\text{corr}(\hat{z}_k, \hat{z}_{\text{tot}})$ introduces a dependence between the communities that one finds with the NG null network and the extent to which nodes in those communities are representative of the mean time series for the sample [as measured by $\text{corr}(\hat{z}_k, \hat{z}_{\text{tot}})$]. In situations in which one may wish to vary one's node sample (e.g., by changing the size of asset classes), one needs to bear such dependencies in mind when interpreting the communities that one obtains.

4.2 Effect of inter-layer coupling on a multilayer partition

In Section 4.1, we set the inter-layer connection weights to zero in the multilayer network. The solution to the multilayer modularity-maximization problem (2.24) then depends solely on the values in the modularity matrix of each time layer, and the multilayer modularity-maximization problem reduces to performing single-layer modularity maximization on each layer independently.

Recall the multilayer modularity-maximization problem

$$\max_{C \in \mathcal{C}} \left[\sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs} \delta(c_{i_s}, c_{j_s}) + 2\omega \sum_{s=1}^{|\mathcal{T}|-1} \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_{s+1}}) \right] .$$

A solution to this problem is a partition of an $N|\mathcal{T}|$-node multilayer network. Its communities can contain nodes from the same layer and nodes from different layers. Nodes from different layers can be the same node at different times ((i_s, i_r) with $s \neq r$) or different nodes at different times ((i_s, j_r) with $i \neq j$ and $s \neq r$). We say that a node i_s remains in the same community (respectively, changes communities) between consecutive layers s and $s+1$ if $\delta(c_{i_s}, c_{i_{s+1}}) = 1$ (respectively, $\delta(c_{i_s}, c_{i_{s+1}}) = 0$).

Positive ordinal, diagonal, and uniform inter-layer connections favor nodes remaining in the same community between consecutive layers. Every time a node does not change communities between two consecutive layers (i.e., $\delta(c_{i_s}, c_{i_{s+1}}) = 1$), a positive contribution of 2ω is added to the multilayer quality function. One thereby favors communities that do not to change in time because community assignments are transitive: if $\delta(c_{i_s}, c_{j_s}) = 1$ and $\delta(c_{i_s}, c_{i_{s+1}}) = \delta(c_{j_s}, c_{j_{s+1}}) = 1$, then $\delta(c_{i_{s+1}}, c_{j_{s+1}}) = 1$. We define the *persistence* of a multilayer partition to be the total number of nodes that do not change communities between layers:

$$\mathrm{Pers}(C) := \sum_{s=1}^{|\mathcal{T}|-1} \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_{s+1}}) \in \{0, \ldots, N(|\mathcal{T}|-1)\}\,, \tag{4.3}$$

and we call the *intra-layer modularity* of a multilayer partition the contributions to the quality function from single-layer modularity matrices:

$$Q(C|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; 0) = \sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s}, c_{j_s}). \tag{4.4}$$

As indicated in equation (4.3), $\mathrm{Pers}(C)$ is an integer between 0, which occurs when no node ever remains in the same community across layers, and $N(|\mathcal{T}|-1)$, which occurs when every node always remains in the same community. (See [22] for a closely related measure called "flexibility" that has been applied to functional brain networks and see [136] for a related measure called "classification error metric".) Let $\mathrm{Pers}(C)|_{s,r}$ denote the number of nodes that are in the same community in layer s and layer r:

$$\mathrm{Pers}(C)|_{s,r} := \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_r}) \in \{0, \ldots, N\}\,, \tag{4.5}$$

so that $\mathrm{Pers}(C) = \sum_{s=1}^{|\mathcal{T}|-1} \mathrm{Pers}(C)|_{s,s+1}$. We use $\mathrm{Pers}(C)|_s$ as shorthand notation for $\mathrm{Pers}(C)|_{s,s+1}$. Persistence provides an insightful way of rewriting the multilayer modularity-maximization problem:

$$\max_{C \in \mathcal{C}} \left[\sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s}, c_{j_s}) + 2\omega \mathrm{Pers}(C) \right]. \tag{4.6}$$

The multilayer maximization problem thus measures a trade-off between intra-layer modularity (the first term in (4.6)) and temporal persistence across layers (the second term in (4.6)).[3]

In the next two subsections, we illustrate how the set of intra-layer partitions induced by a multilayer partition with $\omega > 0$ on individual layers can differ from intra-layer partitions obtained with $\omega = 0$. We refer to a "globally optimal partition" as an "optimal partition" in Sections 4.2.1 and 4.2.2 for ease of writing.

[3] With this reformulation of the multilayer maximization problem, one can think of $\omega \geq 0$ as a parameter that favors persistence rather than as an edge weight in the multilayer network.

4.2.1 Toy examples

4.2.1.1 Changes in connectivity patterns

This toy examples illustrates how inter-layer coupling can enable us to detect and differentiate between changes in connectivity patterns across layers. In Fig. 4.4, we show an unweighted multilayer network with $|\mathcal{T}| = 10$ layers and $N = 8$ nodes in each layer. Every layer except for layers 3 and 6 contains two 4-node cliques. In layer 3, node 5_3 is connected to nodes $\{1_3, 2_3\}$ instead of nodes $\{6_3, 7_3, 8_3\}$. In layer 6, node 5_6 is connected to nodes $\{1_6, 2_6, 3_6, 4_6\}$ instead of nodes $\{6_6, 7_6, 8_6\}$. We show the layers of the multilayer network in panels (a)–(c) of Fig. 4.4. We examine its communities using a U null network with a resolution-parameter $\gamma = 1$. Layer s then has the following single-layer modularity matrix:

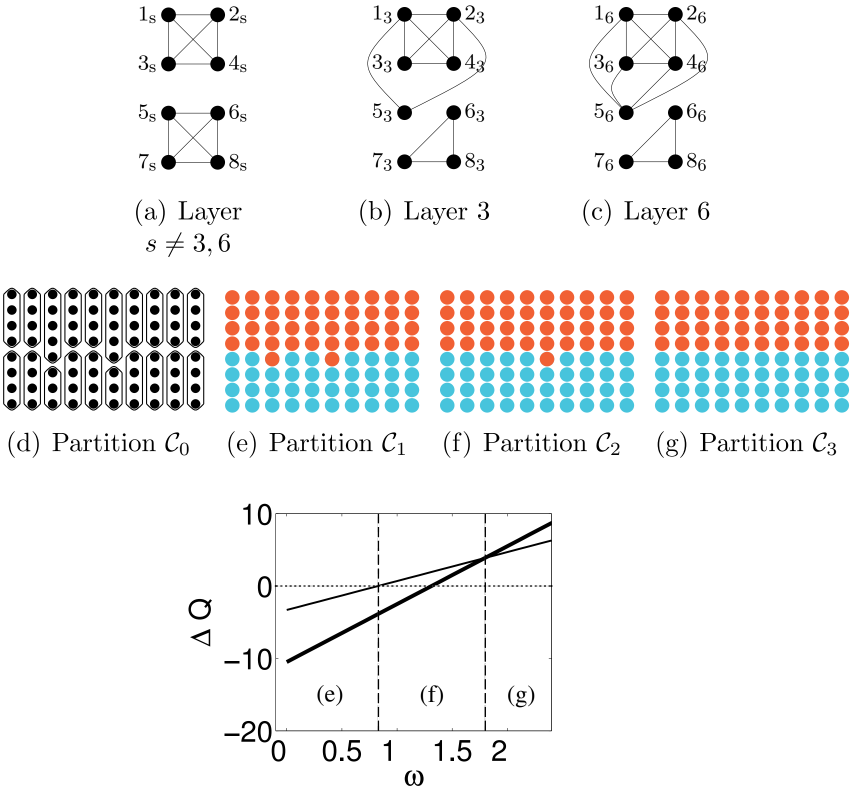
$$B_{ijs} = \begin{cases} 1 - \langle \boldsymbol{A}_s \rangle \,, & \text{if } i \text{ is connected to } j \\ -\langle \boldsymbol{A}_s \rangle \,, & \text{otherwise}\,. \end{cases}$$

The optimal partition in each layer is unique and is $C_s = \{\{1_s, 2_s, 3_s, 4_s\}, \{5_s, 6_s, 7_s, 8_s\}\}$ in layer s for $s \notin \{3, 6\}$ and is $C_s = \{\{1_s, 2_s, 3_s, 4_s, 5_s\}, \{6_s, 7_s, 8_s\}\}$ in layers 3 and 6. When the value of inter-layer coupling is 0, the optimal multilayer partition is the union of $|\mathcal{T}|$ disconnected optimal single-layer partitions. The resulting multilayer partition $\mathcal{C}_0 = \bigcup_{i=1}^{10} C_s$, which we show in panel (d) of Fig. 4.4, has a persistence of $\text{Pers}(\mathcal{C}_0) = 0$. For any $\omega > 0$, any partition with the same intra-layer partitions as $\mathcal{C}_0$ and a nonzero value of persistence yields a higher value of multilayer modularity than $\mathcal{C}_0$. This follows immediately from the expression of the multilayer quality function:

$$Q(C|\boldsymbol{\mathcal{B}}) = \sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs} \delta(c_{i_s}, c_{j_s}) + 2\omega \text{Pers}(C)\,.$$

Increasing persistence without changing intra-layer partitions increases the last term of $Q(C|\boldsymbol{\mathcal{B}})$ without changing the other terms. (In Section 4.2.2, we prove that $\omega > 0$ is both necessary and sufficient for an optimal partition to have a positive value of persistence.) To obtain the multilayer partition in panel (e), we combine all of the sets in panel (d) that contain 1_s into one set and all of the sets that contain N_s into another set. This partition has a persistence equal to $N(|\mathcal{T}| - 1) - 4$, and any other way of combining the sets in $\mathcal{C}_0$ yields a lower value of persistence.

We now examine Fig. 4.4 further. We consider the multilayer partitions in panels (e)–(g). The example in panel (e) illustrates the structural changes from both layer 3 [see panel (b)] and layer 6 [see panel (c)], the example in panel (f) illustrates only the change from layer 6, and panel (g) does not illustrate either change. As we soon



(a) Layer $s \neq 3, 6$

(b) Layer 3

(c) Layer 6

(d) Partition $\mathcal{C}_0$

(e) Partition $\mathcal{C}_1$

(f) Partition $\mathcal{C}_2$

(g) Partition $\mathcal{C}_3$

(h) Change in multilayer modularity value with respect to partition $\mathcal{C}_1$

Figure 4.4: Toy example illustrating the use of ordinal diagonal and uniform inter-layer coupling for detecting changes in community structure across layers. We consider ten layers ($|\mathcal{T}| = 10$) with eight nodes ($N = 8$) in each layer. We show the network structures in (a) layers $s \notin \{3, 6\}$, (b) layer 3, and (c) layer 6. Panels (d)–(g) illustrate four different multilayer partitions. In each panel, the s^{th} column of circles represents the nodes in the s^{th} layer, which we order from 1 to 8. We show sets of nodes in the same community using solid curves in panel (d) (to avoid having to use 20 distinct colors) and using colors in panels (e)–(g). In panel (h), we show the difference between the multilayer modularity value between the partition in panels (f) (thin line) and (g) (thick line) and the partition in panel (e) for different values of ω. We include the horizontal dotted line to show the point at which the thin line intercepts the horizontal axis. The panel labels in the regions defined by the area between two consecutive vertical lines in panel (h) indicate which of the multilayer partitions in panels (e), (f), and (g) has a higher value of modularity.

quantify in terms of modularity cost, the change in layer 6 is the "stronger" of the two changes. We let $\mathcal{C}_1$ denote the multilayer partition in panel (e), $\mathcal{C}_2$ denote the multilayer partition in panel (f), and $\mathcal{C}_3$ denote the multilayer partition in panel (g). Note that $\text{Pers}(\mathcal{C}_1) < \text{Pers}(\mathcal{C}_2) < \text{Pers}(\mathcal{C}_3)$. The value ω of inter-layer coupling determines which partition of these three has the highest value of multilayer modularity. To see this, we compute the modularity cost of changing static community structure within layers in partition $\mathcal{C}_1$ in favor of persistence. (Such a computation is a multilayer version of the calculations for static networks in [77].) The intra-layer modularity cost in $\mathcal{C}_1$ of moving node 5_s from the community $\{1_s, 2_s, 3_s, 4_s, 5_s\}$ to the community $\{6_s, 7_s, 8_s\}$ in layers $s \in \{3, 6\}$ is

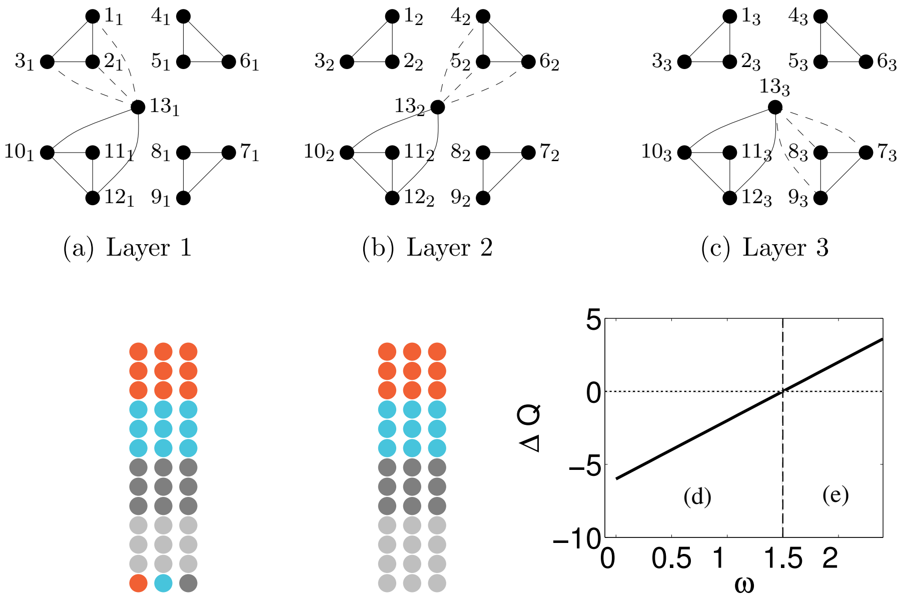
$$\begin{aligned}\Delta Q(s) &= 2\left(\sum_{j\in\{6,7,8\}} B_{5js} - \sum_{j\in\{1,2,3,4\}} B_{5js}\right)\\ &= \begin{cases} -4 + 2\langle \boldsymbol{A}\rangle_3 \approx -3.3\,, & \text{if } s = 3\\ -8 + 2\langle \boldsymbol{A}\rangle_6 \approx -7.2\,, & \text{if } s = 6\,.\end{cases}\end{aligned}$$

The inter-layer modularity cost from this move is $+4\omega$ in both cases; the first $+2\omega$ contribution of the $+4\omega$ follows by symmetry of $\boldsymbol{\mathcal{B}}$, and the second $+2\omega$ contribution of the $+4\omega$ follows from the fact that either move increases persistence by $+2$. Consequently, for $0 < 4\omega < |\Delta Q(3)|$, the partition in panel (e) yields a higher multilayer modularity value than the partitions in (f) and (g). When $|\Delta Q(3)| < 4\omega < |\Delta Q(6)|$, the multilayer modularity value of the partition in (f) is higher than those of (e) or (g). Finally, when $4\omega > |Q(6)|$, the partition in panel (g) has the highest multilayer modularity value. When $4\omega = |\Delta Q(3)|$ (respectively, $4\omega = |\Delta Q(6)|$), the multilayer partition in panels (e) and (f) (respectively, (f) and (g)) have the same value of multilayer modularity. We illustrate these results in Fig. 4.4(h) by plotting $Q(\mathcal{C}_2|\boldsymbol{\mathcal{B}}) - Q(\mathcal{C}_1|\boldsymbol{\mathcal{B}})$ and $Q(\mathcal{C}_3|\boldsymbol{\mathcal{B}}) - Q(\mathcal{C}_1|\boldsymbol{\mathcal{B}})$ against ω. This example is a simple illustration of how inter-layer connections can help distinguish between changes in connectivity patterns: stronger changes (in terms of modularity cost) persist across larger values of inter-layer coupling (see [19, 159] for other approaches to "change point detection" in temporal networks).

4.2.1.2 Shared connectivity patterns

In the previous toy example, the intra-layer partitions induced on each layer by the multilayer partitions in Fig. 4.4(e,f,g) are optimal for at least one layer when $\omega = 0$ (see Fig. 4.4(d)). This second example illustrates how inter-layer coupling can identify intra-layer partitions that are not optimal for any individual layer when $\omega = 0$ but which reflect connectivity patterns that are shared across layers.

In Fig. 4.5, we consider an unweighted multilayer network with $|\mathcal{T}| = 3$ layers and $N = 13$ nodes in each layer. Every s^{th} layer contains four 3-node cliques and a node that is connected to each of the three nodes in the s^{th} clique, and to nodes 10_s and 12_s in the 4^{th} clique. We show the layers of the multilayer network in panels



(a) Layer 1 (b) Layer 2 (c) Layer 3

(d) Partition $\mathcal{C}_1$ (e) Partition $\mathcal{C}_2$ (f) Change in multilayer modularity value with respect to partition $\mathcal{C}_1$

Figure 4.5: Toy example illustrating the use of ordinal diagonal and uniform inter-layer coupling for detecting shared connectivity patterns across layers. We consider three layers ($|\mathcal{T}| = 3$) with thirteen nodes ($N = 13$) in each layer. We show the network structures in (a) layer 1, (b) layer 2, and (c) layer 3. Solid lines represent edges present in all three layers and dashed lines represent edges that are only present in one of the layers. Panels (d) and (e) illustrate two different multilayer partitions. In each panel, the s^{th} column of circles represents the nodes in the s^{th} layer, which we order 1 to 13. We show sets of nodes in the same community using colors in panels (d) and (e). In panel (f), we show the difference between the multilayer modularity value between the partition in panel (e) and the partition in panel (d) for different values of ω. We include the horizontal dotted line to show the point at which the line intercepts the horizontal axis. The panel labels in the regions defined by the area between two consecutive vertical lines in panel (f) indicate which of the multilayer partitions in panels (d) and (e) has a higher value of multilayer modularity.

(a)–(c). We examine its communities using a U null network with a resolution-parameter value of $\gamma = 1$. The optimal partition in each layer is unique and is $\{\{1_1, 2_1, 3_1, 13_1\}, \{4_1, 5_1, 6_1\}, \{7_1, 8_1, 9_1\}, \{10_1, 11_1, 12_1\}\}$ for layer 1, $\{\{1_2, 2_2, 3_2\}$, $\{4_2, 5_2, 6_2, 13_2\}, \{7_2, 8_2, 9_2\}, \{10_2, 11_2, 12_2\}\}$ for layer 2, and $\{\{1_3, 2_3, 3_3\}, \{4_3, 5_3, 6_3\}$, $\{7_3, 8_3, 9_3, 13_3\}, \{10_3, 11_3, 12_3\}\}$ for layer 3. We obtain the multilayer partition $\mathcal{C}_1$ in panel (d) by combining these sets such that induced intra-layer partitions are optimal for each layer when $\omega = 0$ and persistence is maximized between layers. The multilayer partition $\mathcal{C}_2$ in panel (e) reflects connectivity patterns that are shared by all layers (i.e., node 13_s is with the fourth 3-node clique instead of the s^{th} 3-node clique); but its intra-layer partitions are not optimal for any layer when $\omega = 0$. By carrying out similar calculations to those in the previous toy example, one can show that when $\omega > 3/2$,[4] the multilayer partition in panel (e) yields a higher modularity value than the multilayer partition in panel (d). We illustrate this result in Fig. 4.5(f) by plotting $Q(\mathcal{C}_2|\boldsymbol{\mathcal{B}}) - Q(\mathcal{C}_1|\boldsymbol{\mathcal{B}})$ against ω. This example is a simple illustration of how inter-layer connections can help identify connectivity patterns that are shared across layers.

4.2.2 Some properties of multilayer partitions

We now ask how introducing positive ordinal diagonal and uniform coupling (i.e., $\omega > 0$) alters the set of maximum-modularity partitions of static networks (i.e., the case $\omega = 0$). We assume throughout this section that $|\mathcal{T}| \geq 2$. To clearly differentiate between intra-layer and inter-layer modularity contributions, we denote the quality function by

$$Q(C|\boldsymbol{B_1}, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; \omega) = \sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s}, c_{j_s}) + 2\omega \text{Pers}(C)\,,$$

instead of $Q(C|\boldsymbol{\mathcal{B}})$ throughout the section. In the discussion to follow, it will be helpful to recall our assumption from Section 2.2 that each of the partitions in the set $\mathcal{C}$ of $N|\mathcal{T}|$-node partitions contains sets that do not have multiple connected components in the graph with adjacency matrix $\boldsymbol{\mathcal{B}}$.

Let $\mathcal{C}_{\max}(\omega)$ denote the set of optimal partitions for the multilayer modularity-maximization problem (4.6), and let $C^{\omega}_{\max}$ be an arbitrary partition in $\mathcal{C}_{\max}(\omega)$. We prove several propositions that hold for an arbitrary choice of the matrices $\boldsymbol{B}_s$ (for example, if one uses the modularity quality function with a U null network and a resolution parameter value of 1, then $\boldsymbol{B}_s = \boldsymbol{A}_s - \langle \boldsymbol{A}_s \rangle \boldsymbol{1}_N$). Recall that we denote the set of nodes in layer s by $\mathcal{N}_s = \{1_s, \ldots, N_s\}$. The restriction of a set of nodes $C_k \subseteq \{1_1, \ldots, N_1; 1_2, \ldots, N_2; \ldots; 1_{|\mathcal{T}|}, \ldots, N_{|\mathcal{T}|}\}$ to a layer s is $C_k|_s = C_k \cap \mathcal{N}_s$, and the partition induced by a multilayer partition $C \in \mathcal{C}$ on layer s is

$$C|_s = \{C_k|_s, C_k \in C\}\,.$$

We use this notation repeatedly throughout this section.

[4] i.e., when $4\omega + 6[2(1 - \langle \boldsymbol{A} \rangle_s) - \langle \boldsymbol{A} \rangle_s - 3(1 - \langle \boldsymbol{A} \rangle_s)] > 0$, with $\langle \boldsymbol{A} \rangle_1 = \langle \boldsymbol{A} \rangle_2 = \langle \boldsymbol{A} \rangle_3$ by construction in this example.

Proposition 4.2.1. $\text{Pers}(C^{\omega}_{\max}) > 0 \Leftrightarrow \omega > 0\,.$

Proposition 4.2.1 ensures that as soon as (and only when) the value of ω is strictly positive, the value of persistence of an optimal solution is also positive. To prove this, it suffices to observe that if one rearranges sets in a multilayer partition by combining some of the sets into the same set *without* changing the partitions induced on individual layers, then one only changes the value of persistence in the expression of multilayer modularity. For example, this phenomenon occurs in Fig. 4.4 when going from the partition in panel (d) to the partition in panel (e).

Proof. $\Rightarrow$: We prove the contrapositive. Assume that $\omega = 0$ and consider a multilayer partition C such that $\text{Pers}(C) > 0$. The partition C contains at least one set with multiple connected components (because $\text{Pers}(C) > 0$ and nodes in different layers are not connected), and C is not optimal by our assumption that global optima do not contain sets with multiple connected components.

$\Leftarrow$: Assume that $\omega > 0$ and consider a multilayer partition C such that $\text{Pers}(C) = 0$. We will show that C is not optimal. Let i_r be an arbitrary node in $\{1_1, \dots, N_1; \dots; 1_{|\mathcal{T}|}, \dots, N_{|\mathcal{T}|}\}$ and let C_{i_r} denote the set in C that contains i_r. Let $C^{'}$ be the partition obtained from C by combining all sets that contain i_s, for some s, into one set:

$$C^{'} = \left(C \setminus \bigcup_{s=1}^{|\mathcal{T}|}\{C_{i_s}\}\right) \cup \left\{\bigcup_{s=1}^{|\mathcal{T}|} C_{i_s}\right\},$$

where C_{i_s} denotes the set in C that contains i_s. Consequently,

$$Q(C^{'}|\boldsymbol{B}_1, \dots, \boldsymbol{B}_{|\mathcal{T}|}; \omega) \geq Q(C|\boldsymbol{B}_1, \dots, \boldsymbol{B}_{|\mathcal{T}|}; \omega) + 2\omega(|\mathcal{T}| - 1)\,,$$

so C is not optimal. (Note that $2\omega(|\mathcal{T}| - 1)$ is strictly positive for $\omega > 0$ since we have assumed that $|\mathcal{T}| \geq 2$.) □

Proposition 4.2.2. *Let $C^{\omega}_{\max} \in \mathcal{C}_{\max}(\omega)$ and $C_k \in C^{\omega}_{\max}$. The set of layers*

$$\{s \in \{1, \dots, |\mathcal{T}|\} : C_k|_s \neq \varnothing\}$$

is either the empty set or a set of contiguous layers.

Proposition 4.2.2 ensures that if a community becomes empty in a given layer, then it remains empty in all subsequent layers. This result follows directly from the sparsity pattern of $\boldsymbol{\mathcal{B}}$ and our assumption that optimal solutions do not contain communities with multiple connected components. (More generally, this proposition applies to all partitions in $\mathcal{C}$ under our assumption.)

Proposition 4.2.3. $C^{\omega}_{\max}|_s = C^{\omega}_{\max}|_{s+1} \Leftrightarrow \mathrm{Pers}(C^{\omega}_{\max})|_s = N$.

Proposition 4.2.3 connects the notion of persistence between a pair of layers to the notion of change in community structure within layers. Various numerical experiments that have been performed with ordinal diagonal and uniform inter-layer coupling consist of varying the value of ω and using information about when nodes change communities between layers as an indication of change in community structure within these layers [21, 22, 140]. The equivalence relation in Proposition 4.2.3 motivates the use of $\mathrm{Pers}(C)|_s$ (or a variant thereof) as an indication of intra-layer change in community structure.

Proof. $\Leftarrow$: This follows straightforwardly by transitivity of community assignments: if $\delta(c_{j_s}, c_{j_{s+1}}) = \delta(c_{i_s}, c_{i_{s+1}}) = 1$ for all i, j, then $\delta(c_{i_s}, c_{j_s}) = 1$ if and only if $\delta(c_{i_{s+1}}, c_{j_{s+1}}) = 1$ for all i, j. (This direction holds for *any* multilayer partition; it need not be optimal.)

$\Rightarrow$: Let $C \in \mathcal{C}$ by a multilayer partition such that $C|_s = C|_{s+1}$ and $\mathrm{Pers}(C)|_s < N$ for some $s \in \{1, \ldots, |\mathcal{T}|\}$. We show that C is not optimal.[5] Consider a set $C_k \in C$ such that $C_k|_s \neq \varnothing$. If $\delta(c_{i_s}, c_{i_{s+1}}) = 1$ (respectively, $\delta(c_{i_s}, c_{i_{s+1}}) = 0$) for some $i_s \in C_k|_s$, then $\delta(c_{j_s}, c_{j_{s+1}}) = 1$ (respectively, $\delta(c_{j_s}, c_{j_{s+1}}) = 0$) for all $j_s \in C_k|_s$ by transitivity of community assignments and because $C|_s = C|_{s+1}$ by hypothesis. Because $\mathrm{Pers}(C)|_s < N$ by hypothesis, there exists at least one set $C_l|_s$ (where $C_l \in C$) of nodes such that $\delta(c_{i_s}, c_{i_{s+1}}) = 0$ for all $i_s \in C_l|_s$. Let $C_m|_{s+1}$, with $C_m \in C$, denote the set of nodes in layer $s + 1$ that contains i_{s+1} for all $i_s \in C_l|_s$. Consider the set $\cup_{r \leq s} C_l|_r$ of nodes in C_l that are in layers $\{1, \ldots, s\}$ and the set $\cup_{r>s} C_m|_r$ of nodes in C_m that are in layers $\{s + 1, \ldots, |\mathcal{T}|\}$. Because $\delta(c_{i_s}, c_{i_{s+1}}) = 0$ for all $i_s \in C_l|_s$, it follows by Proposition 4.2.2 that $C_l = \cup_{r \leq s} C_l|_r$ and $C_m = \cup_{r>s} C_m|_r$. Define the partition $C^{'}$ by

$$C' = \Big(C \setminus \left(\{C_l\} \cup \{C_m\}\right)\Big) \bigcup \Big(\{C_l \cup C_m\}\Big) .$$

[5] Imposing $\mathrm{Pers}(C)|_s = N$ by setting $c_{i_{s+1}} = c_{i_s}$ is not sufficient because changing $\mathrm{Pers}(C)|_s$ locally can change $\mathrm{Pers}(C)|_{s+1}$ or $\mathrm{Pers}(C)|_{s-1}$.

This partition satisfies $C'|_r = C|_r$ for all $r \in \{1, \dots, |\mathcal{T}|\}$, $\text{Pers}(C')|_r = \text{Pers}(C)|_r$ for all $r \neq s$, and $\text{Pers}(C')|_s > \text{Pers}(C)|_s$. It follows that $Q(C'|\boldsymbol{B}_1, \dots, \boldsymbol{B}_{|\mathcal{T}|}; \omega) > Q(C|\boldsymbol{B}_1, \dots, \boldsymbol{B}_{|\mathcal{T}|}; \omega)$ and C is not optimal. □

Propositions 4.2.1, 4.2.2, and 4.2.3 apply to an optimal partition obtained with any positive value of ω. The next two propositions concern the existence of "boundary" values for ω.

Proposition 4.2.4. *There exists $\omega_0 > 0$ such that*

$$\textit{if } \omega < \omega_0\,, \;\; \textit{then} \bigcup_{s=1}^{|\mathcal{T}|} C^{\omega}_{\max}|_s \in \mathcal{C}_{\max}(0)\,.$$

Proposition 4.2.4 reinforces the idea of thinking of ω as the cost of breaking static community structure within layers in favor of larger values of persistence across layers. It demonstrates that there is a positive value of inter-layer coupling such that for any smaller value of coupling, multilayer modularity maximization only gives more information than single-layer modularity maximization in that it identifies the set of partitions in $\mathcal{C}_{\max}(0)$ with highest persistence. The proof of this property relies on the fact that the set of possible modularity values for a given modularity matrix is finite.

Proof. Let C be an arbitrary partition such that $\cup_{s=1}^{|\mathcal{T}|} C|_s \notin \mathcal{C}_{\max}(0)$. We will show that there exists a value ω_0 of the inter-layer coupling parameter ω such that C is never optimal for any inter-layer coupling less than ω_0. Given a sequence of single-layer modularity matrices $\{\boldsymbol{B}_1, \dots, \boldsymbol{B}_{|\mathcal{T}|}\}$, the set of possible multilayer modularity values for a fixed value of $\omega > 0$ is finite and is given by

$$\mathcal{Q}_\omega = \left\{Q(C|\boldsymbol{B}_1, \dots, \boldsymbol{B}_{|\mathcal{T}|}; \omega), C \in \mathcal{C}\right\}\,,$$

where C is a multilayer partition. Let $Q_0^1 = \max \mathcal{Q}_0, Q_0^2 = \max \mathcal{Q}_0 \setminus \{Q_0^1\}$, and $\Delta Q = Q_0^1 - Q_0^2 > 0$. By hypothesis,

$$Q(C|\boldsymbol{B}_1, \dots, \boldsymbol{B}_{|\mathcal{T}|}; 0) < Q(C^0_{\max}|\boldsymbol{B}_1, \dots, \boldsymbol{B}_1; 0)\,,$$

where $C^0_{\max} \in \mathcal{C}_{\max}(0)$. Furthermore, by definition of persistence, it follows that

$$Q(C|\boldsymbol{B}_1, \dots, \boldsymbol{B}_{|\mathcal{T}|}; \omega) \leq Q_0^2 + 2\omega N(|\mathcal{T}| - 1) \qquad (4.7)$$

for all values of ω. By choosing $\omega < \omega_0$, with $\omega_0 = \Delta Q/\big[2N(|\mathcal{T}|-1)\big]$, we obtain

$$Q(C|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};\omega) \leq Q_0^2 + 2\omega N(|\mathcal{T}|-1) < Q_0^2 + \Delta Q = Q_1^0\,,$$

so C is not optimal for any inter-layer coupling less than ω_0. □

Clearly, $\omega_0 = \Delta Q/\big[2N(|\mathcal{T}|-1)\big]$ is not an upper bound for the set $\big\{\omega \in \mathbb{R}^+ : \bigcup_{s=1}^{|\mathcal{T}|} C^{\omega}_{\max}|_s \in \mathcal{C}_{\max}(0)\big\}$,[6] but our main concern is that the smallest upper bound of this set is *not* 0 (in fact, we have shown that it must be no less than $\Delta Q/\,[2N(|\mathcal{T}|-1)] > 0$.)

Proposition 4.2.5. *There exists $\omega_\infty > 0$ such that*

$$\textit{if } \omega > \omega_\infty\,,\ \textit{then } \mathrm{Pers}(C^{\omega}_{\max})|_s = N \textit{ for all } s \in \{1,\ldots,|\mathcal{T}|\}\,.$$

Proposition 4.2.5 implies that a sufficiently large value of inter-layer coupling ω guarantees that $C^{\omega}_{\max}|_s$ remains the same across layers (by Proposition 4.2.3). The proof of this proposition is similar to the proof of Proposition 4.2.4.

Proof. Let C be an arbitrary multilayer partition such that $\mathrm{Pers}(C)|_s < N$ for some $s \in \{1,\ldots,|\mathcal{T}|\}$. We show that there exists a value $\omega_\infty > 0$ of the inter-layer coupling parameter ω such that C is never optimal for $\omega > \omega_\infty$. We first rewrite the quality function as

$$Q(C|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};\omega) = \beta_1 + 2\omega(N(|\mathcal{T}|-1) - A)\,,$$

where $\beta_1 = \sum_{s=1}^{|\mathcal{T}|}\sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s},c_{j_s})$ and $A \geq 1$ because $\mathrm{Pers}(C) < N(|\mathcal{T}|-1)$ by assumption. Now consider the set of values on the diagonal blocks of the multilayer modularity matrix $\boldsymbol{\mathcal{B}}$:

$$\boldsymbol{\mathcal{B}}_{\mathrm{diag}} = \big\{B_{ijs}|i,j \in \{1,\ldots,N\}, s \in \{1,\ldots,|\mathcal{T}|\}\big\}\,, \tag{4.8}$$

and let $\max(\boldsymbol{\mathcal{B}}_{\mathrm{diag}})$ and $\min(\boldsymbol{\mathcal{B}}_{\mathrm{diag}})$, respectively, denote the maximum and minimum values of the set $\boldsymbol{\mathcal{B}}_{\mathrm{diag}}$. Without loss of generality,[7] we assume that $\min(\boldsymbol{\mathcal{B}}_{\mathrm{diag}}) < 0$

[6] For example, one could replace $N(|\mathcal{T}|-1)$ in (4.7) by $N(|\mathcal{T}|-1) - \mathrm{Pers}(\mathcal{C}_{\max}(0))$, where $\mathrm{Pers}(\mathcal{C}_{\max}(0))$ denotes the maximum value of persistence that one can obtain by combining sets in each partition of $\mathcal{C}_{\max}(0)$ without changing the partitions induced on individual layers. Proposition 4.2.4 still holds if one takes $\omega_0 = \Delta Q/\big[2\big(N(|\mathcal{T}|-1) - \mathrm{Pers}(\mathcal{C}_{\max}(0))\big)\big]$, where $\Delta Q/\big[2\big(N(|\mathcal{T}|-1) - \mathrm{Pers}(\mathcal{C}_{\max}(0))\big)\big] > \Delta Q/\big[2N(|\mathcal{T}|-1)\big]$, because $\mathrm{Pers}(\mathcal{C}_{\max}(0)) \geq |\mathcal{T}|-1$ in any multilayer network.

[7] If $\min(\boldsymbol{\mathcal{B}}_{\mathrm{diag}})$ and $\max(\boldsymbol{\mathcal{B}}_{\mathrm{diag}})$ have the same sign or if either is equal to zero, then every diagonal block of $\boldsymbol{\mathcal{B}}$ either has all nonnegative entries or all nonpositive entries. In both cases, an optimal partition $C^{\omega}_{\max}$ has maximal persistence for any value of $\omega > 0$, because $C^{\omega}_{\max}|_s$ is given either by a single community or by N singleton communities for all s. Consequently, by Proposition 4.2.3, $\mathrm{Pers}(C^{\omega}_{\max})|_s = N$ for all s.

and $\max(\boldsymbol{\mathcal{B}}_{\text{diag}}) > 0$. Let C' be any multilayer partition with a maximal value of persistence. It then follows that

$$Q(C'|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; \omega) = \beta_2 + 2\omega N(|\mathcal{T}| - 1)$$

for some $\beta_2 \in \mathbb{R}$. Because $A \geq 1$, choosing

$$2\omega > |\mathcal{T}|N^2\big[\max(\boldsymbol{\mathcal{B}}_{\text{diag}}) - \min(\boldsymbol{\mathcal{B}}_{\text{diag}})\big] \geq \beta_1 - \beta_2$$

ensures that C' yields a higher value of multilayer modularity than C for any β_1 and for all $A \in \{1, \ldots, N(|\mathcal{T}| - 1)\}$. □

The following proposition follows directly from Proposition 4.2.5.

Proposition 4.2.6. *There exists $\omega_\infty > 0$ such that*

$$\textit{For all } r \in \{1, \ldots, |\mathcal{T}|\}, C^{\omega}_{\max}|_r \textit{ is a solution of } \max_{C \in \mathcal{C}} Q\left(C|\sum_{s=1}^{|\mathcal{T}|} \boldsymbol{B}_s\right)$$

for all $\omega > \omega_\infty$.

Propositions 4.2.5 and 4.2.6 imply the existence of a "boundary value" for ω above which single-layer partitions induced by optimal multilayer partitions (1) are the same on all layers and (2) are optimal solutions for the single-layer modularity-maximization problem defined on the mean modularity matrix.

Proof. Suppose that $\omega > \omega_\infty$, where ω_∞ is as defined in Proposition 4.2.5, and let $C^{\omega}_{\max} \in \mathcal{C}_{\max}(\omega)$. By Proposition 4.2.5, it then follows that $\text{Pers}(C^{\omega}_{\max}) = N(|\mathcal{T}| - 1)$ and community assignments in $C^{\omega}_{\max}$ are the same across layers. Consequently, for $\omega > \omega_\infty$

$$\begin{aligned}
C^* &= \underset{C \in \mathcal{C}}{\text{argmax}} \sum_{s=1}^{|\mathcal{T}|} \sum_{i,j}^{N} B_{ijs}\delta(c_{i_s}, c_{j_s}) + 2\omega \text{Pers}(C) \\
\Leftrightarrow C^* &= \underset{C \in \mathcal{C}}{\text{argmax}} \sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs}\delta(c_i, c_j) + 2\omega N(|\mathcal{T}| - 1) \\
\Leftrightarrow C^* &= \underset{C \in \mathcal{C}}{\text{argmax}} \sum_{i,j}^{N} \left(\sum_{s=1}^{|\mathcal{T}|} B_{ijs}\right)\delta(c_i, c_j)\,,
\end{aligned}$$

where c_i denotes the community assignment of node i in all layers. □

The next two propositions formalize the intuition that an optimal multilayer partition measures a trade-off between time-independent community structure within layers (i.e., intra-layer modularity) and persistence of community structure across layers.

Proposition 4.2.7. *Let $\omega_1 > \omega_2 > 0$. For all $C^{\omega_2}_{\max} \in \mathcal{C}_{\max}(\omega_2)$, one of the following two conditions must hold:*

$$\begin{aligned} &(1) \quad C^{\omega_2}_{\max} \in \mathcal{C}_{\max}(\omega_1)\,, \\ \textit{or } &(2) \quad \mathrm{Pers}(C^{\omega_2}_{\max}) < \mathrm{Pers}(C^{\omega_1}_{\max}) \textit{ for all } C^{\omega_1}_{\max} \in \mathcal{C}_{\max}(\omega_1)\,. \end{aligned}$$

Proof. Let $C^{\omega_2}_{\max} \in \mathcal{C}_{\max}(\omega_2)$. If $C^{\omega_2}_{\max} \in \mathcal{C}_{\max}(\omega_1)$, then condition (1) is satisfied. Suppose that $C^{\omega_2}_{\max} \notin \mathcal{C}_{\max}(\omega_1)$, and assume that $\mathrm{Pers}(C^{\omega_2}_{\max}) \geq \mathrm{Pers}(C^{\omega_1}_{\max})$ for some $C^{\omega_1}_{\max} \in \mathcal{C}_{\max}(\omega_1)$. By definition of optimality, $C^{\omega_2}_{\max} \notin \mathcal{C}_{\max}(\omega_1)$ implies that

$$Q(C^{\omega_2}_{\max}|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; \omega_1) < Q(C^{\omega_1}_{\max}|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; \omega_1)\,, \tag{4.9}$$

where $\omega_1 > \omega_2$ by hypothesis. By writing

$$Q(C^{\omega_k}_{\max}|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; \omega_{k'}) = \sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs}\delta(c^{\omega_k}_{i_s}, c^{\omega_k}_{j_s}) + 2\omega_{k'}\mathrm{Pers}(C^{\omega_k}_{\max})\,,$$

where $c^{\omega_k}_{i_s}$ is the community assignment of node i_s in $C^{\omega_k}_{\max}$ and $k, k' \in \{1, 2\}$; and by substituting ω_1 by $\omega_2 + \Delta$ for some $\Delta > 0$, one can show that the inequality (4.9) implies

$$Q(C^{\omega_2}_{\max}|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; \omega_2) < Q(C^{\omega_1}_{\max}|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; \omega_2)\,,$$

which contradicts the optimality of $C^{\omega_2}_{\max}$. □

One can similarly prove the following proposition.

Proposition 4.2.8. *Let $\omega_1 > \omega_2 > 0$. For all $C^{\omega_2}_{\max} \in \mathcal{C}_{\max}(\omega_2)$, one of the following two conditions must hold:*

$$\begin{aligned} &(1) \quad C^{\omega_2}_{\max} \in \mathcal{C}_{\max}(\omega_1)\,, \\ \textit{or } &(2) \quad Q(C^{\omega_2}_{\max}|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; 0) > Q(C^{\omega_1}_{\max}|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; 0) \textit{ for all } C^{\omega_1}_{\max} \in \mathcal{C}_{\max}(\omega_1)\,. \end{aligned}$$

Proof. Let $C^{\omega_2}_{\max} \in \mathcal{C}_{\max}(\omega_2)$. If $C^{\omega_2}_{\max} \in \mathcal{C}_{\max}(\omega_1)$, then condition (1) is satisfied. Suppose that $C^{\omega_2}_{\max} \notin \mathcal{C}_{\max}(\omega_1)$, and assume that

$$Q(C^{\omega_2}_{\max}|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};0) \leq Q(C^{\omega_1}_{\max}|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};0) \tag{4.10}$$

for some $C^{\omega_1}_{\max} \in \mathcal{C}_{\max}(\omega_1)$. By definition of optimality, $C^{\omega_2}_{\max} \notin \mathcal{C}_{\max}(\omega_1)$ implies that

$$Q(C^{\omega_2}_{\max}|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};\omega_1) < Q(C^{\omega_1}_{\max}|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};\omega_1)\,, \tag{4.11}$$

where $\omega_1 > \omega_2$ by hypothesis. By writing

$$Q(C^{\omega_k}_{\max}|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};\omega_1) = Q(C^{\omega_k}_{\max}|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};0) + 2\omega_1\mathrm{Pers}(C^{\omega_k}_{\max}),$$

where $k \in \{1,2\}$ and by using (4.10), one can show that (4.11) implies that

$$Q(C^{\omega_2}_{\max}|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};\omega_2) < Q(C^{\omega_1}_{\max}|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};\omega_2).$$

for all $\omega_2 < \omega_1$. This contradicts the optimality of $C^{\omega_2}_{\max}$. □

(To develop intuition for Propositions 4.2.7 and 4.2.8, it is helpful to think of a multilayer quality function $Q(C|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};\omega)$ for a given partition C as a linear function of ω that crosses the vertical axis at $Q(C|\boldsymbol{B}_1,\ldots,\boldsymbol{B}_{|\mathcal{T}|};0)$ with slope $\mathrm{Pers}(C)$ (see, for e.g., the last panel of Fig. 4.4 and Fig. 4.5).)

The next three corollaries follow straightforwardly from Propositions 4.2.7 and 4.2.8. The first states that the highest achievable value of persistence for an optimal partition obtained with a given value of inter-layer coupling is a non-decreasing function in ω. The second states that the highest achievable value of intra-layer modularity for an optimal partition obtained with a given value of inter-layer coupling is a non-increasing function in ω. The third corollary states that if two distinct values of ω have the same set of optimal partitions, then this set is also optimal for all intermediate values.

Corollary 4.2.9. *Let $\omega_1 > \omega_2 > 0$. Then*

$$\mathrm{Pers}(\mathcal{C}_{\max}(\omega_1)) \geq \mathrm{Pers}(\mathcal{C}_{\max}(\omega_2))\,,$$

where $\mathrm{Pers}(\mathcal{C}_{\max}(\omega)) := \max\left\{\mathrm{Pers}(C^{\omega}_{\max}), C^{\omega}_{\max} \in \mathcal{C}_{\max}(\omega)\right\}$.

Corollary 4.2.10. *Let* $\omega_1 > \omega_2 > 0$. *Then*

$$Q\left(\mathcal{C}_{\max}(\omega_1)|\boldsymbol{B_1},\dots,\boldsymbol{B}_{|\mathcal{T}|};0\right) \leq Q\left(\mathcal{C}_{\max}(\omega_2)|\boldsymbol{B_1},\dots,\boldsymbol{B}_{|\mathcal{T}|};0\right)\,,$$

where $Q(\mathcal{C}_{\max}(\omega)|\boldsymbol{B_1},\dots,\boldsymbol{B}_{|\mathcal{T}|};0) := \max\left\{Q(C^{\omega}_{\max}|\boldsymbol{B_1},\dots,\boldsymbol{B}_{|\mathcal{T}|};0), C^{\omega}_{\max} \in \mathcal{C}_{\max}(\omega)\right\}$.

Corollary 4.2.11. *Assume that* $\mathcal{C}_{\max}(\omega_1) = \mathcal{C}_{\max}(\omega_2)$ *for* $\omega_1 > \omega_2 > 0$. *Then*

$$\mathcal{C}_{\max}(\omega_1) = \mathcal{C}_{\max}(\omega) = \mathcal{C}_{\max}(\omega_2) \text{ for all } \omega \in (\omega_2, \omega_1)\,.$$

One can extend the proofs of Propositions 4.2.1–4.2.8 so that they apply for inter-layer coupling that is uniform between each pair of contiguous layers but may differ from pair to pair. In other words, one can obtain similar results for the maximization problem

$$\max_{C\in\mathcal{C}}\left[\sum_{s=1}^{|\mathcal{T}|}\sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s},c_{j_s}) + 2\sum_{s=1}^{|\mathcal{T}|-1}\omega_s \text{Pers}(C)|_s\right]\,.$$

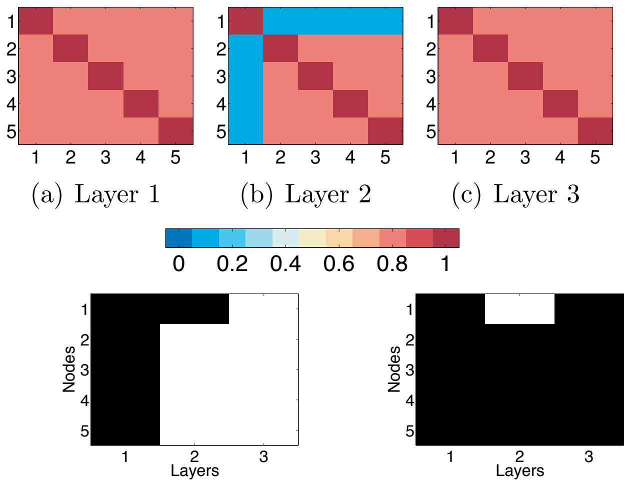
Propositions 4.2.1–4.2.6 trivially extend to this case, and one can extend Propositions 4.2.7–4.2.8 if (for example) one assumes that $\omega^1_s > \omega^2_s > 0$ for all $s \in \{1,\dots,|\mathcal{T}|-1\}$, where $\boldsymbol{\omega}^{(1)}$ and $\boldsymbol{\omega}^{(2)}$ are $(|\mathcal{T}|-1)$-dimensional vectors. (For example, one could set $\boldsymbol{\omega}^{(2)} = \omega\boldsymbol{\omega}^{(1)}$ and vary $\omega > 0$.)

4.2.3 Computational issues

4.2.3.1 Louvain

We now examine issues that can arise when using the Louvain heuristic (see Section 2.3.2) to maximize multilayer modularity (2.24).

Underemphasis of persistence Consider the example network in Fig. 4.6, which is a 3-layer network that has 5 nodes in each layer. Suppose that all nodes are strongly connected to each other in layers 1 and 3, and that the edge weight between node 1_2 and nodes $\{2_2, 3_2, 4_2, 5_2\}$ is smaller in layer 2 than the edge weight between node 1_s and nodes $\{2_s, 3_s, 4_s, 5_s\}$ when $s = 1, 3$. We use the uniform null network with $\gamma = 0.5$ and set $\omega = 0.1$. This produces a multilayer modularity matrix in which all the single-layer modularity entries B_{ijs} except those of node 1_2 are positive and exceed the value of inter-layer coupling. Suppose that one loops over the nodes ordered from 1 to $N|\mathcal{T}|$ in the first phase of the Louvain heuristic. The initial partition consists of $N|\mathcal{T}|$ singletons, and each node is then moved to



(a) Layer 1 (b) Layer 2 (c) Layer 3

(d) Multilayer partition before post-processing

(e) Multilayer partition after post-processing

Figure 4.6: Toy example illustrating the effect of post-processing on a multilayer partition by increasing multilayer modularity via community-assignment swaps that increase the value of persistence but do not change intra-layer partitions. The colors in panels (a)–(c) scale with the entries of the adjacency matrix. Panel (d) (respectively, panel (e)) represents the output multilayer partition obtained with Louvain before (respectively, after) post-processing. The horizontal axis represents the layers, and the vertical axis represents the nodes. The shading in panels (d,e) represents the community assignments of nodes in each layer.

the set that maximally increases modularity. The partition at the end of phase 1 is $\{\{1_1, 2_1, 3_1, 4_1, 5_1, 1_2\}, \{2_2, 3_2, 4_2, 5_2\}, \{1_3, 2_3, 3_3, 4_3, 5_3\}\}$. In phase 2, the second and third sets merge to form a single set,[8] and the Louvain heuristic gets trapped in a local optimum in which the smaller set of nodes (i.e., $\{1_1\}$) remains in the same community across layers 1 and 2 and the larger set of nodes (i.e., $\{2_1, 3_1, 4_1, 5_1\}$) changes community. We show this multilayer partition in Fig. 4.6(d). Repeating this experiment 1000 times using a randomized node order at the start of each iteration of phase 1 of the Louvain heuristic yields the same multilayer partition. One can modify this multilayer partition to obtain a new partition with a larger value of multilayer modularity by increasing the value of persistence across layers without changing intra-layer partitions (we use this idea in the proof of Proposition 4.2.1). We show an example of this situation in Fig. 4.6(e).

In Fig. 4.6(d), we illustrate the above issue visually via abrupt changes in col-

[8] Note that combining the first and second set into a single set decreases modularity because the value of inter-layer coupling is too small to compensate for the decrease in intra-layer contributions to modularity.

ors between layers (these are more noticeable in larger networks). Such changes are misleading because they imply a strong decrease in persistence that might not be accompanied by a strong change in intra-layer partitions. In Fig. 4.6(d), for example, the intra-layer partitions differ in the community assignment of only a single node. To mitigate this problem, we apply a post-processing function to all output partitions that maximally increases persistence between layers without changing the partitions that are induced on each layer (thereby producing a partition with a larger value of multilayer modularity). We do this by relabeling the community assignments of nodes in each layer such that (1) the number of nodes that remain in the same community between consecutive layers is maximally increased and (2) the partition induced on each layer by the original multilayer partition is unchanged. One can implement this post-processing procedure using the *Hungarian algorithm* [113]. Consider a partition $C' \in \mathcal{C}$ that one wants to post-process. The partition C' induces $|\mathcal{T}|$ intra-layer partitions $\{C'|_s, s = 1, \ldots, |\mathcal{T}|\}$ on the multilayer network. One can maximize $\text{Pers}(C')$ without changing the intra-layer partitions induced by C' by maximizing $\text{Pers}(C')|_s$ independently between all pairs of consecutive layers (because inter-layer coupling is ordinal). For every pair of consecutive layers, this yields the following maximization problem

$$\begin{aligned}&\max_{C \in \mathcal{C}} \text{Pers}(C)\\ &\text{such that } C|_s = C'|_s \text{ and } C|_{s+1} = C'|_{s+1}\,,\end{aligned}$$

where $\mathcal{C}$ is the set of $2N$-node partitions. One can reformulate this maximization problem as a *weighted bipartite matching* problem. Nodes in the bipartite network are the communities of C' restricted to layers s and $s+1$ (i.e., the sets in $C'|_s$ and $C'|_{s+1}$) and the edge weight between $C_k|_s$ and $C_k|_{s+1}$, $C_k \in C'$, is

$$|\{i \in \{1, \ldots, N\} : i_s \in C_k|_s \text{ and } i_{s+1} \in C_k|_{s+1}\}| \in \{0, \ldots, N\}.$$

Finding a maximum weight matching in a weighted bipartite graph is called the *assignment problem* in combinatorial optimization and one can solve it using the Hungarian algorithm [58, 142].

Abrupt drop in the number of intra-layer merges The Louvain heuristic faces a second problem in multilayer networks. When the value of inter-layer coupling satisfies

$$\omega > \max(\boldsymbol{\mathcal{B}}_{\text{diag}})\,, \tag{4.12}$$

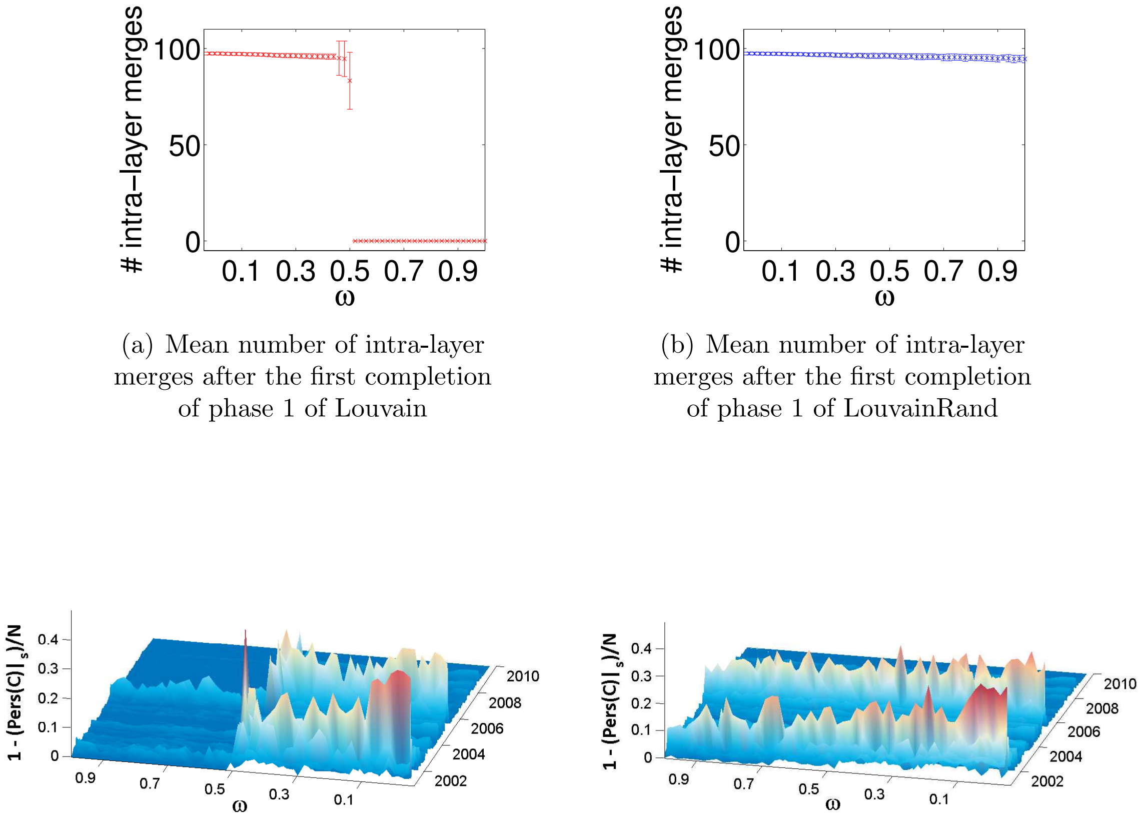
where $\boldsymbol{\mathcal{B}}_{\text{diag}}$ is the set of values on the diagonal blocks of $\boldsymbol{\mathcal{B}}$ defined in equation (4.8), the inter-layer contributions to multilayer modularity are larger than the intra-layer contributions for all pairs of nodes. Consequently, only inter-layer merges occur during the first completion of phase 1 in the Louvain algorithm. In Fig.4.7(a), we illustrate this phenomenon using AssetClasses. The mean number of intra-layer merges drops from roughly $N = 98$ (almost every node contains at least one other node from the same layer in its set) to 0 once $\omega > \max(\boldsymbol{\mathcal{B}}_{\text{diag}})$, with $\max(\boldsymbol{\mathcal{B}}_{\text{diag}}) \approx 0.455$ in this case. For ω values larger than $\max(\boldsymbol{\mathcal{B}}_{\text{diag}})$, every set after the first completion of phase 1 only contains copies of each node in different layers and, in particular, does not contain nodes from the same layer. This can yield abrupt changes in the partitions induced on individual layers of the output multilayer partition

In Fig. 4.7(c), we show an example using AssetClasses of how the above issue can lead to an abrupt change in a quantitative measure computed from a multilayer output partition obtained with the Louvain heuristic. We note that the mean size of sets (averaged over 100 runs) after the first completion of phase 1 of the Louvain algorithm for $\omega > \max(\boldsymbol{\mathcal{B}}_{\text{diag}})$ is relatively small for AssetClasses. (The mean is 3 nodes per set, and the maximum possible number of nodes per set is $|\mathcal{T}| = 238$, because each of these sets only contains copies of the same node when $\omega > \max(\boldsymbol{\mathcal{B}}_{\text{diag}})$.) Nevertheless, as ω increases past $\max(\boldsymbol{\mathcal{B}}_{\text{diag}})$, there is a sudden drop in the value of $(1 - \text{Pers}(C)|_s/N)$ between consecutive layers in the output partition [see Fig. 4.7(c)]. Nonzero values of $(1-\text{Pers}(C)|_s/N)$ indicate that community assignments have changed between layers s and $s+1$ (by Proposition 4.2.3). Roughly speaking, Fig. 4.7(c) suggests that when $\omega > \max(\boldsymbol{\mathcal{B}}_{\text{diag}})$, one abruptly moves away from the situation $\omega \approx \omega_0$ to a scenario that is closer to $\omega \approx \omega_\infty$. We show plots of intra-layer modularity, persistence before post-processing, and persistence after post-processing, for an output partition obtained with Louvain in Fig. 4.8(a,b,c).

The above phenomenon manifests when the values of inter-layer coupling are large relative to the entries of $\boldsymbol{\mathcal{B}}_{\text{diag}}$. In the correlation multilayer networks that we consider (or in multilayer networks with single-layer adjacency matrices in $\{0,1\}^{N\times N}$), entries of the adjacency matrix satisfy $|A_{ijs}| \leq 1$. Assuming that one uses the modularity quality function on each layer and that $P_{ijs} \geq 0$ (e.g., $P_{ijs} = \langle \boldsymbol{A}_s \rangle$), this implies that

$$\max(\boldsymbol{\mathcal{B}}_{\text{diag}}) \leq 1 \text{ for all } \gamma \in [\gamma^-, \gamma^+] \,.$$

For networks in which the modularity cost of changing intra-layer partitions in favor of persistence is large in comparison to the values of $\max(\boldsymbol{\mathcal{B}}_{\text{diag}})$, it might be desirable



(a) Mean number of intra-layer merges after the first completion of phase 1 of Louvain

(b) Mean number of intra-layer merges after the first completion of phase 1 of LouvainRand

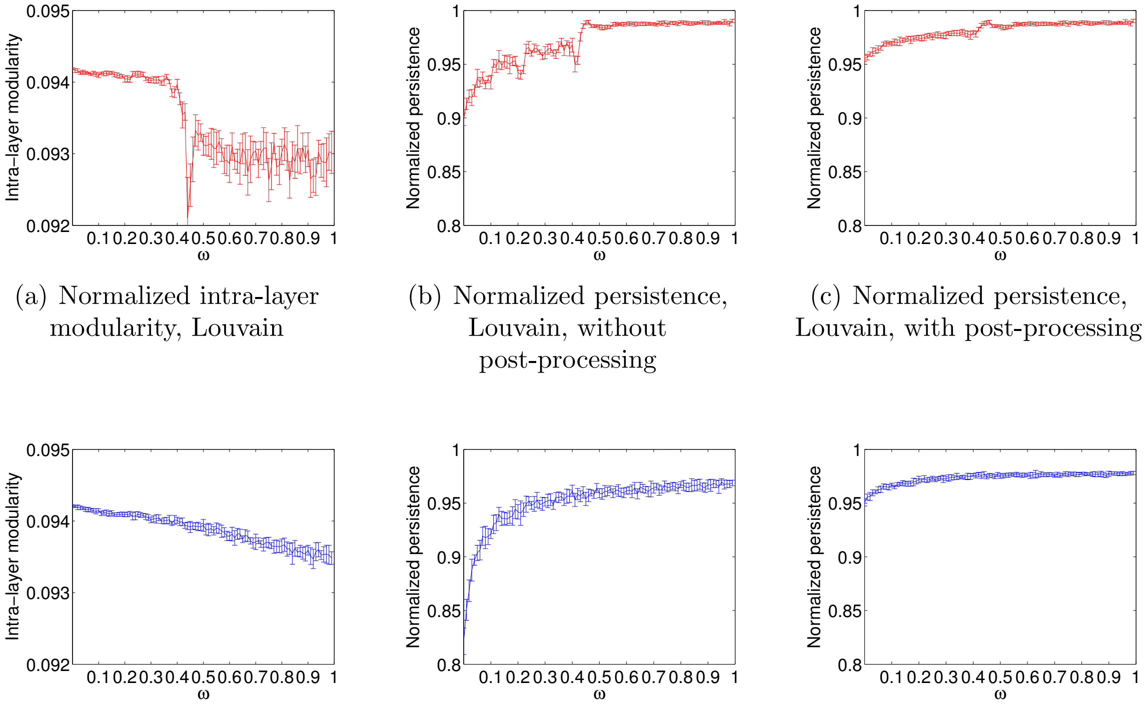
(c) Mean number of nodes that change community assignment between layers in the output partition of Louvain

(d) Mean number of nodes that change community assignment between layers in the output partition of LouvainRand

Figure 4.7: Comparison between the Louvain and LouvainRand heuristic algorithms. The sample of inter-layer coupling values is the set $\{0, 0.02, \ldots, 0.98, 1\}$ with a discretization step of 0.02 between each pair of consecutive values. (a,b) The number of nodes that have been merged with at least one node from the same layer after the first completion of phase 1 of (a) the Louvain heuristic and (b) the LouvainRand heuristic. For each heuristic, we show the average number of nodes over $|\mathcal{T}| = 238$ layers and 100 iterations. The error bars in panels (a,b) indicate standard deviations. (c,d) The value of $1 - \text{Pers}(C)|_s/N$ in an output multilayer partition averaged over 100 runs of (c) the Louvain heuristic and (d) the LouvainRand heuristic.

to use $\omega > 1$ to gain insight into a network's multilayer community structure (e.g., this occurs in both toy examples of Section 4.2.1).

To mitigate this problem, we change the condition for merging nodes in the Louvain heuristic. Instead of moving a node to a community that maximally increases modularity, we move a node to a community chosen uniformly at random from those that increase modularity. We call this heuristic *LouvainRand* [2], and we illustrate the results of using it on AssetClasses in Figs. 4.7(b,d). Although LouvainRand can increase the output variability (by increasing the search space of the optimiza-



(a) Normalized intra-layer modularity, Louvain

(b) Normalized persistence, Louvain, without post-processing

(c) Normalized persistence, Louvain, with post-processing

(d) Normalized intra-layer modularity, LouvainRand

(e) Normalized persistence, LouvainRand, without post-processing

(f) Normalized persistence, LouvainRand, with post-processing

Figure 4.8: We perform 10 runs of Louvain and LouvainRand on AssetClasses for each value of $\omega \in \{0.01, \ldots, 0.99, 1\}$ (this gives a total of 100 ω values). We show (a) nomalized intra-layer modularity, (b) normalized persistence before post-processing, and (c) normalized persistence after post-processing with Louvain; and (d) normalized intra-layer modularity, (e) normalized persistence before post-processing, and (f) normalized persistence after post-processing with LouvainRand. (Recall that intra-layer modularity is not affected by post-processing.) We normalize persistence by $N(|\mathcal{T}| - 1)$ and intra-layer modularity by $\sum_{s=1}^{|\mathcal{T}|} \mathbf{1}^T \boldsymbol{A}_s \mathbf{1}$. All values are averages over 10 runs of the corresponding algorithm. The error bars in all panels indicate standard deviations.

tion process), it seems to mitigate the problem for multilayer networks with ordinal, diagonal, and uniform coupling. We show plots of intra-layer modularity, persistence before post-processing, and persistence after post-processing, for an output partition obtained with LouvainRand in Fig. 4.8(d,e,f) averaged over 10 runs of the algorithm. We carry out further numerical comparisons between Louvain and LouvainRand on benchmark multilayer networks in Chapter 5.

4.2.3.2 Spectral bipartitioning

We end our discussion on computational issues with a brief comment on Newman's spectral bipartitioning heuristic in Section 2.3.3 for multilayer networks with ordinal, diagonal, and uniform inter-layer coupling. Consider a multilayer modularity matrix

$$\boldsymbol{\mathcal{B}}_\omega = \begin{bmatrix} \boldsymbol{B}_1 & \omega\boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \omega\boldsymbol{I} & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \ddots & \ddots & \ddots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \omega\boldsymbol{I} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \omega\boldsymbol{I} & \boldsymbol{B}_{|\mathcal{T}|} \end{bmatrix}, \tag{4.13}$$

where we include the subscript ω for clarity. Every single-layer modularity matrix $\boldsymbol{B}_s$ is symmetric and thus has a set of N orthonormal eigenvectors $\{\boldsymbol{\mu}_{s1}, \dots, \boldsymbol{\mu}_{sN}\}$ with corresponding eigenvalues $\alpha_{s1} \leq \dots \leq \alpha_{sN}$ [192]. It follows that the multilayer modularity matrix $\boldsymbol{\mathcal{B}}_0$ has a set of $N|\mathcal{T}|$ orthonormal eigenvectors given by

$$\mathcal{V}(\boldsymbol{\mathcal{B}}_0) = \left\{ \begin{bmatrix} \boldsymbol{\mu}_{11} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix}, \dots, \begin{bmatrix} \boldsymbol{\mu}_{1N} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix}; \dots; \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \\ \boldsymbol{\mu}_{|\mathcal{T}|1} \end{bmatrix}, \dots, \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \\ \boldsymbol{\mu}_{|\mathcal{T}|N} \end{bmatrix} \right\},$$

with a corresponding set of eigenvalues $\{\alpha_{11}, \dots, \alpha_{1N}; \dots; \alpha_{|\mathcal{T}|1}, \dots, \alpha_{|\mathcal{T}|N}\}$. Denote the i^{th} eigenvector (respectively, eigenvalue) of $\boldsymbol{\mathcal{B}}_0$ by $\boldsymbol{v}_{0i}$ (respectively, λ_{0i}) and consider the eigenvalue problem

$$\boldsymbol{\mathcal{B}}_\omega \boldsymbol{v}_{\omega i} = \lambda_{\omega i} \boldsymbol{v}_{\omega i}\,. \tag{4.14}$$

In the discussion to follow, we assume that ω is "small" and we use first order perturbation analysis to examine how an off-diagonal perturbation of $\boldsymbol{\mathcal{B}}_0$ with uniform, diagonal, and ordinal inter-layer coupling can affect the eigenvalues and eigenvectors of the multilayer modularity matrix. Write (4.13) as $\boldsymbol{\mathcal{B}}_\omega = \boldsymbol{\mathcal{B}}_0 + \omega\boldsymbol{\mathcal{B}}^{(1)}$, $\boldsymbol{v}_{\omega i} = \boldsymbol{v}_{0i} + \omega\boldsymbol{v}^{(1)} + O(\omega^2)$, and $\lambda_{\omega i} = \lambda_{0i} + \omega\lambda^{(1)} + O(\omega^2)$. Substituting these three expressions in (4.14) and equating first order terms, one obtains the equality

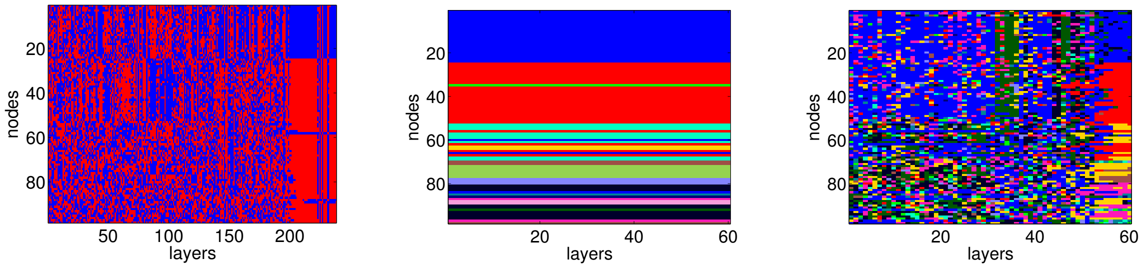
$$\boldsymbol{\mathcal{B}}_0\boldsymbol{v}^{(1)} + \boldsymbol{\mathcal{B}}^{(1)}\boldsymbol{v}_{0i} = \lambda^{(1)}\boldsymbol{v}_{0i} + \lambda_{0i}\boldsymbol{v}^{(1)}\,.$$

By writing $\boldsymbol{v}^{(1)}$ as a linear combination of $\{\boldsymbol{v}_{01}, \dots, \boldsymbol{v}_{0N|\mathcal{T}|}\}$ and using the orthonormality of the vectors $\boldsymbol{v}_{0j}$, one obtains the following expressions for $\boldsymbol{v}^{(1)}$ and $\lambda^{(1)}$

$$\boldsymbol{v}^{(1)} = \sum_{i \neq j} \frac{\boldsymbol{v}_{0j}^T\big(\boldsymbol{\mathcal{B}}^{(1)}\boldsymbol{v}_{0i}\big)}{\lambda_{0i} - \lambda_{0j}} \boldsymbol{v}_{0j}\,, \tag{4.15}$$

$$\lambda^{(1)} = \boldsymbol{v}_{0i}^T\big(\boldsymbol{\mathcal{B}}^{(1)}\boldsymbol{v}_{0i}\big)\,, \tag{4.16}$$

under the assumption that eigenvalues of different layers are pairwise distinct.



(a) Output multilayer partition on AssetClasses, $|C| = 2$

(b) Output partition for toy example with identical layers, $|C| = 14$

(c) Output partition for toy example with identical layers in all but the last layer, $|C| = 12$

Figure 4.9: We use the uniform null network with $\gamma = 1$ and $\omega = 0.1$. We indicate the number of communities in the corresponding output partition in each panel. (a) Example output partition with spectral bipartitioning on AssetClasses, (b) Output partition with spectral bipartitioning on a 60 layer network in which every layer is given by the first layer $\boldsymbol{A}_1$ of AssetClasses (see Fig. 3.4(a)), and (c) Example output partition with spectral bipartitioning on a 60 layer network in which every layer $s < 60$ is given by the first layer $\boldsymbol{A}_1$ of AssetClasses and the last layer $s = 60$ is given by the last layer $\boldsymbol{A}_{238}$ (see Fig. 3.4(f)) of AssetClasses, which has a leading eigenvalue that is larger than the leading eigenvalue of $\boldsymbol{A}_1$.

In our case, the matrix $\boldsymbol{\mathcal{B}}^{(1)}$ has the form

$$\boldsymbol{\mathcal{B}}^{(1)} = \begin{bmatrix} \mathbf{0}_N & \boldsymbol{I} & \mathbf{0} & \dots & \mathbf{0} \\ \boldsymbol{I} & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \boldsymbol{I} \\ \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{I} & \mathbf{0}_N \end{bmatrix} .$$

The sparsity pattern of $\boldsymbol{\mathcal{B}}^{(1)}$ (or more generally, the sparsity pattern of any matrix with zero diagonal blocks) implies that $\boldsymbol{v}_{0i}^T\big(\boldsymbol{\mathcal{B}}^{(1)}\boldsymbol{v}_{0i}\big) = 0$ for all $\boldsymbol{v}_{0i} \in \mathcal{V}(\boldsymbol{\mathcal{B}}_0)$, and thus that $\lambda^{(1)} = 0$ in (4.16). The term $\boldsymbol{v}_{0j}^T\big(\boldsymbol{\mathcal{B}}^{(1)}\boldsymbol{v}_{0i}\big)$ in $\boldsymbol{v}^{(1)}$ is only nonzero if $\boldsymbol{v}_{0j}$ is an eigenvector in $\mathcal{V}(\boldsymbol{\mathcal{B}}_0)$ that corresponds to a layer s that is contiguous to the layer that corresponds to $\boldsymbol{v}_{0i}$ (because only the first off-diagonal blocks of $\boldsymbol{\mathcal{B}}^{(1)}$ are nonzero). It follows that $\boldsymbol{v}^{(1)}$ in (4.15) has at most $2N$ nonzero terms.

The point of this discussion is to illustrate that for "small enough" ω the eigenvectors of $\boldsymbol{B}_\omega$ are "close" to those of $\boldsymbol{B}_0$ and thus tend to be "localized" on individual layers (see [188] for similar observations). This can be problematic when using a heuristic that bipartitions a multilayer network based on the sign of the leading eigenvector of $\boldsymbol{B}_\omega$. We show example output multilayer partitions with spectral bipartitioning on AssetClasses in Fig. 4.9.

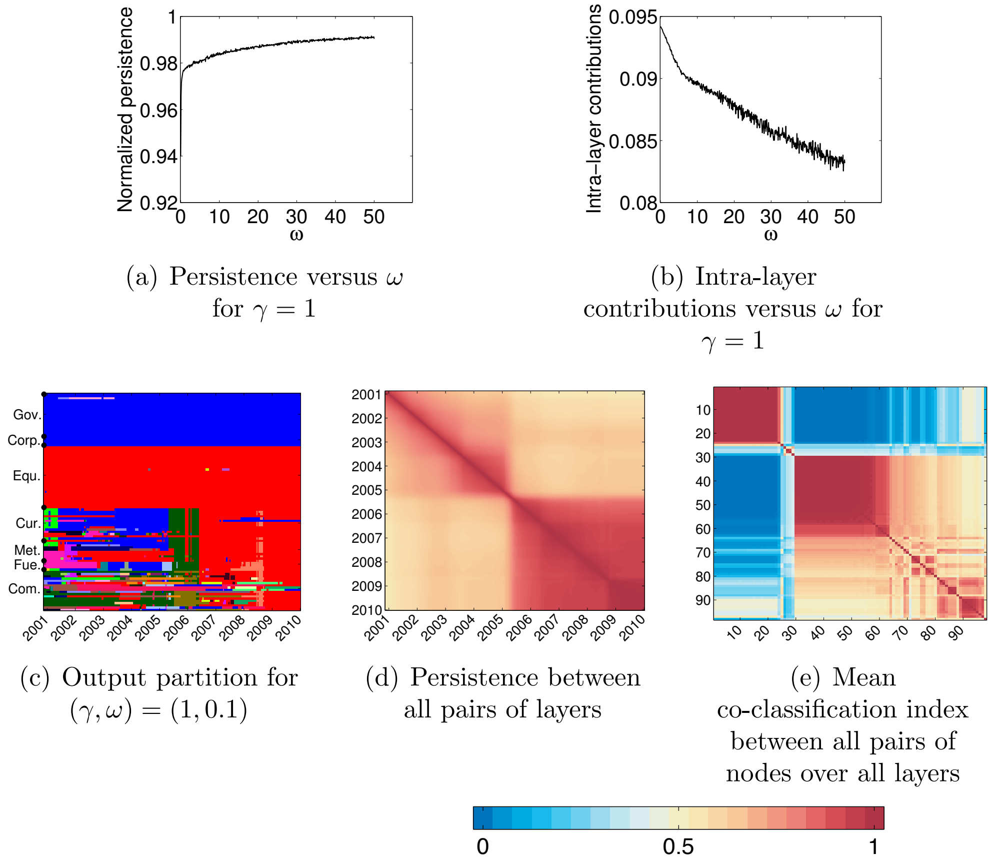
4.2.4 Multilayer community structure in asset correlation networks

To end this chapter, we show preliminary computational experiments with AssetClasses data set that illustrate some of the concepts that we discussed. We fix the value of the resolution parameter γ in all experiments and vary the value of inter-layer coupling ω. We use the uniform null network (i.e., $P_{ijs} = \langle \boldsymbol{A} \rangle_s$) and set $\gamma = 1$. We use the LouvainRand heuristic to identify multilayer partitions and apply our post-processing procedure that increases persistence without changing partitions induced on individual layers to all output multilayer partitions. We showed in Proposition 4.2.5 that for $2\omega > 2\omega_\infty = |\mathcal{T}|N^2\big[\max(\boldsymbol{\mathcal{B}}_{\text{diag}}) - \min(\boldsymbol{\mathcal{B}}_{\text{diag}})\big]$, the set $\mathcal{C}_{\max}(\omega)$ of global optima no longer changes and every optimal partition in this set has maximal persistence.[9] In the AssetClasses data set, $|\mathcal{T}|N^2\left[\max(\boldsymbol{\mathcal{B}}_{\text{diag}}) - \min(\boldsymbol{\mathcal{B}}_{\text{diag}})\right] \leq 2|\mathcal{T}|N^2$, with $N = 98$ and $|\mathcal{T}| = 238$. However, for the purposes of these experiments, we take the set $\{0, 0.1, \dots, 49.9, 50\}$ with a discretization step of 0.1 between consecutive values (giving 501 values in total) as our sample of ω values.

In agreement with the properties derived in Propositions 4.2.7 and 4.2.8, we observe in Fig. 4.10(a) that normalized persistence (given by $\text{Pers}(C)/[N(|\mathcal{T}| - 1)]$) tends to be larger for larger values of inter-layer coupling, and in Fig. 4.10(b) that intra-layer modularity (which we normalize by $\sum_{s=1}^{|\mathcal{T}|}(\boldsymbol{1}^T \boldsymbol{A}_s \boldsymbol{1})$) tends to be smaller for larger values of inter-layer coupling. The increase of persistence and the decrease of intra-layer modularity need not be monotonic, because we are a finding a set of local optima for each value of ω rather than the set of global optima.

In Fig. 4.10(c), we show a sample output of the multilayer partition (which contains 35 communities). (See Section 3.3 for our definitions of the asset-class abbreviations.) Some of the changes in community structure correspond to known events (e.g., Lehman Bankruptcy in September 2008 [marked by an increase in the size of the equity asset class]). Observe that the two largest communities are the ones that contain the government bond assets and the equity assets. In particular, the community that contains equities becomes noticeably larger between 2006 and 2007, and again towards the end of 2008 [after the pink streak between 2008 and 2009 in Fig. 4.10(d)]. For larger values of the resolution parameter, this community instead becomes noticeably larger only in 2008. (By inspecting the correlation matrices in

[9] Note that there can also be smaller values of ω_∞ for which this is true; in other words, we did not show that ω_∞ is the smallest lower bound of the set $\{w : \text{Pers}(C^{\omega}_{\max}) = N(|\mathcal{T}|-1) \text{ for all } C^{\omega}_{\max} \in \mathcal{C}_{\max}(\omega)\}$.



(a) Persistence versus ω for $\gamma = 1$

(b) Intra-layer contributions versus ω for $\gamma = 1$

(c) Output partition for $(\gamma, \omega) = (1, 0.1)$

(d) Persistence between all pairs of layers

(e) Mean co-classification index between all pairs of nodes over all layers

Figure 4.10: Numerical experiments with AssetClasses. We sample the set of interlayer edge weights uniformly from the interval $[0, 50]$ with a discretization step of 0.1 (so there are 501 values of ω in total), and we use the uniform null network (i.e., $P_{ijs} = \langle \boldsymbol{A}_s \rangle$) with $\gamma = 1$. (a) The persistence normalized by $N(|\mathcal{T}| - 1)$ for each value of ω averaged over 20 runs of LouvainRand. (b) The intra-layer modularity $\sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s}, c_{j_s})$ normalized by $\sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} A_{ijs}$ for each value of ω averaged over 20 runs of LouvainRand. (d) Sample output multilayer partition. Each point on the horizontal axis represents a single time window, and each position on the vertical axis is an asset. We order the assets by asset class, and the colors represent communities. (e) Association matrix of normalized persistence values between all pairs of layers averaged over all values of $\omega \in [0, 50]$ in our sample and 20 runs for each value. The normalized persistence between a pair of layers $\{s, r\}$ is $\sum_{i=1}^{N} \delta(c_{is}, c_{ir})/N$. (f) Association matrix indicating the co-classification of nodes averaged over the set of partitions induced on each layer for each value of ω and 20 runs of LouvainRand.

Fig. 3.4 for example, one can see that the increase in correlation between equities and other assets is greater in 2008 than in 2006.)

In Fig. 4.10(d), we show the matrix of mean values of persistence between all pairs of layers. The $(s, r)^{\text{th}}$ entry is the term $\text{Pers}(C)|_{s,r}/N = \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_r})/N$ (4.5), where $s, r \in \{1, \ldots, |\mathcal{T}|\}$ need not be from consecutive layers, averaged all values of $\omega \in [0, 50]$ in our sample, and multiple runs for each value of ω. Instead of only plotting $\text{Pers}(C)|_s$ for consecutive layers, Fig. 4.10(d) gives some indication as to whether nodes change communities between layers s and $s+1$ to join a community that contains a copy of some of these nodes from another time layer (i.e., $\sum_{i=1}^{N} \delta(c_{i_{s+1}}, c_{i_r}) \neq 0$ for some r) or to join a community that does not contain a copy of these nodes in any other time layer (i.e., $\sum_{i=1}^{N} \delta(c_{i_{s+1}}, c_{i_r}) = 0$ for all r). Figure 4.10(d) also gives some insight into whether there are sets of consecutive layers across which persistence values remain relatively large. This may shed light on when connectivity patterns change in a multilayer network. As indicated by the values on the color scale, the values of persistence in Fig. 4.10(d) remain relatively high across all layers (which can partly be explained by the fact that equities and bonds remain in the same community across almost all layers, and these constitute roughly 50% of the node sample). The most noticeable separation into diagonal blocks in the middle of Fig. 4.10(d) corresponds to the change in Fig. 4.10(c) between 2005 and 2006, at which various currencies, metals, and fuels separate from the bond community (blue) to form a dark green community. The smaller diagonal block at the bottom right of Fig. 4.10(d) corresponds to the change in Fig. 4.10(c) after the Lehman Bankruptcy between 2008 and 2009 [after the pink streak in Fig. 4.10(c)].

In Fig. 4.10(e), we show the co-classification index of nodes in partitions induced on individual layers, which we average over layers, all values of $\omega \in [0, 50]$ in our sample, and multiple runs for each value of ω (we re-order the nodes to emphasize diagonal blocks in the association matrix). This figure yields insight into what sets of nodes belong to the same community across layers for increasing values of ω. This may shed light on connectivity patterns that are shared across layers. Unsurprisingly, the first diagonal block mainly corresponds to bond assets and the second diagonal block mainly corresponds to equity assets. Figures 4.10(d,e) complement each other: at a given γ resolution, the former gives an idea about *when* community structure has changed, and the latter gives an idea about *how* it has changed. We use these ideas in Chapter 6, where we perform further numerical experiments on both data sets AssetClasses and SingleAssetClass.

The insight gained from Fig. 4.10 is useful to financial practitioners in that it highlights changes in correlation structure that did not stand out with more standard data analysis techniques, in which connectivity patterns on individual layers are considered independently (e.g., the change between 2005 and 2006, and the pink streak between 2008 and 2009). Furthermore, the multilayer partition in Fig. 4.10(c) can provide a single ordering of the N assets that one can use for visualizing the correlation matrices of AssetClasses between 2000 and 2010 (as done in Fig. 4.2(a)). The ordering used in Fig. 4.2(a) emphasizes block-diagonal structure in the correlation matrix of the last layer of AssetClasses, but may not be appropriate for correlation matrices in other layers. By computing the association matrix for the set of partitions induced on individual layers in the multilayer output partition of Fig. 4.10(c), one can extract a single node ordering that takes into account connectivity patterns in all layers.

We also note that one can repeat the experiments in Fig. 4.10 with a nonuniform choice of interlayer coupling such that values tend to be lower for "dissimilar" correlation matrices and larger for "similar" correlation matrices. For example, a first intuitive choice would be to take the value of inter-layer coupling as the autocorrelation between time series from consecutive time windows. The value of autocorrelation between logarithmic returns is statistically insignificant for our choice of $(|T|, \delta t)$, but one could perform further experiments with shorter time window lengths. One can also perform experiments in which the value of inter-layer coupling is uniform between each pair of contiguous layers but may differ from pair to pair (e.g., one can use a measure of similarity between contiguous correlation matrices, or a measure of similarity between partitions induced on individual layers obtained with $\omega = 0$ and then repeat this process iteratively). One could then repeat the experiments in Fig. 4.10 with ω as a scaling parameter for the nonuniform vector of inter-layer coupling (as described at the end of Section 4.2.2) and compare the results to those of the uniform case.

4.3 Summary

Modularity maximization in temporal multilayer networks is a clustering technique that produces a time-evolving partition of nodes. We have investigated two questions that arise when using this method: (1) the role of null networks in modularity maximization, and (2) the effect of inter-layer edges on the multilayer modularity-maximization problem.

At the heart of modularity maximization is a comparison between what one anticipates and what one observes. The ability to specify what is anticipated is a desirable feature of modularity maximization, because one can explicitly adapt it for different applications. However, one needs to be very careful with one's choice of null network because it determines what one regards as densely connected in a network: different choices in general yield different communities. We illustrated in Section 4.1 for financial correlation networks that one should be cautious when interpreting communities that one obtains with a null network in which the distribution of edge weights is sample-dependent. Such a choice can have a large impact on results, and can lead to misleading conclusions.

Furthermore, we defined a measure that we called *persistence* and showed that an optimal partition in multilayer modularity maximization reflects a trade-off between time-independent community structure within layers (i.e., intra-layer modularity) and persistence of community structure across layers. One can try to exploit this in practice to detect changes in connectivity patterns and shared connectivity patterns in a time-dependent network. In 4.2.2, we proved several properties that describe the effect of uniform, ordinal, and diagonal inter-layer coupling on multilayer modularity maximization. Our multilayer analysis only depends on the form of the maximization problem in (2.24) and still holds if one uses a quality function other than the modularity quality function, provided it has the form. Although our theoretical results do not necessarily apply to the local optima that one attains in practice, they do provide useful guidelines for how to interpret the outcome of a computational heuristic: if a multilayer partition is inconsistent with one of the proven properties, then it must be an artifact of the heuristic and not a feature of the quality function.

Finally, we illustrated an issue that can arise when one uses spectral bipartitioning to solve the multilayer maximization problem. We also showed that the Louvain heuristic can pose two issues when applied to multilayer networks with ordinal, diagonal, and uniform coupling. We proposed ways to try to mitigate the issues with Louvain and showed several numerical experiments on real data as illustrations.

Chapter 5

Temporal Multilayer Benchmark

Community-detection benchmarks for single-layer networks generally consist of generating a network partition, often called a planted partition (see Section 2.1.1), and a network that "fits" a planted partition (e.g., a network sampled from a stochastic block model where each block is a community and the expected edge density within communities is higher than the expected edge density between communities) [45, 49, 73, 88]. While it is perhaps not best practice to assess and compare methods and heuristic algorithms mainly via (real or synthetic) benchmarks, it is common practice in community detection [117, 125, 150, 163, 207]. The main reason for this is twofold. While most would agree that a community should correspond to a set of nodes that is "surprisingly well-connected", there is no agreed-upon definition of community that one can compare against. Different applications warrant different interpretations of "surprisingly well-connected" and different methods were often developed with different definitions in mind [68, 163]. Furthermore, most community-detection methods cannot be solved in polynomial time and popular scalable heuristics currently have few or no theoretical guarantees on how closely an identified partition resembles an optimal partition [37, 77]. It is clear that efforts towards establishing further theoretical grounding for both community-detection methods and heuristics need further development. Nonetheless, community-detection benchmarks can be useful in that they can provide some insight on whether and how certain methods or algorithms may be effective (or, also useful to know, ineffective).

Community detection in multilayer networks is a recent area of research [31, 109] and there is currently no "standard" (synthetic or real) multilayer benchmarks that one can use to compare multilayer community detection methods or heuristics. Furthermore, in contrast to the single-layer case (where there is some consensus), there is currently no consensus on the intuition behind what a community should represent in a multilayer network, and where possibly such an intuition should depend on how the layers are related (e.g., temporal or multiplex [109]) and on what the inter-layer edges represent (e.g., uniform parameter or edge-weight with a physical interpretation).

The purpose of this chapter is to propose a benchmark for community detection in temporal multilayer networks. Our goal is not to model the temporal evolution of communities, but to suggest a way of generating an ordered sequence of single-layer partitions and networks that are correlated in some way. In contrast to single-layer community benchmarks, which one can use to generate a sequence of networks with uncorrelated planted community structure, we incorporate a simple probabilistic model for the persistence of community assignments between successive layers to generate a sequence of single-layer networks with correlated planted community structure.

This chapter is organized as follows. We first give an overview of existing community-detection benchmarks for single-layer and multilayer networks. We then describe our community-detection benchmark for temporal multilayer networks. We highlight some of its properties and comment on the effect of some of its parameters on the resulting benchmark multilayer partition. Finally, we perform several numerical experiments using different methods and computational heuristics to try and gain some insight into the following points: (1) how does varying ω in multilayer modularity maximization affect the recovery of a planted partition for different heuristics and different parameter choices of the benchmark? and (2) how does varying the "relax-rate" in multilayer map minimization (see Section 2.2.3) affect the recovery of a planted partition for different parameter choices of the benchmark?

5.1 Description of existing benchmarks

5.1.1 Single-layer benchmarks

Various benchmarks have been developed for testing community-detection methods and heuristic algorithms in single-layer networks. One of the earlier well-known benchmarks was a planted-partition model introduced by Girvan and Newman (GN) in 2002 [73]. The planted partition consists of 128 nodes divided into four equal-sized communities (of 32 nodes). The benchmark networks are unweighted and edges are generated using a stochastic block model (see Section 2.1.1) for community structure [88]. That is, edges between nodes in the same community are present with probability p_{in} and edges between nodes in different communities are present with probability p_{out}, where $0 \leq p_{\text{out}} < p_{\text{in}}$. The values of p_{in} and p_{out} are chosen such that the expected degree of each node is 16 (i.e., $\mathbb{E}[\,k_i\,] = 32p_{\text{in}} + 96p_{\text{out}} = 16$, so that $0.125 < p_{\text{in}} \leq 0.5$). This benchmark is a special case of the *planted l-partition model* [45], which is a generalization of the *planted bisection model* [99].

On the one hand, the GN benchmark is appealing because it can be used as a test in which the division of nodes into clusters is clear and simple. On the other hand, community detection is performed in practice on real networks and the GN benchmark produces networks with unrealistic properties: the number of nodes is relatively small, the expected strength is the same across all nodes, and the expected community size is the same across all communities [49, 117, 120]. For this reason, various benchmarks have been suggested in an attempt to produce more realistic networks.

Danon *et al.* generalized the GN benchmark in 2006 to allow for heterogenous expected strength and heterogenous community sizes [49]. The planted partition in their generalization consists of N nodes divided into l clusters each with size l_i. Nodes in the same cluster are connected by an edge with probability p_{in} and nodes in different communities are connected by an edge with probability p_{out}, where $0 \leq p_{\text{out}} < p_{\text{in}}$. The authors chose $p_{\text{in}} = F/l_i$, $0 \leq F \leq l_i$, so that the expected number of edges within a community scales linearly with its size rather than the square of its size, as it would in a GN-type stochastic block model. In this benchmark, the heterogeneity of expected strength is fully determined by the heterogeneity in community sizes (in contrast to the LFR benchmark, which we describe below).

The most popular benchmark for single-layer networks is the Lancichinetti–Fortuna -to–Radicchi (LFR) benchmark. A first version was introduced in [120] for generating unweighted networks with heterogenous degrees and heterogeneous community sizes. In this benchmark, the authors fit a network to a partition (rather than the other around) by adopting a procedure that can briefly be summarized as follows. They first sample a degree sequence from a power-law distribution (with an upper and lower bound on degree) and generate an unweighted network with the sampled degree sequence using a configuration model "half-stub" approach, where one generates k_i "half-edges" for each node and subsequently connects half-edges uniformly at random [147]. They then sample a sequence of community sizes that sum to N from a power-law distribution (with an upper and lower bound on community size) and place each node i in exactly one community such that $(1-\mu)k_i$ of its edges fall within communities and μk_i of its edges fall between communities, where $\mu \in [0, 1]$ is termed the *topological mixing parameter*. That is, the fraction of edges that lie inside a node's community is $1 - \mu$, and the fraction of edges with endpoints outside a node's community is μ. To enforce this condition, the authors perform "several rewiring steps" in their implementation that preserve node degrees but that potentially require adjusting community sizes [120]. Two of the authors later generalized this benchmark in [117] to allow for directed and weighted networks, as well as networks with overlapping community structure. Several other authors have also suggested single-layer community-detection benchmarks to account for various network features encountered in applications (e.g., overlapping community structure [176] and spatial networks [175]). There has also been some effort towards highlighting unrealistic features of the LFR benchmark [100] and towards developing more realistic single-layer benchmarks [155].

5.1.2 Multilayer benchmarks

As community detection in multilayer networks is a relatively recent research area, there is still no "standard" multilayer community detection benchmark. We mention a few suggestions that have recently appeared in the literature. An early multilayer benchmark appeared in [56], with the first generalization of map minimization [171] to "multiplex" networks.[1] The authors of [56] compare their method to the generalization of modularity to "multiplex" networks suggested in [140] on the following benchmark multilayer networks. They first generate an unweighted single-layer network and a planted single-layer partition with the LFR benchmark described in the previous section. They then sample L networks from the single-layer network by keeping each edge of the network with probability $1/L$ (the sampled networks are sparser when there are more layers). This generates L single-layer networks that "fit" the LFR-generated planted partition. They repeat this process M times to produce a multilayer network with LM layers and M distinct single-layer planted partitions: the first L layers of the multilayer partition all have one planted single-layer partitition, the second L layers of the multilayer network all have another planted single-layer partitition, and so on. The inter-layer community labeling (i.e., the labeling that determines which nodes remain/change communities across layers) they adopt is the following: a node keeps the same community assignment within all M chains of L layers with the same planted single-layer partition, but changes community assignments between chains.

[1] In the multiplex networks considered in [56], layers represent different types of connections between a set of nodes (e.g., interactions on Facebook and interactions on Twitter) rather than a specific type of connection at different points in time, which we refer to in this thesis as a "temporal multilayer network" or simply "multilayer network".

A multilayer benchmark was suggested for temporal multilayer networks in [78]. The goal of the paper is to put forward a model for two types of community evolution events: "growth"/"shrinkage" and "merge"/"split" [156]. In particular, they propose a model for the temporal evolution of the number of nodes in two communities.[2] They investigate three cases: a first in which two communities can grow and shrink, a second in which two communities can merge and split, and a third which consists of four communities and combines both previous cases (the first pair of communities corresponds to the first case and the second pair of communities to the second). For each layer, they then generate a single-layer network thats fits the partition induced on the layer using a stochastic block model [88]. The authors perform numerical experiments with multilayer modularity for different ω values using LouvainRand on each of these three cases.

There are also suggestions in the literature for temporal multilayer benchmarks that are "node-centric" (as opposed to the "community-centric" model in [78]) and where the correlation between partitions induced on individual layers is governed by a probabilistic model on community assignments [72]. This approach is a simple starting point and is analytically tractable, and it is the approach that we adopt for developing our benchmarks. One can find various other recent generalizations of single-layer stochastic block models to multilayer networks in [83, 158, 160, 193, 203].

5.2 Description of our benchmarks

In this section, we propose a benchmark for community detection in temporal multilayer networks with $|\mathcal{T}|$ layers. Every multilayer partition $C \in \mathcal{C}$ induces $|\mathcal{T}|$ single-layer partitions $\{C|_s, s = 1, \dots, |\mathcal{T}|\}$ on the number of layers. We propose a way of generating a multilayer partition in which the members of the set of $|\mathcal{T}|$ single-layer partitions are correlated through a probabilistic model of persistence. The benchmark consists of two main steps: (1) generate a multilayer partition and (2) generate $|\mathcal{T}|$ single-layer networks that "fit" the $|\mathcal{T}|$ single-layer partitions induced by the multilayer partition. We use block models in our numerical experiments.

[2] The evolution of nodes in one community depends on the number of nodes in the other community as the sum of community sizes must equal N. One would need to bear this in mind if attempting to generalize the model in [78] to more than two communities.

Step 1	Generate a single-layer partition for $s = 1$ by choosing a community assignment for each node in layer 1 using $\mathbb{P}_0^s$.
Step 2	**a.** With probability $p \in [0, 1]$, the community assignment of node i_s remains the same in layer $s+1$ (i.e., $\delta(c_{i_s}, c_{i_{s+1}}) = 1$). **b.** With probability $1-p$ the community assignment of node i_{s+1} is chosen using $\mathbb{P}_0^s$.
Step 3	Repeat step 2 until $s + 1 = \lvert\mathcal{T}\rvert$.

Table 5.1: Steps for generating a temporal multilayer benchmark partition.

5.2.1 Benchmark multilayer partition

5.2.1.1 Description

Suppose we fix the number $|\mathcal{T}|$ of layers and the number N of nodes in each layer. As we did earlier, let $s \in \{1, \ldots, |\mathcal{T}|\}$ denote a layer of the multilayer network, i_s a node $i \in \{1, \ldots, N\}$ in layer s, and c_{i_s} the integer set assignment of node i_s in a multilayer partition $C \in \mathcal{C}$. We generate a benchmark multilayer partition via a three-step process:

1. Generate a single-layer partition for $s = 1$ by choosing a community assignment for each node in layer 1 "at random".

2. a. With probability $p \in [0, 1]$, the community assignment of node i_s remains the same in layer $s + 1$ (i.e., $\delta(c_{i_s}, c_{i_{s+1}}) = 1$).

   b. With probability $1 - p$ the community assignment of node i_{s+1} is chosen "at random".

3. Repeat step 2 until $s + 1 = |\mathcal{T}|$.

We call the *null distribution* the probability distribution that assigns nodes to communities "at random" in Step 1 and Step 2b and we denote the null distribution of layer s by $\mathbb{P}_0^s$. We list the steps for generating a multilayer benchmark partition in Table 5.1 and illustrate them with a toy example in Fig. 5.1. Let $\mathcal{C}_{\text{bench}}(\mathbb{P}_0^s; p)$ be the set of possible benchmark multilayer partitions for a fixed choice of $\{\mathbb{P}_0^s, s = 1, \ldots, |\mathcal{T}|\}$ and $p \in [0, 1]$. Before describing possible choices for $\mathbb{P}_0^s$, we point out features of the benchmark that hold for all $\mathbb{P}_0^s$.

Let $C \in \mathcal{C}_{\text{bench}}(\mathbb{P}_0^s; p)$ and let c_{i_s} be the community assignment of node i_s in C.

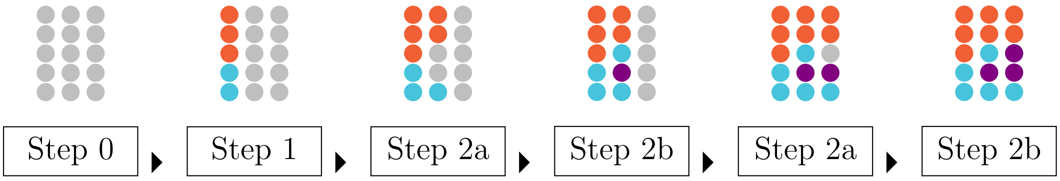


Figure 5.1: Toy example illustrating the steps in Table 5.1 for generating a temporal multilayer benchmark partition. We consider three layers ($|\mathcal{T}| = 3$) and five nodes in each layer ($N = 5$). Each 5×3 rectangular block represents the nodes in the multilayer network at successive steps of the multilayer partition generation process. In each block, the s^{th} column of circles represent the nodes in the s^{th} layer. We order the nodes from 1 to 5 in all layers. Gray cicles represent unassigned nodes and all other colors represent community assignments. Nodes in the same community have the same color. For example, the partition induced on layer 2 is complete in the fourth block. In layer 2, the blue community label gains a node (node 3) and loses a node (node 4), and a new community label appears (purple label).

Proposition 5.2.1. *For all $k \in \{1, \ldots, N|\mathcal{T}|\}$ and $s \in \{1, \ldots, |\mathcal{T}|-1\}$, the following conditions are satisfied:*

$$\mathbb{P}\left[c_{i_{s+1}} = k \middle|\, C|_s\right] = \delta(c_{i_s}, k)p + \mathbb{P}_0^s\left[c_{i_{s+1}} = k\right](1-p)\,, \tag{5.1}$$

$$\mathbb{E}\left[\text{Pers}(C)|_{s,r}\right] \geq Np^{r-s}\,, \quad r > s\,, \tag{5.2}$$

where $C|_s$ is the N-node partition induced on layer s by the multilayer partition C and $\text{Pers}(C)|_{s,r} = \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_r})$, *$r \in \{1, \ldots, |\mathcal{T}|\}$, $r > s$.*

Proposition 5.2.1 follows directly from Step 2 in Table 5.1. It states that community assignments in a given layer only depend on community assignments in the previous layer (i.e., the previous temporal snapshot) and on the null distribution $\mathbb{P}_0^s$. This respects the arrow of time. The weight importance of the previous layer versus the null distribution is determined by the value of p. When $p = 0$, community assignments in a given layer only depend on the null distribution of that layer [i.e., the second term on the right-hand side of (5.1)]. When $p = 1$, community assignments in a given layer only depend on community assignments of the previous layer (and consequently on community assignments in the first layer, by recursion). In the latter case, persistence between consecutive layers is maximal (i.e., $\text{Pers}(C)|_{s,s+1} = N$ for all s) and partitions induced on individual layers are all identical (i.e., $C|_s = C|_{s+1}$ for all s) by Proposition 4.2.3. The difference between $\mathbb{E}\left[\text{Pers}(C)|_{s,r}\right]$ and Np^{r-s} in (5.2) when $p \in [0, 1)$ depends on the null distribution.

Proposition 5.2.2. *For all $k \in \{1, \ldots, N|\mathcal{T}|\}$ and $s \in \{1, \ldots, |\mathcal{T}|\}$*

$$\begin{aligned}
\mathbb{P}\left[c_{i_1} = k\right] &= \mathbb{P}_0^1\left[c_{i_1} = k\right] \,, && (5.3)\\
\mathbb{P}\left[c_{i_s} = k\right] &= \mathbb{P}\left[c_{i_{s-1}} = k\right] p + \mathbb{P}_0^s\left[c_{i_s} = k\right](1-p) \,, \\
&= \mathbb{P}_0^1\left[c_{i_1} = k\right] p^{s-1} + (1-p)\sum_{r=2}^{s-1} \mathbb{P}_0^r\left[c_{i_r} = k\right] p^{s-r} \\
&\quad + (1-p)\mathbb{P}_0^s\left[c_{i_s} = k\right] \,, \quad s > 1 \,. && (5.4)
\end{aligned}$$

Proposition 5.2.2 follows from Step 1 and Step 2 in Table 5.1 and one can prove the third equality by induction on the number of layers. For $p = 1$, the second and third terms in equation (5.4) are 0 and $\mathbb{P}\left[c_{i_s} = k\right]$ only depends on the null distribution in layer 1. For $p = 0$, the first two terms in equation (5.4) are 0 and $\mathbb{P}\left[c_{i_s} = k\right]$ only depends on the null distribution in layer s.

We now comment on some features of Step 2 in Table 5.1 that govern how community assignments vary between consecutive layers. In Step 2a, a community assignment k in layer s can do one of 3 things: it can lose some of its nodes in layer $s+1$ (i.e., $\{i : c_{i_{s+1}} = k\} \subset \{i : c_{i_s} = k\}$); keep all of its nodes (i.e., $\{i : c_{i_{s+1}} = k\} = \{i : c_{i_s} = k\}$); or disappear (i.e., $\{i : c_{i_{s+1}} = k\} = \varnothing$ and $\{i : c_{i_s} = k\} \neq \varnothing$). The null distribution in Step 2b is responsible for a community assignment k gaining new nodes (i.e., $\{i : c_{i_{s+1}} = k\} \not\subset \{i : c_{i_s} = k\}$) or for a community label appearing (i.e., $\{i : c_{i_{s+1}} = k\} \neq \varnothing$ and $\{i : c_{i_s} = k\} = \varnothing$). One needs to bear this in mind when defining the null distributions $\mathbb{P}_0^s$. To illustrate how the null distribution and our model of persistence can interact, we give the conditional probability that a label disappears in layer $s+1$ and the conditional probability that a label appears in layer $s+1$ in Propositions 5.2.3 and 5.2.4, respectively.

Proposition 5.2.3. *For all $k \in \{1, \ldots, N|\mathcal{T}|\}$ and $s \in \{1, \ldots, |\mathcal{T}| - 1\}$*

$$\begin{aligned}
&\mathbb{P}\left[\{i : c_{i_{s+1}} = k\} = \varnothing \,\middle|\, C|_s\right] \\
= \;&\left[(1-p)\left(1 - \mathbb{P}_0^{s+1}[c_{i_{s+1}} = k]\right)\right]^{\sum_{i=1}^N \delta(c_{i_s}, k)} \times \left[p + (1-p)\left(1 - \mathbb{P}_0^{s+1}[c_{i_{s+1}} = k]\right)\right]^{N - \sum_{i=1}^N \delta(c_{i_s}, k)} \,.
\end{aligned}$$

The first term on the right-hand side of the equation in Proposition 5.2.3

$$\left[(1-p)\left(1 - \mathbb{P}_0^{s+1}[c_{i_{s+1}} = k]\right)\right]^{\sum_{i=1}^N \delta(c_{i_s}, k)} \,,$$

is the probability that all of the nodes in layer s with community label k change communities in Step 2a and are not assigned to label k in Step 2b (we use our node independence assumption). The second term in Proposition 5.2.3,

$$\left[p + (1-p)\left(1 - \mathbb{P}_0^{s+1}[c_{i_{s+1}} = k]\right)\right]^{N - \sum_{i=1}^N \delta(c_{i_s}, k)} \,,$$

is the probability that none of the nodes with community label k' in layer s (where $k' \neq k$) acquire the label k in layer $s+1$. The expression between brackets has two parts: either a node keeps the same community label or a node changes community labels and is not reassigned to label k. When $\mathbb{P}_0^{s+1}[c_{i_{s+1}} = k] = 0$ (i.e., the probability of being assigned to label k is zero using the null distribution of layer $s+1$), the expression in (5.2.3) only depends on our model of persistence and simplifies to $(1-p)^{\sum_{i=1}^{N} \delta(c_{i_s},k)}$. Furthermore, as $\mathbb{P}_0^{s+1}[c_{i_{s+1}} = k]$ increases, the probability that a label disappears decreases.

Proposition 5.2.4. *For all* $k \in \{1,\dots,N|\mathcal{T}|\}$ *and* $s \in \{1,\dots,|\mathcal{T}|-1\}$

$$\mathbb{P}\Big[\ \{i : c_{i_{s+1}} = k\} \neq \varnothing \Big|\ \{i : c_{i_s} = k\} = \varnothing\Big] = 1 - \left[p + (1-p)\Big(\sum_{C_j|_s \in C|_s} \mathbb{P}_0^{s+1}[c_{i_{s+1}} = j]\Big)\right]^N .$$

Proposition 5.2.4 gives the probability that at least one node in layer $s+1$ has the label k, given that no node in layer s has the label k. The term in square brackets on the right is the probability that none of the nodes in layer $s+1$ have the label k (conditional on k being absent in layer s). The term inside square brackets has two parts corresponding to the events either that a node keeps its label or that a node changes its label and is assigned to a label that already exists in layer s. When $\sum_{C_j|_s \in C|_s} \mathbb{P}_0^{s+1}[c_{i_{s+1}} = j] = 0$ (i.e., the community labels in layer s and $s+1$ do not overlap), the probability that a community label appears only depends on our model of persistence and is given by $1 - p^N$. Furthermore, higher values of $\sum_{C_j|_s \in C|_s} \mathbb{P}_0^{s+1}[c_{i_{s+1}} = j]$ decrease the probability that a label appears in layer $s+1$.

5.2.1.2 Multinomial null distribution

Suppose the null distribution $\mathbb{P}_0^s$ for each layer is independent of any sampled benchmark multilayer partition. That is, before performing Step 1, Step 2, and Step 3 in Table 5.1 to generate a sample multilayer benchmark partition $C \in \mathcal{C}_{\text{bench}}(\mathbb{P}_0^s; p)$, we fix $|\mathcal{T}|$ vectors (one for each layer) with non-negative entries

$$\{\boldsymbol{p}^s,\ s = 1\dots|\mathcal{T}|\}\,,$$

such that $\mathbf{1}^T\boldsymbol{p}^s = 1$ for all $s \in \{1,\dots,|\mathcal{T}|\}$. Every vector $\boldsymbol{p}^s$ defines the probabilities of a multinomial null distribution that assigns nodes in layer s to $|\boldsymbol{p}^s|$ communities. When $p = 0$, the expected community sizes in each layer are those of the multinomial null distribution in that layer. The support $\boldsymbol{c}^s = \{c_1^s,\dots,c_{|\boldsymbol{p}^s|}^s\}$ of the multinomial distribution of layer s corresponds to community labels and an unassigned node is assigned to the community with label c_k^s with probability

$$\mathbb{P}_0^s\left[c_{i_s} = c_k^s\right] = \begin{cases} p_k^s & \text{if } k \leq |\boldsymbol{p}^s|\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

To gauge the effect of the choices $\{\boldsymbol{p}^s,\ s = 1, \dots, |\mathcal{T}|\}$ and $\{\boldsymbol{c}^s,\ s = 1, \dots, |\mathcal{T}|\}$ on $C \in \mathcal{C}_{\text{bench}}(\mathbb{P}_0^s;\, p)$, we consider a few special cases.

Suppose that one takes these parameters to be the same across all layers (i.e., $\boldsymbol{p}^s = \boldsymbol{p}$ and $\boldsymbol{c}^s = \boldsymbol{c}$ for all s). The expected number of community labels per layer is then the same and the expected number of nodes per layer with community label c_k is also the same and is given by

$$\begin{aligned} \mathbb{E}\big[\ |\{i : c_{i_1} = c_k\}|\ \big] &= Np_k\,, \\ \mathbb{E}\big[\,|\{i : c_{i_s} = c_k\}|\,\big] &= N\mathbb{P}\left[c_{i_s} = c_k\right] \\ &= Np_k\,, \end{aligned}$$

where the third equality follows on using the expression for $\mathbb{P}\left[c_{i_s} = c_k\right]$ in Eq. (5.4) and the assumption that $\boldsymbol{c}^s = \boldsymbol{c}$ for all s. At each iteration of Step 2, one expects Npp_k nodes with label k in layer s to keep the label k layer $s+1$ and $N(1-p)p_k$ nodes with label k in layer s to potentially change labels in layer $s+1$. The additional constraint $\boldsymbol{c}^s = \boldsymbol{c}$ produces a temporal network in which nodes change community labels across layers such that the expected number of communities and the expected size of communities both remain constant.

Now suppose that one takes the parameter $\boldsymbol{p}^s$ to be the same across all layers (i.e., $\boldsymbol{p}^s = \boldsymbol{p}$ for all s) but takes the community assignments $\boldsymbol{c}^s$ to be disjoint between all pairs of layer (i.e., $\boldsymbol{c}^s \cap \boldsymbol{c}^r = \varnothing$ for all $s \neq r$). At each iteration of Step 2, a community label can only lose members;[3] and with probability $1 - p^N$, at least one new label will appear in every subsequent layer. In this case, one expects Npp_k members of community k in layer s to remain in community k in layer $s+1$ and $N(1-p)p_k$ members of community k in layer s to be assigned to new communities (because labels are non-overlapping) in layer $s+1$. This choice produces multilayer partitions in which the expected number of new community labels per layer is nonzero (unless $p = 1$) and the expected size of a given community decreases in time.

Finally, instead of considering the same vector across all layers, suppose that we generate $|\mathcal{T}|$ vectors $\{\boldsymbol{p}^s, s = 1, \dots, |\mathcal{T}|\}$ independently. The difference in vector

[3] Overlap between $\boldsymbol{c}^s$ and $\boldsymbol{c}^{s+1}$ (i.e., $\boldsymbol{c}^{s+1} \cap \boldsymbol{c}^s \neq \varnothing$) is a necessary condition for communities in layer s to gain new members in layer $s+1$.

lengths and vector entries depends on how they are generated (for example if we generate them by sampling each entry successively from a given distribution conditional on the entries summing to 1, these differences will depend on the chosen distribution and its parameters). As with community labels, different choices can have different biases. For example, if one fixes a sequential ordering (i.e., $c_k^s = k$ for all s), labels 1 to $\min_s |\boldsymbol{c}^s|$ have a nonzero probability of appearing in all layers. However, if one chooses completely non-overlapping community labels, then a community label cannot gain new nodes.

The point of the above discussion is to illustrate that the choices $\{\boldsymbol{p}^s, s = 1, \ldots, |\mathcal{T}|\}$ and $\{\boldsymbol{c}^s, s = 1, \ldots, |\mathcal{T}|\}$ can have a rather big effect on $\mathcal{C}_{\text{bench}}(\mathbb{P}_0^s; p)$. The issue of choosing one choice over another is in general nontrivial (and possibly depends on the question one is trying to answer). The focus of the present chapter is to carry out a preliminary comparison between multilayer community-detection methods and between different heuristics for a given method. Accordingly, we consider two choices for $\boldsymbol{p}^s$:

1. $\boldsymbol{p}^s = \boldsymbol{p}$ and $p_i = p_j$ for all $s = 1, \ldots, |\mathcal{T}|$ and $i, j \in \{1, \ldots, |\boldsymbol{p}|\}$;
2. $\boldsymbol{p}^s$ are independent samples drawn from a specified distribution.

In the first case, the expected size of a community is the same across layers and across communities in a given layer (the authors of [72] adopt this choice). In the second case, the expected size of a community can vary across layers and across communities of a given layer. We choose the community assignments $\{\boldsymbol{c}^s, s = 1, \ldots, |\mathcal{T}|\}$ of each layer uniformly at random (without replacement) for both cases from the set $\{1, \ldots, \max_s |\boldsymbol{c}^s|\}$ of integers. This reduces to using the same labels in all layers for our first choice of $\boldsymbol{p}^s$. We include a parameter $\alpha \in \mathbb{N}$ in our implementation that a user can specify to increase the size of this set and select community assignments uniformly at random from $\{1, \ldots, \alpha \max_s |\boldsymbol{c}^s|\}$. For our second choice of $\boldsymbol{p}^s$, we sample the entries of $\boldsymbol{c}^s$ from a power-law distribution with exponent τ (independent of s), and we place a lower bound $|C|_{\min}$ and an upper bound $|C|_{\max}$ on the expected community size. We choose a power-law distribution because it is common practice in the literature (e.g., [117, 120]) but stress that one could substitute this choice with any other distribution. We show pseudo-code for generating a vector of expected community sizes in Algorithm 1.

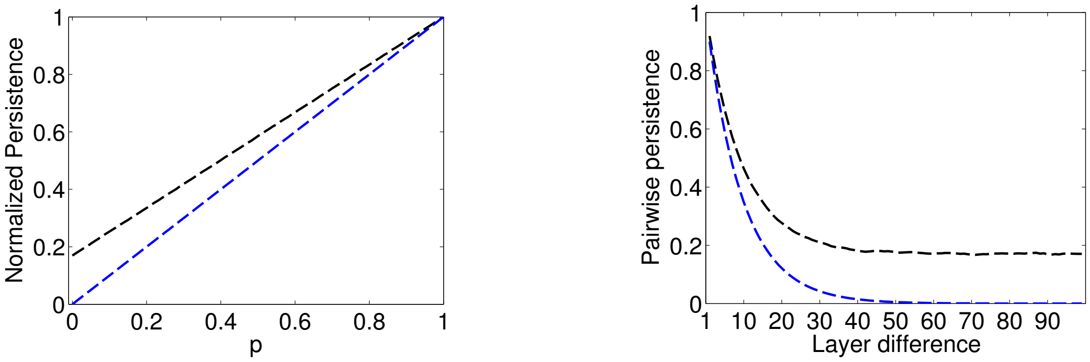
Algorithm 1 Sample expected community sizes from power-law distribution with cutoffs $c_{\min}$ and $c_{\max}$.

```
function SampleCommunitySizes($\tau, c_{\min}, c_{\max}, N$)
    $r = N$
    $i = 1$
    while $r > 0$ do
        $c_i =$Power$(\tau, c_{\min}, c_{\max})$                ▷ sample from power-law
        if $r - c_i < c_{\min}$ then        ▷ stop sampling if remainder is too small
            if $r < c_{\max}$ then
                $c_i = r$
            else if $r - c_{\min} < c_{\min}$ then
                $c_i = r$                                ▷ constraints not satisfied
            else
                $c_i =$Power$(\tau, c_{\min}, \min(c_{\max}, r - c_{\min}))$
            end if
        end if
        $r = r - c_i$
        $i = i + 1$
    end while
    $\boldsymbol{c} =$RandPerm$(\boldsymbol{c})$        ▷ permute community sizes uniformly at random
    $n_c = i - 1$
    return $\boldsymbol{c}, n_c$
end function
```

5.2.1.3 Examples

We end this section with a few examples in order to visualize some features of the benchmark multilayer partition. We show a plot of the value of persistence in Fig. 5.2(a) for a multilayer partition with $(|\mathcal{T}|, N) = (100, 100)$ and $p \in \{0, 0.01, \dots, 0.99, 1\}$. We plot the mean value of persistence over 100 sample multilayer partitions for each value of p. The difference between the dotted black curve (persistence of benchmark multilayer partition averaged over 100 samples) and the dotted blue curve $(pN|\mathcal{T}|)$ in Fig. 5.2(a) represents the contribution of the null distribution (and its related parameter choices) to the value of persistence. If one chooses non-overlapping community labels, then the black curve would be identical to the blue curve in both Fig. 5.2(a) and Fig. 5.2(b). Note additionally that pairwise persistence drops below 0.4 near $s - r = 10$ for the blue curve and near $s - r = 15$ for the black curve. In practice, this implies that although consecutive layers will be similar for $p = 0.9$, those that are ten layers apart will be rather dissimilar.

In Fig. 5.3, we show example partitions generated with our benchmark for a multilayer network with ten layers (i.e., $|\mathcal{T}| = 10$) and one hundred nodes in each



(a) Normalized persistence

(b) Normalized pairwise persistence

Figure 5.2: The sample of p values is the set $\{0, 0.1, \dots, 0.99, 1\}$ with a discretization step of 0.01 between each pair of consecutive values. The null distribution for each layer is a power-law distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. For each value of p, we generate 100 sample benchmark partitions with $(N, |\mathcal{T}|) = (100, 100)$. (a) The value of normalized persistence $\mathrm{Pers}(C)/[N(|\mathcal{T}| - 1)]$ averaged over 100 sample benchmark partitions (black dashed line) and the expected value of persistence $pN(|\mathcal{T}| - 1)$ not accounted for by the null distribution (blue dashed line). (b) The value of normalized pairwise persistence for $p = 0.9$ averaged over 100 sample benchmark partitions and all pairs of layers s, r with the same layer difference $s - r$ (black dashed curve) and the expected value of pairwise persistence $Np^{s-r}(s > r)$ not accounted for by the null distribution (blue dashed curve).

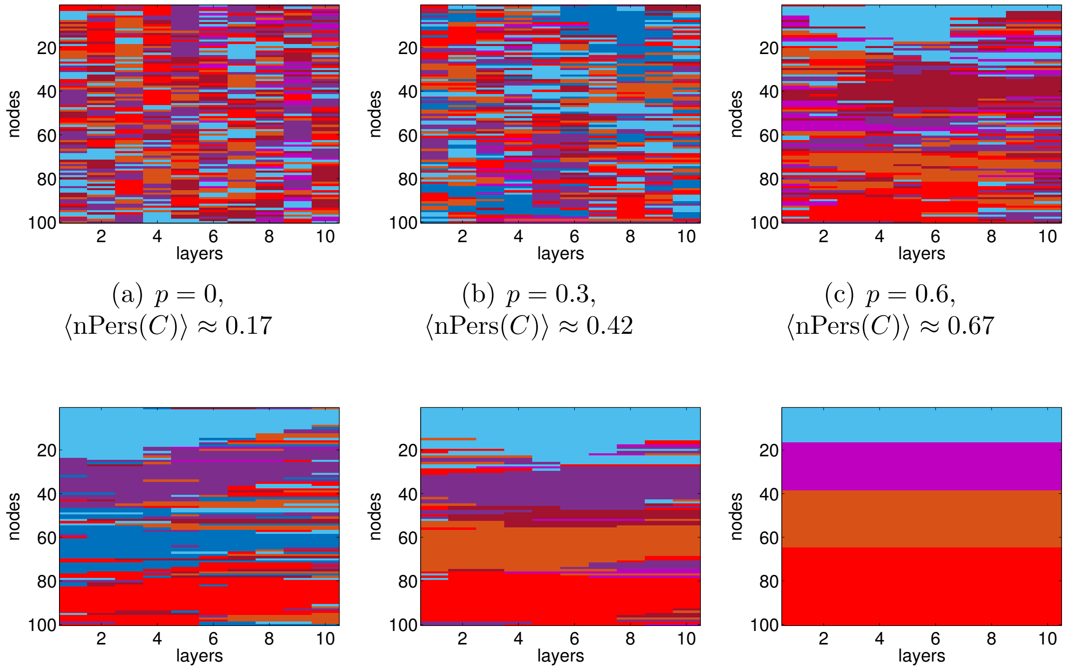
layer (i.e., $N = 100$). We show sample partitions for $p \in \{0, 0.3, 0.6, 0.8, 0.9, 1\}$ and indicate the value of normalized persistence $\mathrm{Pers}(C)/[N(|\mathcal{T}| - 1)]$ for each partition. As one would anticipate, multilayer partitions generated with a larger value of p have a larger value of persistence.

5.2.2 Benchmark multilayer network

There are various models for generating networks that “fit” a single-layer partition [49, 88, 117, 120]. For the purposes of the numerical experiments in this chapter, we use two simple models.

5.2.2.1 Unweighted stochastic block model

Every multilayer partition C that we generate induces $|\mathcal{T}|$ single-layer partitions $\{C|_1, \dots, C|_{|\mathcal{T}|}\}$. The main model we use to generate a multilayer network that fits a multilayer partition is an unweighted stochastic block model on intra-layer edges [88]. For every layer s, we generate a network such that every pair of nodes in the same



(a) $p = 0$, $\langle \mathrm{nPers}(C) \rangle \approx 0.17$

(b) $p = 0.3$, $\langle \mathrm{nPers}(C) \rangle \approx 0.42$

(c) $p = 0.6$, $\langle \mathrm{nPers}(C) \rangle \approx 0.67$

(d) $p = 0.8$, $\langle \mathrm{nPers}(C) \rangle \approx 0.83$

(e) $p = 0.9$, $\langle \mathrm{nPers}(C) \rangle \approx 0.92$

(f) $p = 1$, $\langle \mathrm{nPers}(C) \rangle = 1$

Figure 5.3: We fix $(N, \mathcal{T}) = (100, 10)$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We show example output benchmark partitions for (a) $p = 0$, (b) $p = 0.3$, (c) $p = 0.6$, (d) $p = 0.8$, (e) $p = 0.9$, and (f) $p = 1$ and the mean normalized persistence ($\langle \mathrm{nPers}(C) \rangle$) computed over 100 sample multilayer partitions with the same parameter values (rounded the nearest second decimal). Standard deviation for all samples is at least on the order of 10^{-2} and decreases as p increases (until it reaches 0 when $p = 1$). Colors represent community assignments. For each multilayer partition, we choose a node ordering that emphasizes (whenever possible) persistent community structure in that multilayer partition.

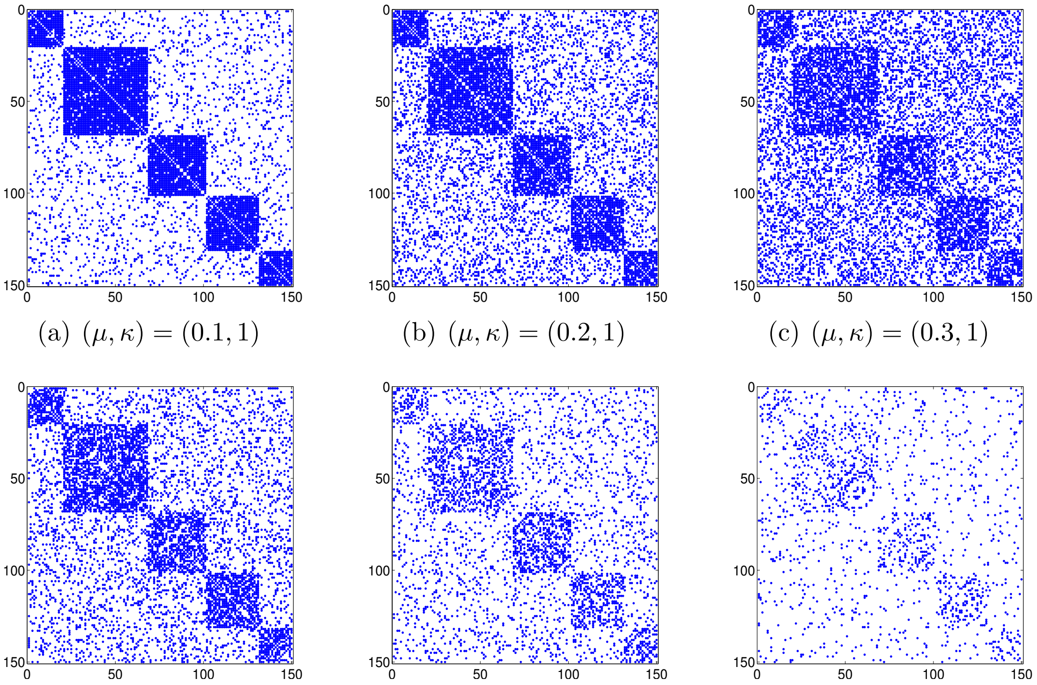
.

community in the partition $C|_s$ is connected with probability $1 - \mu$ and every pair of nodes in different communities is connected with probability $\mu \in [0, 1]$ [88]. That is,

$$\mathbb{P}\left[A_{ijs} = 1\right] = \begin{cases} 1 - \mu & \text{if } \delta(c_{i_s}, c_{j_s}) = 1\,, \\ \mu & \text{otherwise}\,, \end{cases} \tag{5.5}$$

where $\boldsymbol{A}_s \in \{0, 1\}^{N \times N}$ is the adjacency matrix of layer s and A_{ijs} is the $(i, j)^{\text{th}}$ entry of $\boldsymbol{A}_s$. Suppose that $C|_s$ has K communities $C_1|_s, \ldots, C_K|_s$ and denote the size $\left|C_k|_s\right|$ of the k^{th} community by α_k for ease of writing. The expected number of edges within a community of size α_k using this model is

$$\mathbb{E}\big[|\{A_{ij} = 1 | i \in C_k, j \in C_l\}|\big] = \begin{cases} (1 - \mu)\alpha_k(\alpha_k - 1) & \text{if } C_k = C_l\,, \\ \mu \alpha_k \alpha_l & \text{otherwise}\,\,. \end{cases}$$



(a) $(\mu, \kappa) = (0.1, 1)$ (b) $(\mu, \kappa) = (0.2, 1)$ (c) $(\mu, \kappa) = (0.3, 1)$

(d) $(\mu, \kappa) = (0.2, 0.8)$ (e) $(\mu, \kappa) = (0.2, 0.5)$ (f) $(\mu, \kappa) = (0.2, 0.2)$

Figure 5.4: We fix $N = 150$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\text{min}} = 0.1N$, and the upper bound on the expected community size is $|C|_{\text{max}} = 0.3N$. We show example adjacency matrices for (a) $(\mu, \kappa) = (0.1, 1)$, (b) $(\mu, \kappa) = (0.2, 1)$, (c) $(\mu, \kappa) = (0.3, 1)$, (d) $(\mu, \kappa) = (0.2, 0.8)$, (e) $(\mu, \kappa) = (0.2, 0.5)$, and (f) $(\mu, \kappa) = (0.2, 0.2)$. Nodes are ordered by community assignment in all panels and we use the same planted partition for all adjacency matrices. Each panel shows the sparsity pattern of each adjacency matrix: entries equal to 1 are shown in blue and entries equal to 0 are shown in white (one can generate these figures with the `spy` command in MATLAB).

We also include a parameter κ to control the sparsity of edges in a network given a choice μ:

$$\mathbb{P}\left[A_{ijs} = 1\right] = \begin{cases} \kappa(1-\mu) & \text{if } \delta(c_{i_s}, c_{j_s}) = 1\,, \\ \kappa\mu & \text{otherwise}\,. \end{cases} \tag{5.6}$$

The model in (5.5) corresponds to the choice $\kappa = 1$ in (5.6). For a given choice of (μ, κ), the expected edge density within a community is always greater or equal to the expected edge density between communities when $\mu \in [0, 0.5]$. For a fixed choice of μ, the expected number of edges in the network is larger for a larger choice of κ. We show example adjacency matrices for different choices of (μ, κ) in Fig. 5.4.

5.2.2.2 Weighted block model

We also use weighted block models similar to the toy examples that we used in Section 4.1. That is,

$$A_{ijs} = \begin{cases} a & \text{if } \delta(c_{i_s}, c_{j_s}) = 1\,, \\ b & \text{otherwise}\,, \end{cases} \tag{5.7}$$

where $a > b \geq 0$. For example, in a partition with three communities of sizes three, four, and two, the adjacency matrix of layer s (with nodes ordered by community assignments) is given by

$$\boldsymbol{A}_s = \left[\begin{array}{ccc|cccc|cc} a & a & a & b & b & b & b & b & b \\ a & a & a & b & b & b & b & b & b \\ a & a & a & b & b & b & b & b & b \\ \hline b & b & b & a & a & a & a & b & b \\ b & b & b & a & a & a & a & b & b \\ b & b & b & a & a & a & a & b & b \\ b & b & b & a & a & a & a & b & b \\ \hline b & b & b & b & b & b & b & a & a \\ b & b & b & b & b & b & b & a & a \end{array}\right].$$

A key difference between this single-layer network model and the previous one is that *all* inter-community edges have a lower edge weight than intra-community edge weights (this is only guaranteed in the previous model if $\mu = 0$). This can make planted partitions more easily detectable by Louvain-like heuristics when using the model in (5.7).[4]

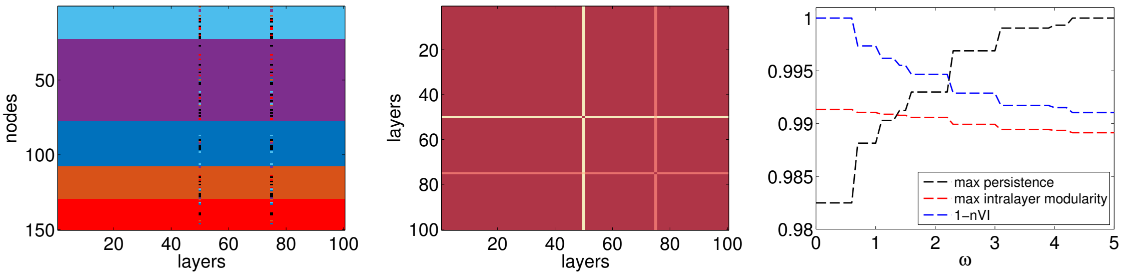
5.2.2.3 Examples

Before performing numerical experiments to compare the performance of different methods and heuristics, we give four toy examples to illustrate using our benchmark the potential use of the parameter ω in practical contexts.

Changes in connectivity patterns

We first illustrate how varying the parameter ω can help differentiate between changes in community structure. We generate a multilayer network that is similar to the toy example considered in Section 4.2. Consider a multilayer network where $(|\mathcal{T}|, N) =$

[4] For examples, suppose that one uses multilayer modularity maximization with a uniform null network and $\langle \boldsymbol{A}_s \rangle > b$ for all s. In the non-stochastic block model (5.7), all inter-community entries in the single-layer modularity matrices are negative. It follows that for small enough ω (e.g., $\omega = 0$), merging pairs of nodes in the same layer but in different communities will not be feasible moves in the first iteration of phase 1 of Louvain as they cannot increase the quality function. This does not hold for the model in (5.6).



(a) Planted multilayer partition (b) Persistence matrix (c) Multilayer quality function and 1−nVI

Figure 5.5: We fix $(N, |\mathcal{T}|) = (150, 100)$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate a multilayer partition with $p = 1$ and perturb the partition induced on layer 50 and 75 with $p = 0.5$. (We use the same perturbed partition for both layers.) We generate a multilayer network using the non-stochastic block model in (5.7) and set $(a, b) = (0.3, 0)$ in layer 50, $(a, b) = (0.1, 0)$ in layer 75, and $(a, b) = (1, 0)$ in all remaining layers. We use multilayer modularity maximization with the uniform null network, $\gamma = 1$, and LouvainRand to generate output partitions. The sample of ω value is $\{0, 0.1, \ldots, 4.9, 5\}$ with a discretization step of 0.1 between consecutive values (giving 51 distinct values). We perform one run for each value of ω. We post-process each output partition and iterate LouvainRand on the post-processed partition as an initial partition until it converges. That is, we iterate with the post-processed output as the starting partition instead of an all singleton starting partition. We use the Hungarian algorithm for post-processing [113], which maximizes persistence without changing intra-layer partitions for a given multilayer partition (see Section 4.2.3.1). The maximum number of iterations until convergence is 4 for all values of ω. (a) Planted partition. (b) Pairwise persistence $|\mathcal{T}| \times |\mathcal{T}|$ matrix (with $(s, r)^{\text{th}}$ entry $\sum_{i,j=1}^{N} \delta(c_{i_s}, c_{i_r})$). Colors scale with the entries of the matrix from deep blue (entries close to 0.5) to deep red (entries close to 1). (c) Normalized persistence, normalized intra-layer modularity, and nVI (with respect to the planted partition) of the output partition with the largest value of multilayer modularity for each value of ω. (We plot 1−nVI to include all plots in the same figure.) We normalize intra-layer modularity by the sum of positive entries on the diagonal blocks of the multilayer adjacency matrix (i.e., $\sum_{i,j,s} B_{ijs}$ for $B_{ijs} > 0$). The largest difference in absolute value between the plots in (c) and the same plots computed for the identified output partition (as opposed to the one with largest value of multilayer modularity) is 10^{-4}.

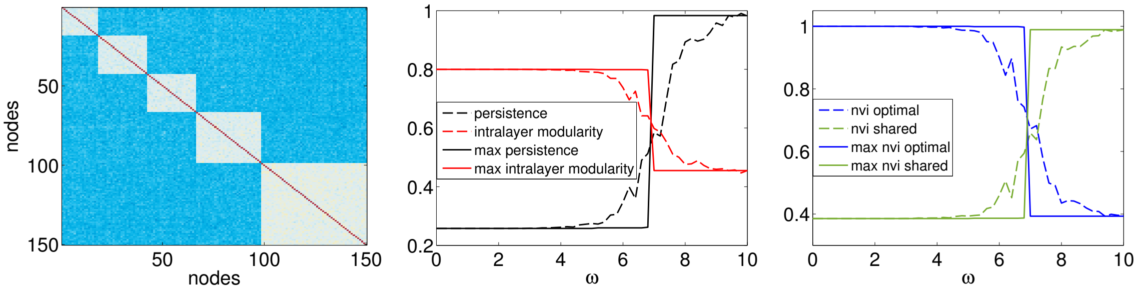
$(100, 150)$ and in which the partition $C|_s$ induced on each layer is the same in all but two layers. In these two layers, we perturb the single-layer partition by applying Step 2 in Table 5.1 to $C|_s$ with $p = 0.5$.[5] We set inter-community edge weights to zero in all layers (i.e., $b = 0$) and intra-community edge weights to one (i.e., $a = 1$)

[5] Note that one could generate this multilayer partition directly from our benchmark by introducing into the parameter p a dependence on each layer and longer memory (i.e., by making the multilayer generation process non-Markovian). For example, a node in layer s, $s > 2$, has a probability p_{s1} of keeping its label from layer $s - 1$ and p_{s2} of keeping its label from layer $s - 2$.

in all but two layers. In particular, we set $a = 0.3$ in layer 50 and $a = 0.1$ in layer 75. We choose these values so that $b < \langle \boldsymbol{A}_s \rangle$ in all layers and so that the planted partition is harder to change (in terms of modularity cost) in layer 50 than in layer 75 (since $C|_{50} = C|_{75}$ and $0.1 < 0.3$). We show the planted partition in Fig. 5.5(a) and the matrix of pairwise persistence between all pairs of layers averaged over increasing values of ω in Fig. 5.5(b). (We describe in detail how we generate each panel in the figure caption of Fig. 5.5.) As anticipated, larger values of ω are required to change the planted partition in layer 50 in favor of persistence. In 5.5(c) we plot the persistence, intra-layer modularity, and normalized variation of information (nVI, see Section 2.4) with respect to the planted partition as a function of ω. For small enough values of ω, the planted partition is identified and for large enough values of ω, the output partition has optimal persistence. For intermediate values, the single-layer partition in layer 75 is first affected (by inspection, when $\omega \lesssim 2.6$) followed by the single-layer partition in layer 50 (by inspection, when $\omega \gtrsim 2.6$).

Shared connectivity patterns

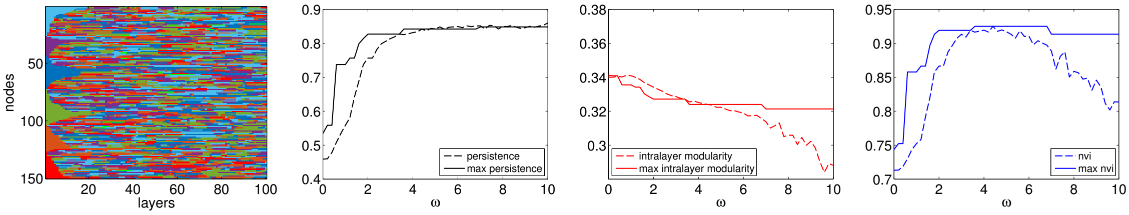
As a second simple experiment, we illustrate how varying the parameter ω can help detect block-diagonal structure that is present in all layers but is not optimal for any individual layer (see, e.g., the toy example in Fig. 4.5 of Section 4.2). We generate a first multilayer partition with $p = 0$ and a second with $p = 1$. The first will constitute the planted partition if one were to consider each layer individually and the second will constitute the planted partition which is not optimal for any individual layer but which is shared across layers. We anticipate that the first multilayer partition will be detected for small enough values of ω (intra-layer modularity is favored over persistence) and the second will be detected for large enough values of ω (persistence is favored over intra-layer modularity). For each layer we generate two adjacency matrices using the non-stochastic block model: one that fits the first partition ($\boldsymbol{A}_s^1$) and one that fits the second ($\boldsymbol{A}_s^2$). We set $(a, b) = (1, 0)$ in $\boldsymbol{A}_s^1$ for the first planted partition and $(a, b) = (0.5, 0)$ in $\boldsymbol{A}_s^2$ for the second planted partition (note that $\boldsymbol{A}_s^2 = \boldsymbol{A}_r^2$ for all s, r because $p = 1$ in the second planted partition). We then sum the adjacency matrices of each layer to obtain a proxy of a multilayer network with a sub-optimal but shared planted partition (i.e., $\boldsymbol{A}_s^3 = \boldsymbol{A}_s^1 + \boldsymbol{A}_s^2$). This is the multilayer network we use in this experiment. Figure 5.6(a) shows the association matrix computed over the set of partition induced on individual layers for increasing values of ω (nodes are ordered according to their community assignments in the shared planted partition). The association matrix clearly reflects the planted shared structure. Figures 5.6(b,c)



(a) Association matrix (b) Multilayer quality function (c) 1−nVI

Figure 5.6: We fix $(N, \mathcal{T}) = (150, 100)$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\text{min}} = 0.1N$, and the upper bound on the expected community size is $|C|_{\text{max}} = 0.3N$. We generate a multilayer partition with $p = 0$ and a multilayer partition with $p = 1$. We generate two multilayer networks using the non-stochastic block model and set $(a, b) = (1, 0)$ for the first multilayer partition and $(a, b) = (0.5, 0)$ for the second. We perform all numerical experiments on a multilayer network with adjacency matrix for each layer given by the sum of the corresponding adjacency matrices for each of the two planted partitions. We use multilayer modularity maximization with the uniform null network, $\gamma = 1$, and LouvainRand to generate output partitions. The sample of ω values is $\{0, 0.2, \ldots, 9.8, 10\}$ with a discretization step of 0.2 between consecutive values (giving 51 distinct values). We perform one run for each value of ω. We post-process each output partition and iterate LouvainRand on the post-processed partition until it converges. The mean number of iterations across ω values is 2 and the maximum number of iterations is 5. (a) Association matrix indicating the co-classification index of nodes averaged over the set of partitions induced on each layer and over ω values in $\{0, 0.2, \ldots, 0.98, 1\}$. Colors scale with the entries of the matrix from dark blue (entries close to 0) to deep red (entries close to 1). (b) Normalized persistence and intra-layer modularity for each value of ω of the identified output partition (dashed black and dashed red curves, respectively) and of the output partition with the largest value of multilayer modularity (solid black and solid red curves, respectively). (c) 1−nVI with respect to the optimal and shared planted partitions for each value of ω of the identified output partition (dashed blue and dashed green curves, respectively) and of the output partition with the largest value of multilayer modularity (solid line blue and solid line green curves, respectively). We normalize intra-layer modularity by the sum of positive entries on the diagonal blocks of the multilayer adjacency matrix (i.e., $\sum_{i,j,s} B_{ijs}$ for $B_{ijs} > 0$).

show plots of persistence, intra-layer modularity, and nVI with respect to each of the two planted multilayer partition. Figures 5.6(b,c) suggest the existence of two regimes: one in which the first planted partition yields a larger value of multilayer modularity and one in which the second (shared) planted partition yields a larger value of multilayer modularity.

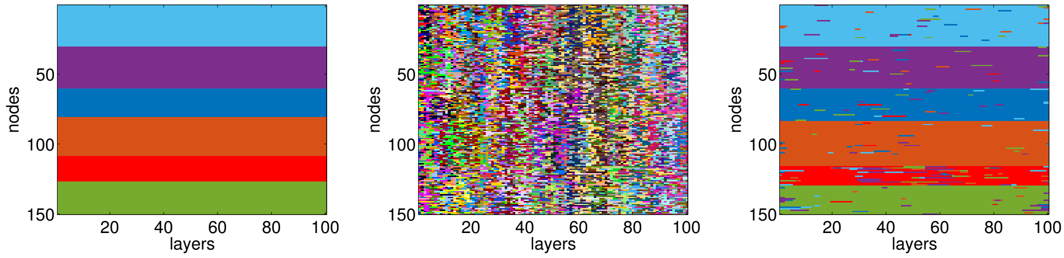


(a) Planted partition (b) Persistence (c) Intra-layer modularity (d) 1−nVI

Figure 5.7: We fix $(N, |\mathcal{T}|) = (150, 100)$ and a power-law null distribution with exponent $\tau = -1$, the lower bound one the expected community size is $|C|_{\text{min}} = 0.1N$, and the upper bound on the expected community size is $|C|_{\text{max}} = 0.3N$. We generate a multilayer partition with $p = 0.8$. We generate a multilayer network using a stochastic block model. We choose 50 layers uniformly at random and set $(\mu, \kappa) = (0.4, 1)$ in these layers and $(\mu, \kappa) = (0.1, 1)$ in the remaining layers. We use multilayer modularity maximization with the uniform null network, $\gamma = 1$, and LouvainRand to generate output partitions. The sample of ω values is $\{0, 0.2, \ldots, 9.8, 10\}$ with a discretization step of 0.1 between consecutive values (giving 51 distinct values). We perform one run for each value of ω. We post-process each output partition and iterate LouvainRand on the post-processed partition until it converges. The maximum number of iterations until convergence is 5. (a) Planted partition. (b,c) Normalized persistence and intra-layer modularity for each value of ω of the identified output partition (dashed black and dashed red curves, respectively) and of the output partition with the largest value of multilayer modularity (solid black and solid red curves, respectively). (d) 1−nVI with respect to the planted partitions for each value of ω of the identified output partition (dashed blue curve) and of the output partition with the largest value of multilayer modularity (solid blue curve). We normalize intra-layer modularity by the sum of positive entries on the diagonal blocks of the multilayer adjacency matrix (i.e., $\sum_{i,j,s} B_{ijs}$ for $B_{ijs} > 0$).

Noisy connectivity patterns

We generate a multilayer network with $p = 0.8$ and $(N, |\mathcal{T}|) = (150, 100)$, and we generate a multilayer network that fits the planted partition using the stochastic block model in equation (5.6). We choose 50 layers uniformly at random and set $(\mu, \kappa) = (0.4, 1)$ in these layers and $(\mu, \kappa) = (0.1, 1)$ in the remaining layers. We choose these values such that the planted partition cannot be recovered by LouvainRand when performed on the noisy layers individually but can be recovered for the remaining layers. We show the planted multilayer partition in Fig. 5.7(a). We order nodes according to community assignments in the first layer. As anticipated from Fig. 5.2, when $p = 0.8$ the similarity between single-layer planted partitions with respect to the first layer's planted partition decays quickly. We show plots of persistence, intra-layer modularity, and 1−nVI with respect to the planted partition in Fig. 5.7(b,c,d). These plots indicate that one can recover a partition that is close to the planted partition



(a) Planted partition

(b) Output multilayer partition with $\omega = 0.1$

(c) Output multilayer partition with $\omega = 1.6$

Figure 5.8: We fix $(N, |\mathcal{T}|) = (150, 100)$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate a multilayer partition with $p = 1$ and a multilayer network using the stochastic block model in (5.6) with $(\mu, \kappa) = (0.1, 0.05)$. We use multilayer modularity maximization with the uniform null network, $\gamma = 1$, and LouvainRand to generate output partitions for $\omega = 0.1$ and $\omega = 1.6$. We perform one run for each value of ω. We post-process each output partition and iterate LouvainRand on the post-processed partition until it converges. The maximum number of iterations for convergence is 5. (a) Planted partition. (b) Output partition for $\omega = 0.1$. (c) Output partition for $\omega = 1.6$.

by increasing ω. In particular, $1 - nVI \approx 0.93$ when $4 \lesssim \omega \lesssim 5$). We do not show the recovered partition because as one would expect from Fig. 5.7(a), it is difficult to tell visually that they are similar. We comment on the increasing gap between the dotted and solid curves in Fig. 5.7(c,d) in the last section of this chapter.

Sparse connectivity patterns

As a last introductory experiment, we consider a case in which the probability of an edge within a community is much higher than the probability of an edge between communities, but each single-layer network is sparse and planted structure cannot be recovered by considering each layer individually. We generate a multilayer partition with $(N, |\mathcal{T}|) = (150, 100)$ and $p = 1$; and a multilayer network that fits the planted partition using the stochastic block model in equation (5.6) with parameters $(\mu, \kappa) = (1, 0.05)$. We perform the same experiment as in the previous three examples for values of $\omega \in \{0, 0.1, \ldots, 4.9, 5\}$. We observe that for higher values of ω, the identified partition is more similar to the planted partition. We show example output partitions in Fig. 5.8(b) for $\omega = 0.1$ (where ω is still too low to recover the planted partition) and in Fig. 5.8(c) for $\omega = 1.8$ (where ω is high enough to recover a planted partition that is close to the planted partition). We note that the partition is not exactly the planted partition: the fourth (brown) community is slightly larger and the fifth (red) community is slightly smaller. However, the number of communities drops from 213 in Fig. 5.8(b) to 6 in Fig. 5.8(c).

5.3 Comparison of methods and heuristics

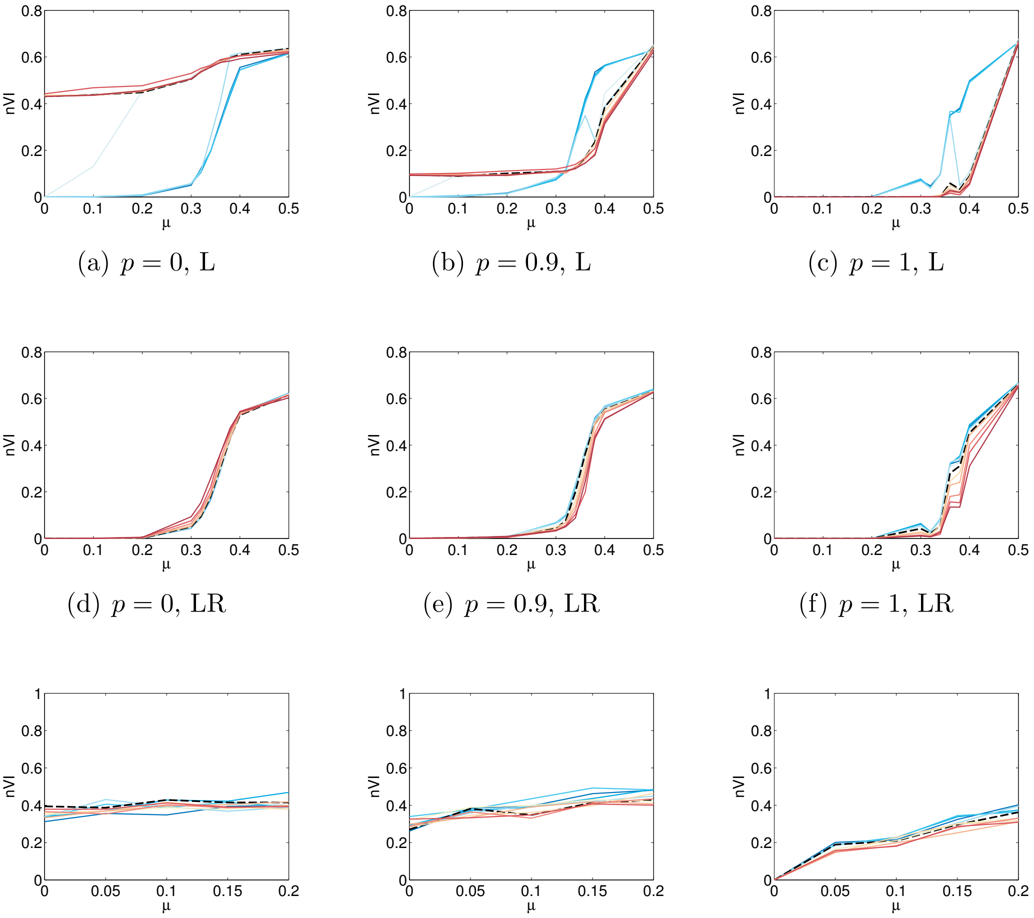
In this section we compare the performance of several heuristics for multilayer modularity maximization and the performance of two multilayer community-detection methods (multilayer modularity maximization and multilayer map minimization, see Section 2.2) on our benchmark. We begin with multilayer modularity maximization. We adopt the following parameter choices for all numerical experiments with multilayer modularity maximization. We fix $(N, \mathcal{T}) = (150, 100)$ and use the uniform null network with $\gamma = 1$. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \dots, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\text{min}} = 0.1N$, and the upper bound on the expected community size is $|C|_{\text{max}} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.05, \dots, 4.45, 0.5\}$ and $\kappa \in \{1, 0.2, 0.05\}$.

5.3.1 Variants of Louvain and spectral bipartitioning

5.3.1.1 Louvain, LouvainRand, and spectral bipartitioning

In our first experiment, we fix $\kappa = 1$ (i.e., the network is dense) and compare the performance of Louvain (L), LouvainRand (LR), and spectral bipartioning (SB) using only the leading eigenvector (see Section 2.3.3) in Fig. 5.9. The color of the curves scales with the value of ω: $\omega < 1$ is in blue, $\omega = 1$ is in black, and $\omega > 1$ is in red. For each pair (p, μ), we generate a multilayer partition and a multilayer network using the stochastic block model in (5.6). We then apply multilayer modularity maximization for $\omega \in \{0, 0.2, \dots, 1.8, 2\}$. Note that for a given pair (p, μ), we use the same partition and multilayer network for all values of ω (but we generate a different multilayer partition and multilayer network for each distinct pair). Although we perform this experiment for all values of $p \in \{0, 0.1, \dots, 0.9, 1\}$, we only include plots for $p \in \{0, 0.9, 1\}$ because these illustrate the three main qualitatively different cases.

We first comment on Fig. 5.9(a,b,c). A first striking feature is the separation between the red curves and the blue curves. This occurs because of the abrupt fall to 0 of the number of intra-layer merges during the first completion of phase 1 in Louvain when $\omega = \max_{i,j,s} B_{ijs}$ (see Section 4.2.3.1). We refer to this phenomenon



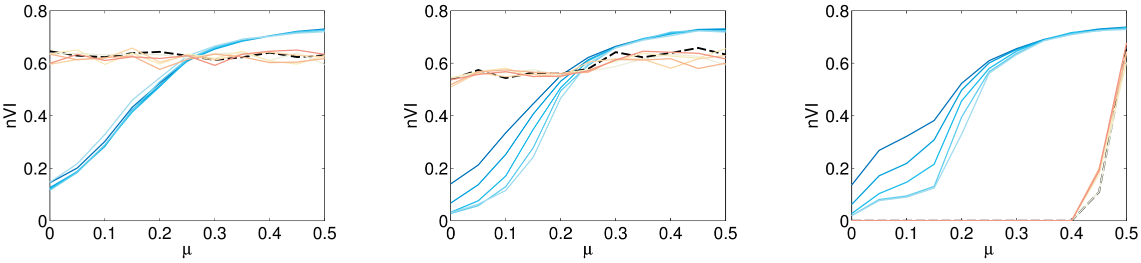
(a) $p = 0$, L
(b) $p = 0.9$, L
(c) $p = 1$, L

(d) $p = 0$, LR
(e) $p = 0.9$, LR
(f) $p = 1$, LR

(g) $p = 0$, SB
(h) $p = 0.9$, SB
(i) $p = 1$, SB

Figure 5.9: We fix $(N, \mathcal{T}) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. The sample of ω values is $\{0, 0.2, \ldots, 1.8, 2\}$. We show plots for $\omega < 1$ with solid blue curves, those for $\omega > 1$ with solid red curves, and that of $\omega = 1$ with a black dashed curve. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.05, \ldots, 4.45, 0.5\}$ and $\kappa = 1$. We compute the nVI (2.33) between the planted multilayer partition and the output multilayer partition for each triplet (p, μ, ω). (a,b,c) Louvain (L) and $p = 0, 0.9, 1$, respectively. (d,e,f) LouvainRand (LR) and $p = 0, 0.9, 1$, respectively. (g,h,i) Spectral bi-partitioning (SB) and $p = 0, 0.9, 1$, respectively. We use one run for each value of ω.

in this chapter as a "phase transition" of Louvain for ease of writing. The plots suggest that the value $\max_{i,j,s} B_{ijs}$ falls between 0.8 and 1 for all values of μ (which one can also check numerically by plotting $\max_{i,j,s} B_{ijs}$ for the different values of μ). A second observation is that the intercept of the red curves with the vertical axis gradually becomes lower as one increases p. This suggests that for a multilayer



(a) L & PP, $p = 0$, $\kappa = 0.2$ (b) L & PP, $p = 0.9$, $\kappa = 0.2$ (c) L & PP, $p = 1$, $\kappa = 0.2$

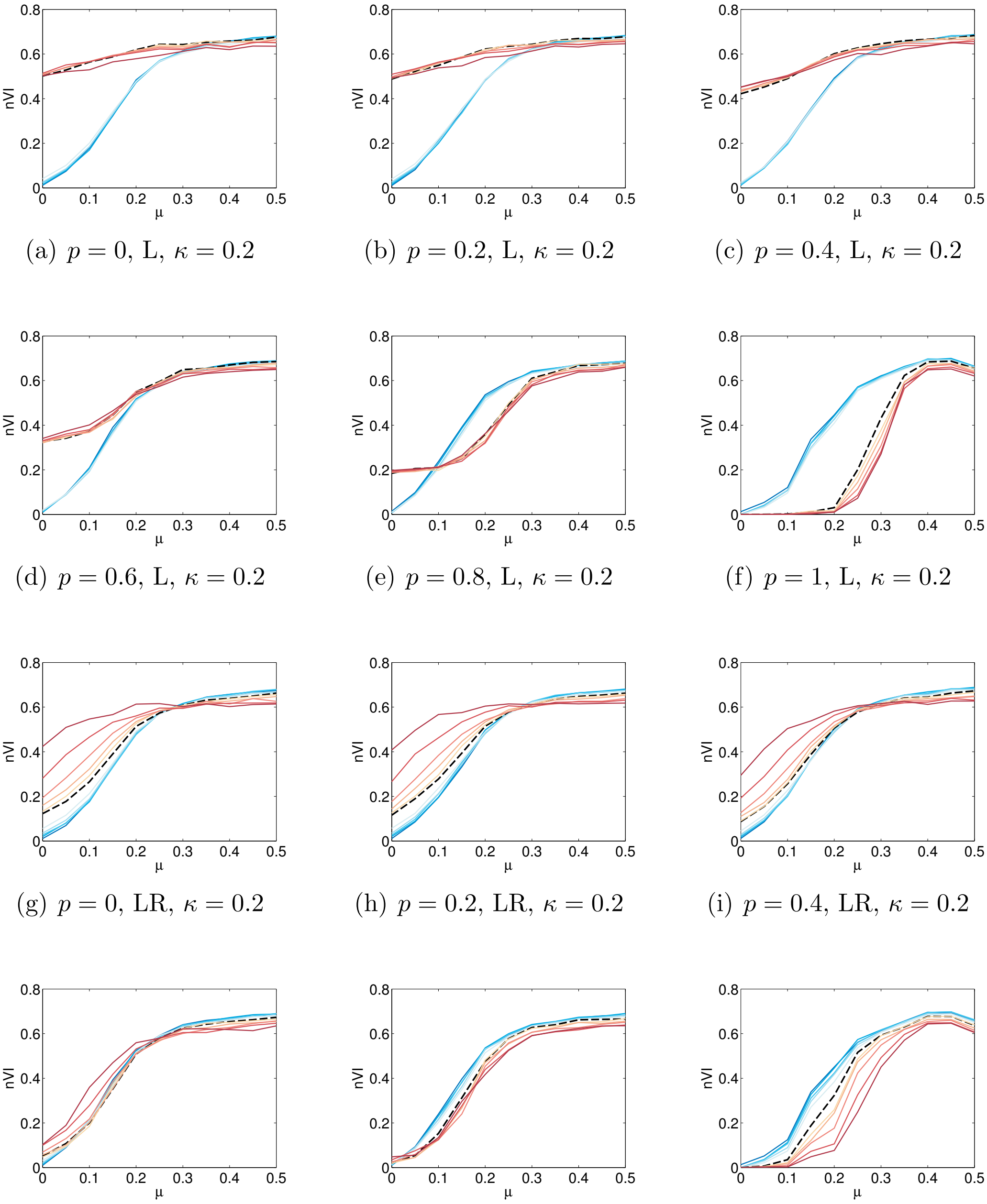
Figure 5.10: We fix $(N, \mathcal{T}) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. The sample of ω values is $\{0, 0.2, \ldots, 1.8, 2\}$. We show plots for $\omega < 1$ with solid blue curves, those for $\omega > 1$ with solid red curves, and that of $\omega = 1$ with a black dashed curve. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\text{min}} = 0.1N$, and the upper bound on the expected community size is $|C|_{\text{max}} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.05, \ldots, 4.45, 0.5\}$ and $\kappa = 0.2$. We post-process (PP) multilayer partitions after each pass of phase 1 during its first completion (post-processing in this way can alter partitions induced on individual layers in the final output partition). (a,b,c) Louvain (L) and $p = 0, 0.9, 1$, respectively.

network in which the planted partition has a larger value of persistence, the output partition identified with Louvain after the phase transition gets closer to the planted partition. This is consistent with our understanding of the phase transition. When the value of ω reaches $\max_{i,j,s} B_{ijs}$, only inter-layer merges form during the first completion of Louvain's phase 1. That is, when $\omega > \max_{i,j,s} B_{ijs}$, each set of nodes at the end of the first completion of phase 1 has the form $\{i_{s_1}, \ldots, i_{s_k}\}$, for some $\{s_1, \ldots, s_k\} \subseteq \{1, \ldots, |\mathcal{T}|\}$. We refer to these sets as "inter-layer chains". When $\omega > \max_{i,j,s} B_{ijs}$, the output partition becomes more representative of a partition in which the modularity matrix (or parts of the modularity matrix, depending on how inter-layer chains are formed during the first completion of phase 1 as one loops over nodes in a random order) of certain layers are aggregated together. Provided these chains are sufficiently short (because in our benchmark, only a few consecutive layers look alike, as observed in Fig. 5.2(b) and Fig.5.7(a)), one could imagine that the output of Louvain after the phase transition could reflect the planted partition more accurately for larger p.

Furthermore, the output of Louvain after the phase transition could also reflect the planted partition more accurately than before the phase transition for larger values of μ (since aggregation of single-layer modularity matrices can emphasize planted structure and can thus help mitigate "noise" induced by larger values of μ). In fact, when $p = 1$, we observe in Fig. 5.9(c) that the output partition identified with Louvain after the phase transition is at least as close to the planted partition as the output partition identified before the phase transition for all values of μ. This occurs because any aggregation of single-layer modularity matrices is representative of planted structure in any layer (because planted structure is the same in all layers when $p = 1$). We therefore observe that Louvain in Fig. 5.9(c) performs similarly to LouvainRand in Fig. 5.9(f) up to roughly $\mu \approx 0.3$, after which Louvain is better (since the inter-layer chains help to mitigate the "noise" induced by larger values of μ). To test our interpretation, we re-apply Louvain and post-process the multilayer partition after each pass of the first phase during its first completion. Doing this causes the inter-layer chains to have length $|\mathcal{T}|-1$ once $\omega > \max_{i,j,s} B_{ijs}$ (and Louvain becomes equivalent to performing modularity maximization on the mean single-layer modularity matrix). The qualitative behaviour of the phase transition is similar for all values of $p < 1$ and different at $p = 1$. We show the cases $p \in \{0, 0.9, 1\}$ in Fig. 5.10(a,b,c).

In the case of LouvainRand, the qualitative behaviour of the plots is similar for all values of $p \in \{0, 0.1, \ldots, 1\}$ and all values of $\omega \in \{0, 0.2, \ldots, 2\}$. One explanation for this is that the network is sufficiently dense that the multilayer partition can either be recovered by all values of ω or by none. We note that although it is not clear in the figures (because the curves are close together), the order of the curves changes near $p = 0.5$. That is, the red curves are slightly above the blue curves, instead of the other way around. It seems that for values of p smaller than 0.5, ω favors more persistence than is present in the planted partition. We repeat this experiment with a lower value of κ and show the results for Louvain and LouvainRand for $p \in \{0, 0.2, \ldots, 0.8, 1\}$ in Fig. 5.11. The separation between the blue curves and the red curves for $p = 1$ in Fig. 5.11(l) is more accentuated for lower values of κ (as suggested in Fig. 5.8). We show an example in Fig. 5.12 for $p = 1$ and $\kappa = 0.05$.

In the case of the spectral bipartitioning, the qualitative behaviour of all plots is similar except for $p = 1$. We suspect this is because the eigenvectors of the multilayer modularity matrix are "localized", as discussed in Section 4.2.3.2. It follows that the leading eigenvector mainly contains information about the layer with the leading eigenvalue, and hence the performance of spectral bipartitioining only improves when $p = 1$ (because all layers are identical in this case).



(a) $p = 0$, L, $\kappa = 0.2$

(b) $p = 0.2$, L, $\kappa = 0.2$

(c) $p = 0.4$, L, $\kappa = 0.2$

(d) $p = 0.6$, L, $\kappa = 0.2$

(e) $p = 0.8$, L, $\kappa = 0.2$

(f) $p = 1$, L, $\kappa = 0.2$

(g) $p = 0$, LR, $\kappa = 0.2$

(h) $p = 0.2$, LR, $\kappa = 0.2$

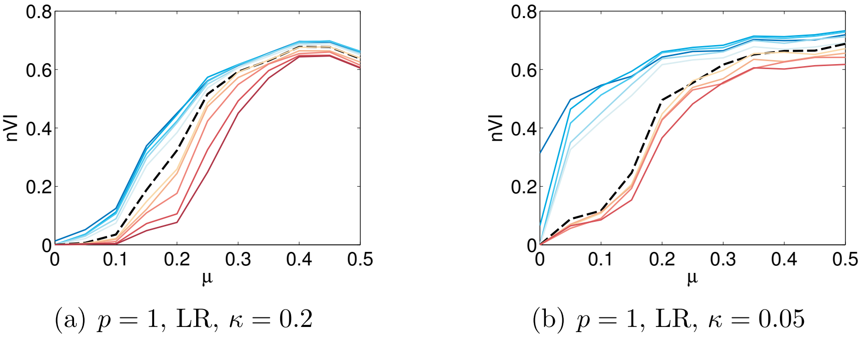
(i) $p = 0.4$, LR, $\kappa = 0.2$

(j) $p = 0.6$, LR, $\kappa = 0.2$

(k) $p = 0.8$, LR $\kappa = 0.2$

(l) $p = 1$, LR, $\kappa = 0.2$

Figure 5.11: We fix $(N, \mathcal{T}) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. The sample of ω values is $\{0, 0.2, \ldots, 1.8, 2\}$. We show plots for $\omega < 1$ with solid blue curves, those for $\omega > 1$ with solid red curves, and that of $\omega = 1$ with a black dashed curve. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \ldots, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.05, \ldots, 4.45, 0.5\}$ and $\kappa = 0.2$. (a,b,c,d,e,f) Louvain (L) and $p \in \{0, 0.2, \ldots, 0.8, 1\}$, respectively. (g,h,i,j,k,l) LouvainRand (LR) and $p \in \{0, 0.2, \ldots, 0.8, 1\}$, respectively. We perform one run of L and LR for each value of ω on each multilayer network.



(a) $p = 1$, LR, $\kappa = 0.2$

(b) $p = 1$, LR, $\kappa = 0.05$

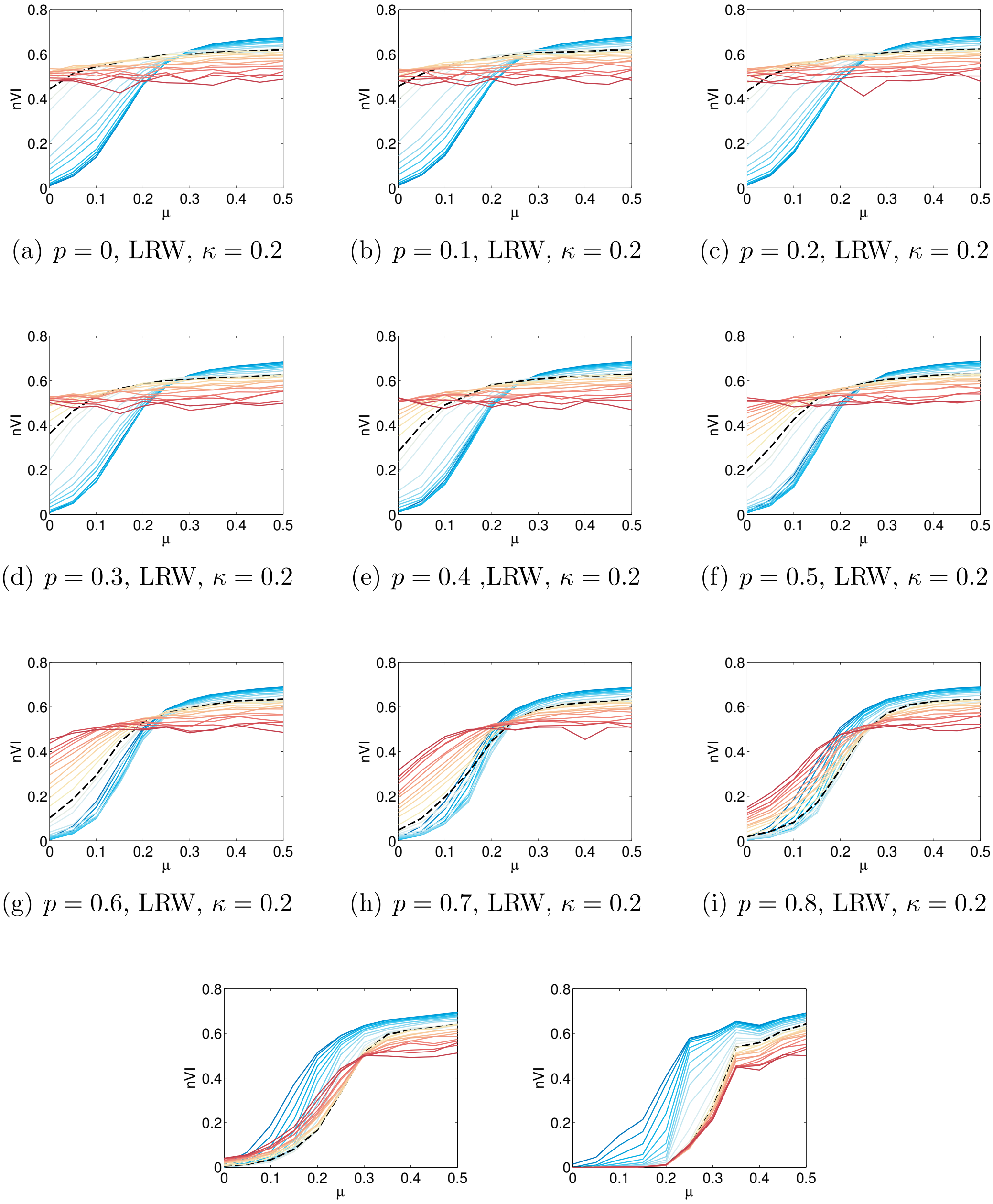
Figure 5.12: We fix $(N, \mathcal{T}) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. The sample of ω values is $\{0, 0.2, \ldots, 1.8, 2\}$. We show plots for $\omega < 1$ with solid blue curves, those for $\omega > 1$ with solid red curves, and that of $\omega = 1$ with a black dashed curve. We generate a multilayer partition using our benchmark in Table 5.1 with $p = 1$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.05, \ldots, 4.45, 0.5\}$ and $\kappa \in \{0.05, 0.2\}$.

5.3.1.2 Variants of Louvain

In this section, we perform similar experiments to those of the previous section with larger values of ω, and the same values of p and μ. We notice that as one increases ω, LouvainRand has a tendency to over-merge nodes into sets within layers (in fact this is what causes the discrepancy between the dashed curves and solid curves in Fig. 5.7(c,d)). (The resulting multilayer partition can even be a single community for large enough values of ω.) If one uses the parameter choices in Fig. 5.11, this phenomenon starts to manifest with LouvainRand for $\omega \approx 5$ for all values of p. The manifestation tends to be more pronounced when p is small and/or when κ is small. One explanation for this is that as one increases ω, many moves that are suboptimal for individual layers but that increase persistence slightly become possible. Because of transitivity of community assignments, this can result in many nodes from the same layer being merged into the same set. In some sense, this is partly a consequence of the fact that the heuristic performs a local search on the space of partitions (in particular, only one node is moved at a time). When ω is large enough, many more moves can increase multilayer modularity (and are thus feasible moves with equal probability in LouvainRand) because inter-layer contribution can compensate for many suboptimal intra-layer merges.

To try and mitigate this problem, we implement a second variant of Louvain where instead of moving a node to a set that increases modularity chosen uniformly at random, we move a node to a set that increases modularity with probability proportional to the increase (we define the probability to be the increase of a particular move divided by the sum of all possible increases). We call this heuristic Louvain-RandWeighted (LRW) and show the results in Fig. 5.13. The phenomenon that we mention in the previous paragraph seems to be less pronounced with this variant of Louvain. Although one would expect the issue to potentially still manifest for some range of ω values, the probability of an inter-layer merge will be much larger than the probability of an intra-layer merge for large enough ω, and one would expect LRW to then favor inter-layer merges over intra-layer merges during the first completion of phase 1.

We observe in Fig. 5.13(i,j), where $p \in \{0.8, 0.9\}$ that – as one would anticipate – increasing the value of ω produces an output partition that is closer to the planted partition up to $\omega \approx 2.4$, after which ω favors more persistence than is present in the planted partition and the nVI between the identified and planted partitions starts to increase. This does not apply to the case $p = 1$ in Fig. 5.13(k) (because a planted partition with $p = 1$ has maximal persistence so no amount of favored persistence is "too high"). We repeat the experiments in Fig. 5.13 with the Jaccarad coefficient, a different normalization of VI, and a uniform choice of null distribution between and within layers (i.e., $\boldsymbol{p}_s = \boldsymbol{p}$ for all s, and $p_i = p_j$ for all $i, j \in \{1, \dots, |\boldsymbol{p}|\}$) in Appendix C to check that the qualitative behaviour observed in Fig. 5.13 is relatively robust to our choice of similarity measure and our choice of null distribution.



(a) $p = 0$, LRW, $\kappa = 0.2$

(b) $p = 0.1$, LRW, $\kappa = 0.2$

(c) $p = 0.2$, LRW, $\kappa = 0.2$

(d) $p = 0.3$, LRW, $\kappa = 0.2$

(e) $p = 0.4$,LRW, $\kappa = 0.2$

(f) $p = 0.5$, LRW, $\kappa = 0.2$

(g) $p = 0.6$, LRW, $\kappa = 0.2$

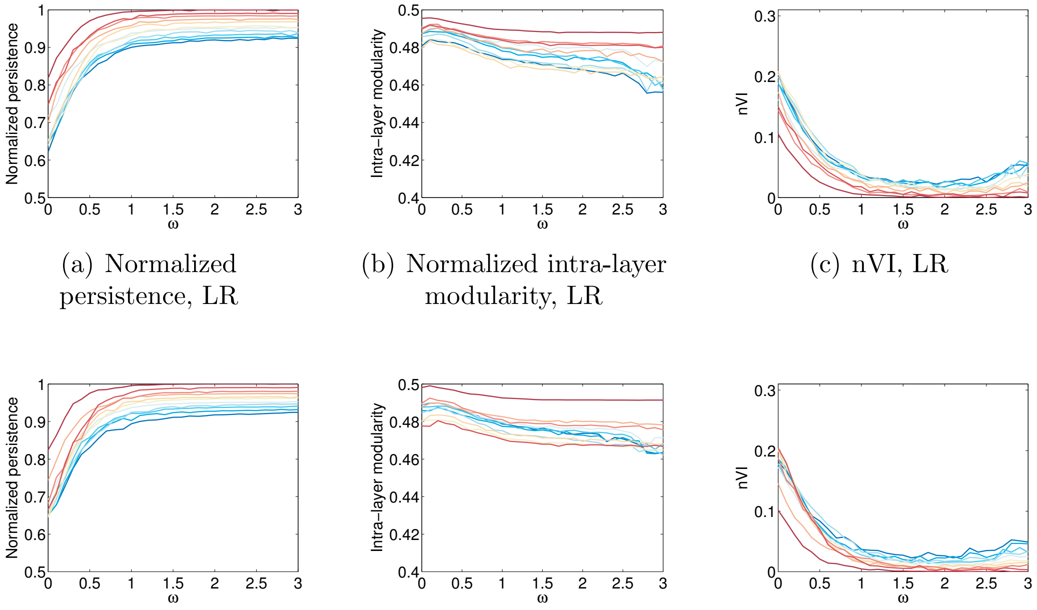
(h) $p = 0.7$, LRW, $\kappa = 0.2$

(i) $p = 0.8$, LRW, $\kappa = 0.2$

(j) $p = 0.9$, LRW, $\kappa = 0.2$

(k) $p = 1$, LRW, $\kappa = 0.2$

Figure 5.13: We fix $(N, \mathcal{T}) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. The sample of ω values is $\{0, 0.2, \ldots, 4.8, 5\}$. We show plots for $\omega < 2.4$ with solid blue curves, those for $\omega > 2.4$ with solid red curves, and that of $\omega = 2.4$ with a black dashed curve. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \ldots, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\text{min}} = 0.1N$, and the upper bound on the expected community size is $|C|_{\text{max}} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.05, \ldots, 4.45, 0.5\}$ and $\kappa = 0.2$. (a,b,c,d,e,f,g,h,i,j,k) LouvainRandWeighted (LRW) and $p \in \{0, 0.1, \ldots, 0.9, 1\}$. We obtain the output partition after iterating LRW on a post-processed output partition until the partition no longer changes (as we did in Figs 5.5–5.8). The number of iterations is no more than five in all cases. In Appendix C, we show the results for output partitions after one iteration.



(a) Normalized persistence, LR

(b) Normalized intra-layer modularity, LR

(c) nVI, LR

(d) Normalized persistence, LRW

(e) Normalized intra-layer modularity, LRW

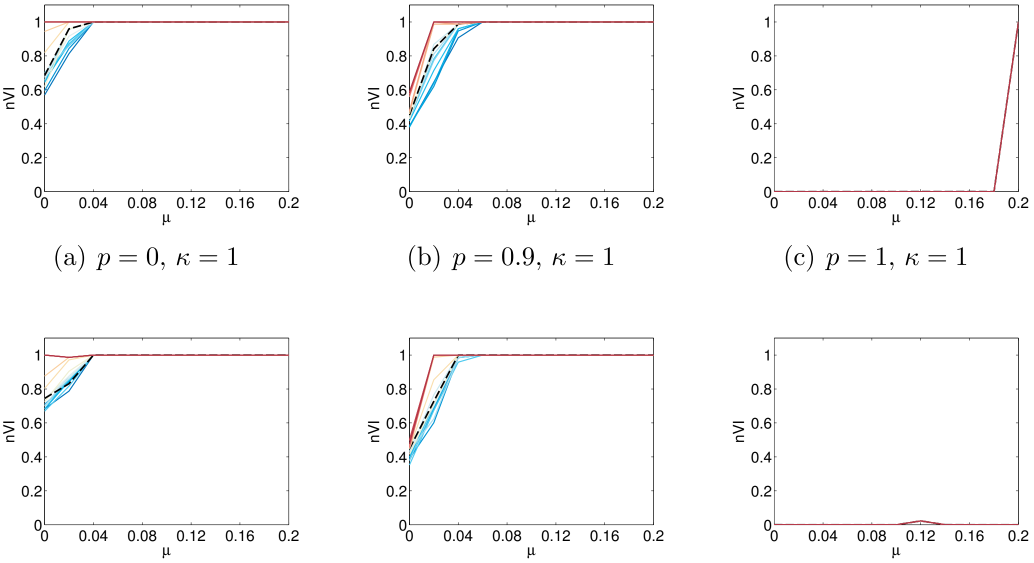
(f) nVI, LRW

Figure 5.14: We fix $(N, |\mathcal{T}|) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0.9, 0.91, \dots, 0.99, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\text{min}} = 0.1N$, and the upper bound on the expected community size is $|C|_{\text{max}} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu = 0.1$ and $\kappa = 0.2$. The sample of ω values is $\{0, 0.1, \dots, 2.9, 3\}$. We show plots of normalized persistence, normalized intra-layer modularity, and nVI between the output and planted multilayer partitions for (a,b,c) LouvainRand, and (d,e,f) LouvainRandWeighted. We normalize persistence by $N(|\mathcal{T}| - 1)$ and intra-layer modularity by $\sum_{s=1}^{|\mathcal{T}|}(\mathbf{1}^T \boldsymbol{A}_s \mathbf{1})$. We obtain the output partitions that we use in these panels after iterating each heuristic on a post-processed output partition until the partition no longer changes (as we did in Figs 5.5–5.8). The number of iterations until convergence is not more than five in all cases. In Appendix C, we show the results for output partitions after one iteration and after convergence for values of $\omega \in \{0, 0.1, \dots, 9.9, 10\}$.

We end this section with an experiment in which we fix $(\mu, \kappa) = (0.1, 0.2)$, so that one cannot recover the planted partition with $\omega = 0$, and we show the values of persistence, intra-layer modularity, and nVI as a function of ω for LouvainRand and LouvainRandWeighted. One can see in our previous experiments in Fig. 5.13 that increasing ω starts to help with the recovery of a planted partition for $p \gtrapprox 0.6$ (this is consistent with Fig. 5.11(g–l)), and that values of $\omega \gtrapprox 2.4$ (red curves) tend to favor more persistence than is present in the planted partition for $p \in \{0.8, 0.9\}$. This is partly because of how our multilayer benchmark partition is designed: the partition induced on individual layers is different across all layers unless $p = 1$, and partitions that are more than ten layers apart are already rather dissimilar (see, e.g., Fig. 5.3). Accordingly, we consider the sample $\omega \in \{0, 0.1, \ldots, 2.9, 3\}$ and take a closer look at the interval $p \in [0.9, 1]$ in Fig. 5.14. (The plots in each panel of Fig. 5.14 now correspond to different values of p and not to different values of ω.) As ω becomes larger, the values of persistence tend to increase and the values of intra-layer modularity tend to decrease in Figs 5.14(a,b,d,e). The nVI between the recovered and planted partition reaches its minimum near $\omega \approx 1.5$ in all cases. As ω increases, the nVI between the planted partition and the identified partition seems to increase more rapidly for smaller values of p (i.e., lower expected value of persistence in the planted partition) in Figs 5.14(c,f) and less rapidly for larger values of p (i.e., larger expected value of persistence in the planted partition) in Figs 5.14(c,f). In theory, the only value of p for which nVI should never increase is $p = 1$, because for large enough values of ω, the globally optimal partition will have maximal persistence by Proposition 4.2.5 (and the only planted partition with maximal persistence irrespective of one's choice of null distribution $\mathbb{P}_0^s$ is that of $p = 1$.) We show the analogous figure to Fig 5.14 for $\omega \in \{0, 0.1, \ldots, 9.9, 10\}$ in Appendix C.

5.3.2 Multilayer Infomap

We end this chapter with preliminary experiments using multilayer map minimization (see Section 2.2.3). We use the publicly available heuristic "Infomap" [1] to identify output partitions. We consider two cases: (1) a "multiplex case", in which the transition probabilities are given in (2.26) (this is the case considered in [56]), and (2) a "temporal case", in which the transition probabilities are given by the diagonal blocks and off-diagonal blocks in (2.26), and all other blocks are set to zero. (In other words, the random walker can only "relax" to an adjacent layer.) We take $(N, |\mathcal{T}|) = (100, 50)$, $\mu \in \{0, 0.2, \ldots, 0.18, 0.2\}$, $r \in \{0, 0.01, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1\}$, and $\kappa \in \{0.2, 1\}$ (note that our discretization for r is finer in $[0, 0.1]$ than in $[0.1, 1]$). Although we perform this experiment for all values of $p \in \{0, 0.1, \ldots, 0.9, 1\}$, we only include plots for $p \in \{0, 0.9, 1\}$ because these illustrate the three main qualitatively different cases. We show the results for all values of $p \in \{0, 0.1, \ldots, 0.9, 1\}$ for both the temporal case and the multiplex case in Appendix C. We first comment on Fig. 5.15, where we take $\kappa = 1$ (as we did in Fig. 5.9). Note that a value of nVI equal to 1 indicates that the identified partition corresponds to a single community. One observation is that multilayer Infomap does not recover the planted partition for any value of μ in our sample when $p < 1$. (Recall from our network model in (5.6) that when $\mu = 0$ and $\kappa = 1$, the planted communities in each layer correspond to disconnected



(a) $p = 0$, $\kappa = 1$

(b) $p = 0.9$, $\kappa = 1$

(c) $p = 1$, $\kappa = 1$

(d) $p = 0$, $\kappa = 1$

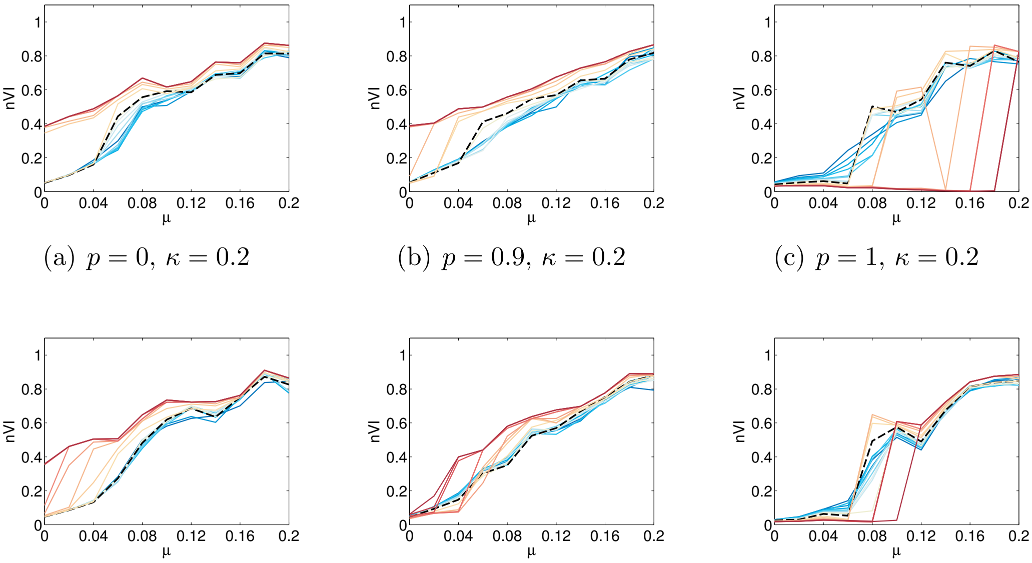
(e) $p = 0.9$, $\kappa = 1$

(f) $p = 1$, $\kappa = 1$

Figure 5.15: We fix $(N, \mathcal{T}) = (100, 50)$. The sample of "relax-rate" values is $r \in \{0, 0.01, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1\}$. We show plots for $r < 0.1$ with solid blue curves, those for $r > 0.1$ with solid red curves, and that of $r = 0.1$ with a black dashed curve. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.02, \ldots, 0.18, 0.2\}$ and $\kappa = 1$. For each pair (p, μ) we generate a multilayer partition and a multilayer network. We perform one run of Infomap for each value of r on each multilayer network. (a,b,c) Multiplex-case Infomap and $p \in \{0, 0.9, 1\}$. (d,e,f) Temporal-case Infomap and $p \in \{0, 0.9, 1\}$.

cliques in the single-layer adjacency matrices.) For small values of μ (e.g., $\mu = 0$), one would expect to obtain better results if one uses single-layer Infomap independently on each layer or if one initializes multilayer Infomap on the output partition of single-layer Infomap instead of an all-singleton partition. A second observation is that when $p = 1$, all curves collapse onto one curve in both the multiplex case and the temporal case. For the multiplex case, the planted partition is identified for all values of $\mu < 0.2$, after which the output partition abrubtly switches to a partition with a single community; and for the temporal case, the planted partition is identified for all values of $\mu \in \{0, 0.02, \ldots, 0.18, 0.2\}$.

In Fig. 5.16, we repeat the experiments illustrated in Fig. 5.15 with $\kappa = 0.2$ instead of $\kappa = 1$. We only include plots for $p \in \{0, 0.9, 1\}$ (because these illustrate the main three qualitatively different cases) and show the results for all values $p \in \{0, 0.1, \ldots, 0.9, 1\}$ in Appendix C. Interestingly, the identified partition is not a single



(a) $p = 0$, $\kappa = 0.2$

(b) $p = 0.9$, $\kappa = 0.2$

(c) $p = 1$, $\kappa = 0.2$

(d) $p = 0$, $\kappa = 0.2$

(e) $p = 0.9$, $\kappa = 0.2$

(f) $p = 1$, $\kappa = 0.2$

Figure 5.16: We fix $(N, \mathcal{T}) = (100, 50)$. The sample of "relax-rate" values is $\{0, 0.01, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1\}$. We show plots for $r < 0.1$ with solid blue curves, those for $r > 0.1$ with solid red curves, and that of $r = 0.1$ with a black dashed curve. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.02, \ldots, 0.18, 0.2\}$ and $\kappa = 0.2$. For each pair (p, μ) we generate a multilayer partition and a multilayer network. We perform one run of Infomap for each value of r on each multilayer network. (a,b,c) Multiplex-case Infomap and $p \in \{0, 0.9, 1\}$. (d,e,f) Temporal-case Infomap and $p \in \{0, 0.9, 1\}$.

community for any parameter choice in our sample when $\kappa = 0.2$, and the value of nVI is significantly lower than it is when $\kappa = 1$ for small values of μ and $p < 1$. The temporal case seems to be qualitatively similar to the multiplex case, though the former seems to outperform the latter slightly when $p < 1$ and the latter seems to output the former when $p = 1$. (One can also observe this in Figs C.10–C.15 of Appendix C.) We note that if we repeat the experiments in Fig. 5.16 with $\kappa = 0.1$, the nVI > 0.2 in all cases. This suggests that the threshold for κ below which multilayer Infomap cannot recover a planted partition lies between 0.2 and 0.1. By contrast, with multilayer modularity maximization it is still possible to recover the planted partition for $\kappa = 0.05$ (see, e.g., Fig 5.12). We repeat the experiments in Fig. 5.15 with $(N, |\mathcal{T}|) = (150, 100)$ (instead of $(N, |\mathcal{T}|) = (100, 50)$) and show the results for $p \in \{0, 0.1, \ldots, 0.9, 1\}$ in Appendix C. A better understanding of whether these qualitative results are an artifact of the Infomap heuristic, features of the quality function, or some combination of both would require further investigation of both the multilayer map minimization method and the computational heuristic.

5.4 Summary

We proposed a benchmark for community-detection in temporal networks. Our benchmark consists of generating (1) a sequence of single-layer partitions that are correlated through a simple model of persistence and (2) a sequence of single-layer networks using a stochastic block model. We perform various numerical experiments to compare the behaviour of different heuristics for solving multilayer modularity maximization and we perform preliminary experiments with multilayer map minimization using a publicly available heuristic "Infomap" [1]. We use a simple benchmark to shed light on features of these methods and heuristics that may not manifest as clearly when using real data.

In our experiments with multilayer modularity maximization, we use spectral bipartitioning (see Section 2.3.3) and two variants of Louvain (see Section 2.3.2): LouvainRand (LR), in which each node is moved to a set chosen uniformly at random that increases modularity; and LouvainRandWeighted (LRW), in which each node is moved to a set with probability given by the increase of the move divided by the sum of all possible increases. The "phase transition" of Louvain that we describe in Section 4.2.3.1 clearly manifests in our benchmark experiments and is marked by a qualitative decline in the heuristic's performance (unless the planted multilayer partition has maximal persistence). We observe that both LR and LRW seem to mitigate this issue. We further observe that both LR and LRW outperform spectral bipartitioning in our experiments. The qualitative results that we obtain for LR and LRW tend to be similar for smaller values of inter-layer coupling, and LRW tends to outperform LR for larger values of inter-layer coupling. In particular, we notice that LR has a tendency to over-merge nodes within layers when one increases the value of inter-layer coupling. Iterating each heuristic on a post-processed (see Section 4.2.3.1) output partition until it no longer changes seems to improve output partitions for both LR and LRW. We illustrated the trade-off between persistence and intra-layer modularity discussed in Section 4.2 with both LR and LRW. Preliminary experiments with multilayer Infomap suggest that multilayer modularity maximization with variants of Louvain outperforms multilayer Infomap on our benchmark.

Chapter 6

Application to Financial Correlation Networks

In this chapter, we perform numerical experiments with the data sets AssetClasses and SingleAssetClass (see Section 3.3 for descriptions) based on the results and observations of Chapters 4 and 5. We use the multilayer-network representation described in Chapter 3 in all experiments. We give some details about the content of communities identified with different heuristics and with different null networks.

6.1 Numerical experiments with AssetClasses

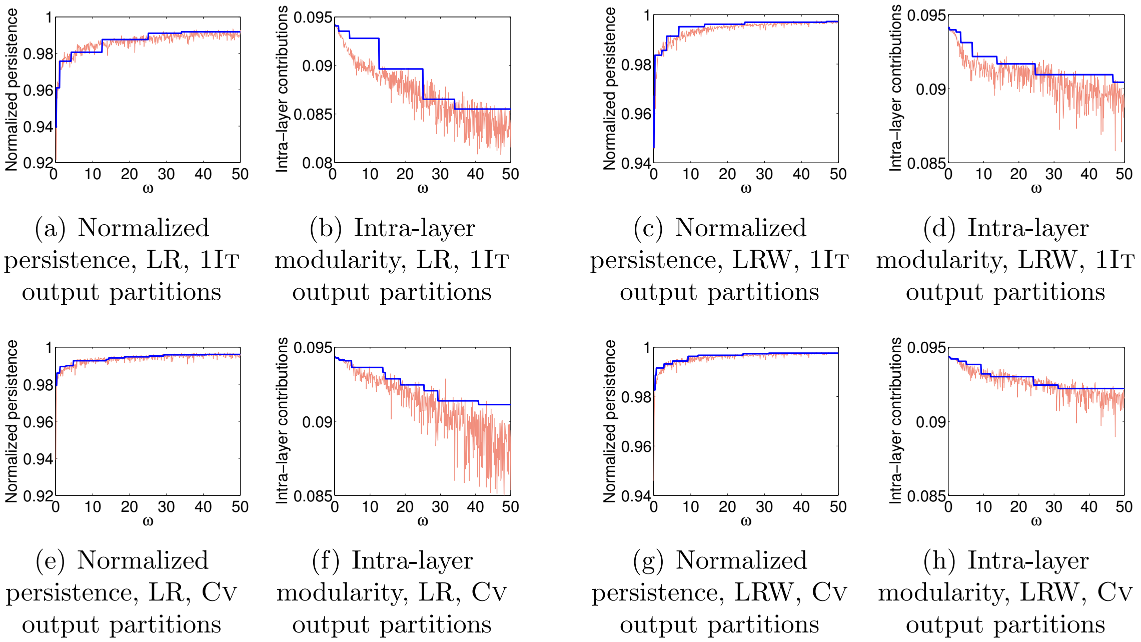
6.1.1 Comparisons with variants of Louvain

We first examine plots of persistence (4.3) and plots of intra-layer modularity (4.4) for output multilayer partitions obtained with LouvainRand (LR) and LouvainRandWeighted (LRW). Recall that LR and LRW only differ from the Louvain heuristic (see Section 2.3.2) in their condition for moving nodes in phase 1. In LR, one moves each node to a set, chosen uniformly at random, that increases the quality function. In LRW, one moves each node to a set, chosen with a probability which is given by the increase of the move divided by the sum of all possible increases.

For each heuristic, we examine two output multilayer partitions: (1) the output multilayer partition obtained with an all singleton initial partition and (2) the output multilayer partition that we obtain with the following three-step procedure:

(a) We store an output multilayer partition obtained with an all singleton initial partition.

(b) We store an output multilayer partition obtained with the previous multilayer output partition as the initial partition.



(a) Normalized persistence, LR, 1It output partitions

(b) Intra-layer modularity, LR, 1It output partitions

(c) Normalized persistence, LRW, 1It output partitions

(d) Intra-layer modularity, LRW, 1It output partitions

(e) Normalized persistence, LR, Cv output partitions

(f) Intra-layer modularity, LR, Cv output partitions

(g) Normalized persistence, LRW, Cv output partitions

(h) Intra-layer modularity, LRW, Cv output partitions

Figure 6.1: Persistence and intra-layer modularity plots for 1IT output partitions with (a,b) LR, and (c,d) LRW; and persistence and intra-layer modularity plots for Cv output partitions with (e,f) LR, and (g,h) LRW. Each point on the red curve is computed with the identified partition (i.e., either an 1IT output partition or a Cv output partition), and each point on the blue curve is computed with the partition that yields the highest value of multilayer modularity for the corresponding ω value amongst all identified partitions. Our sample of ω values is $\{0, 0.1, \ldots, 49.9, 50\}$, and we use the uniform null network (i.e., $P_{ijs} = \langle \boldsymbol{A}_s \rangle$) with $\gamma = 1$ in all cases. We normalize persistence by $N(|\mathcal{T}|-1)$ and intra-layer modularity by $\sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} A_{ijs}$.

(c) We iterate (b) until the output multilayer partition no longer changes.

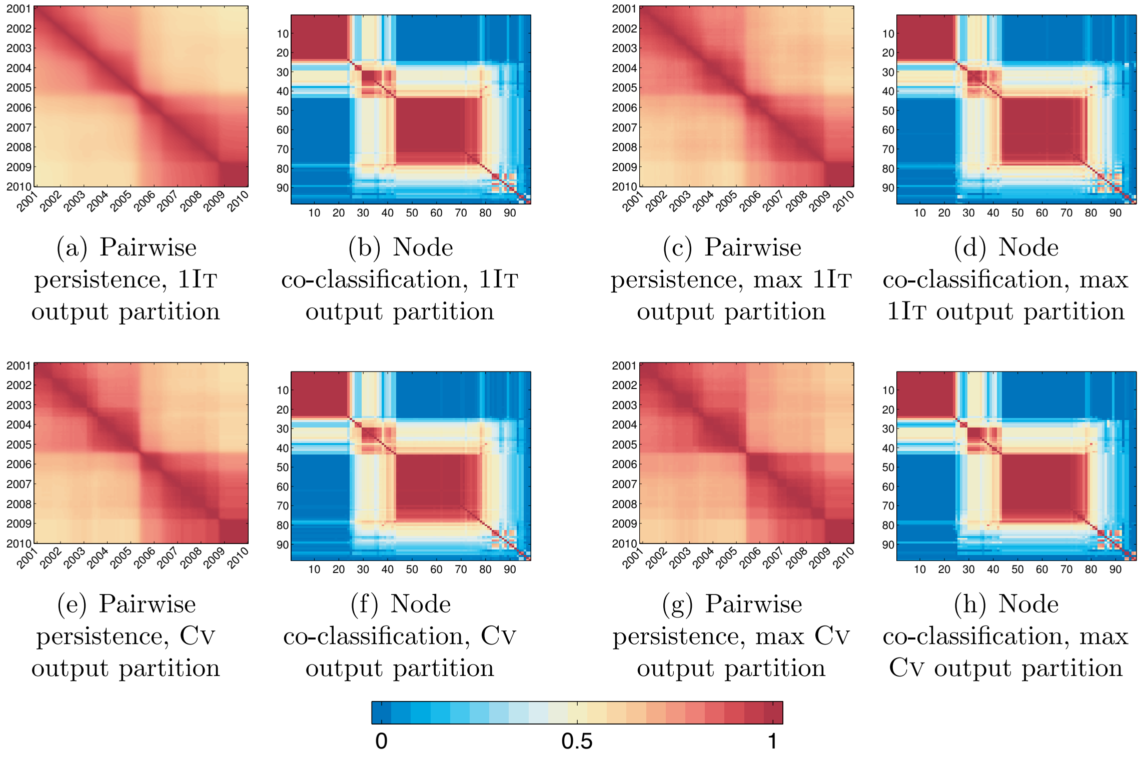
We refer to the output multilayer partition in (1) as “1It output partition” and to the output multilayer partition in (2) as “Cv output partition”. The number of iterations until convergence to the Cv output partition for both heuristics and all values of ω varies between 2 and 6 for the AssetClasses data set. We apply our post-processing procedure from Section 4.2.3.1 to all 1It output partitions and to all Cv output partitions. (Recall that post-processing maximizes the persistence of a multilayer partition without changing the partitions that are induced on individual layers.) We use the uniform null network on correlation matrices linearly shifted to [0, 1] with a resolution parameter of $\gamma = 1$ in all experiments,[1] and we take $\{0, 0.1, \ldots, 49.9, 50\}$ as our sample of ω values (giving 501 values in total).

[1] By linearity of the mean, multilayer modularity maximization with a uniform null network (i.e., $P_{ijs} = \langle \boldsymbol{A}_s \rangle$) on a linearly shifted correlation matrix (i.e., $\boldsymbol{A}_s \in [0, 1]$) and an unshifted correlation matrix (i.e., $\boldsymbol{A}_s \in [-1, 1]$) yield equivalent maximization problems when $\gamma = 1$.

In Fig. 6.1, we show plots of persistence and intra-layer modularity for the 1It output partitions and the Cv output partitions with LR (four panels on the left-hand side) and for the 1It output partitions and the Cv output partitions with LRW (four panels on the right-hand side). We show two curves in panels (a)–(d). Each point on the red curve is computed with the corresponding 1It output partition, and each point on the blue curve is computed with the 1It output partition that yields that largest value of multilayer modularity for the corresponding ω value. Similarly, each point on the red curve in panels (e)–(h) is computed with the corresponding Cv output partition, and each point on the blue curve is computed with the Cv output partition that yields that largest value of multilayer modularity for the corresponding ω value. We store one 1It multilayer partition and one Cv output partition for each value of ω. This gives a total of 501 multilayer partitions per panel.

As anticipated, the blue curves that correspond to persistence are non-decreasing functions of ω and the blue curves that correspond to intra-layer modularity are non-increasing functions of ω. The increase of persistence and the decrease of intra-layer modularity is not monotonic for the red curve, because there is a single local optimum for each value of ω. One would expect the red curves to be "smoother" if one averages over multiple runs of the heuristic [compare Fig. 6.1(b) with Fig. 4.10(b)], as one then has multiple local optima instead of a single local optimum for each ω value. The difference between the red and the blue curves appears to be more pronounced with LR than with LRW, and in particular, seems to be least pronounced in Fig. 6.1(g,h).

In Fig. 6.2, we focus on output partitions obtained with LRW [i.e., the output partitions that we use to generate Fig. 6.1(c,d,g,h)]. We consider 501 1It output partitions, 501 Cv output partitions, a set of 501 output partitions in which the k^{th} partition yields the largest value of multilayer modularity for the k^{th} value of ω in $\{0, 0.1, \ldots, 49.9, 50\}$ amongst all 1It output partitions, and a set of 501 output partitions in which the k^{th} partition yields the largest value of multilayer modularity for the k^{th} value of ω in $\{0, 0.1, \ldots, 49.9, 50\}$ amongst all Cv output partitions. The last two sets of output partitions contain duplicates because the same output multilayer partition can yield the largest value of multilayer modularity for multiple ω values [as suggested by the presence of "plateaus" in Fig. 6.1(c,d,g,h)]. In Figs 6.2(a,c,e,g), we show the normalized $|\mathcal{T}| \times |\mathcal{T}|$ pairwise persistence matrix with entries $\text{Pers}(C)|_{s,r} = \left(\sum_{i=1}^{N} \delta(c_{i_s}, c_{i_r})\right)/N$ averaged over the ω values. In Figs 6.2(b,d,f,h), we show the $N \times N$ association matrix who entries are the co-classification index $\left(\sum_{s=1}^{|\mathcal{T}|} \delta(c_{i_s}, c_{j_s})\right)/|\mathcal{T}|$ between pairs of nodes across layers averaged over the ω values. As we discussed in Chapter 4, the former can give an idea



(a) Pairwise persistence, 1IT output partition

(b) Node co-classification, 1IT output partition

(c) Pairwise persistence, max 1IT output partition

(d) Node co-classification, max 1IT output partition

(e) Pairwise persistence, CV output partition

(f) Node co-classification, CV output partition

(g) Pairwise persistence, max CV output partition

(h) Node co-classification, max CV output partition

Figure 6.2: We show normalized pairwise persistence [i.e., $\text{Pers}(C)|_{s,r} = \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_r})/N$] and pairwise co-classification of nodes across layers [i.e., $\left(\sum_{s=1}^{|\mathcal{T}|} \delta(c_{i_s}, c_{j_s})\right)/|\mathcal{T}|$] averaged over $\omega \in \{0, 0.1, \ldots, 49.9, 50\}$ for (a,b) 1IT output partitions , (c,d) 1IT output partition for each ω with highest multilayer modularity, (e,f) CV output partitions, and (g,h) CV output partitions for each ω with highest multilayer modularity. Dates in panels (a,c,e,g) correspond to the first layer that contains a data point from the specified year. We use LRW and the uniform null network (i.e., $P_{ijs} = \langle \boldsymbol{A}_s \rangle$) with $\gamma = 1$ in all cases.

about when connectivity patterns change in a temporal network, and the latter can give an idea about how they have changed. We reorder all matrices identically in Figs 6.2(b,d,f,h) using a node ordering based on the partitions that we obtain with the U null network that emphasizes block-diagonal structure in each matrix.

Observe that panels in Figs 6.2(a,c,e,g) and panels in Figs 6.2(b,d,f,h) are visually similar. This suggests that pairwise persistence and pairwise node co-classification across layers averaged over ω values in $\{0, 0.1, \ldots, 49.9, 50\}$ with LRW are fairly consistent between 1IT output partitions, CV output partitions, the subset of 1IT output partitions with highest multilayer modularity, and the subset of CV output partitions with highest multilayer modularity.[2] We first focus on Figs 6.2(a,c,e,g). We find

[2] The panels in Fig. 6.2 are also visually similar to Figs 4.10(d,e), which we computed with multiple runs of LR and which we briefly discussed in Section 4.1.2. This suggests that pairwise persistence

that the most noticeable separation between diagonal blocks in the middle of panels (a,c,e,g) occurs near the layer that corresponds to the time interval 01 Aug 03–24 Jun 05. Although it is not clear whether this time interval coincides with a notable event, the correlation network (as it is currently computed) does exhibit a change in connectivity patterns around that period. Roughly speaking, connectivity patterns in the correlation network around 01 Aug 03–24 Jun 05 change so that government and corporate bond assets belong to a first community, equity assets belong to a second community, and most remaining assets either belong to neither of these two communities or they belong to the community that contains equity assets. (See the correlation matrix in Fig. 3.4(d) for an example of how connectivity patterns change during that period.) This change in connectivity patterns becomes more pronounced after the Lehman bankruptcy. (See, e.g., the correlation matrix in Fig. 3.4(f).) The dark red diagonal block in the bottom right corner of Figs 6.2(a,c,e,g) between the end of 2008 and 2010 begins near the layer that corresponds to 01 Dec 06–24 Oct 08. This layer contains five data points after the Lehman bankruptcy.

We now focus on Figs 6.2(b,d,f,h). Recall that diagonal blocks in these panels correspond to assets that are often assigned to the same community across layers (i.e., across time) and across $\omega \in \{0, 0.1, \ldots, 49.9, 50\}$. As pointed out earlier, we use the same node ordering for all panels. Since panels (b,d,f,h) are visually similar, the asset content of diagonal blocks in panels (b,d,f,h) is fairly consistent across panels. We list the assets that belong to some of the more striking red diagonal blocks of Fig. 6.2(h). The first red diagonal block (i.e., nodes 1–24 in Fig. 6.2(h)) contains all 24 government and corporate bonds (i.e., all the assets in Table A.1). The Japanese government bond ("JNGATR") has a lower co-classification index with the rest of the bonds (it is approximately equal to 0.9) relative to the co-classification the rest of the bonds have with each other (approximately equal to 1). The very small diagonal block directly following the bond diagonal block (i.e., nodes 27–29 in Fig. 6.2(h)) corresponds to three of the four fuel assets in Table A.3. These are "HO1" (Heating oil), "CL1" (Crude oil, WTI), and "CO1" (Crude oil, brent). The natural gas fuel "NG1" is not in this block. The third slightly framented diagonal block (i.e., nodes 30–43 in Fig. 6.2(h)) contains 11 currencies and 3 metals. These are "EURUSD" (Euro), "GBPUSD" (Pounds sterling), "SEKUSD" (Swedish krona), "NOKUSD" (Norwegian krone), "CZKUSD" (Czech koruna), "CHFUSD" (Swiss franc), "ZARUSD" (South

and pairwise node co-classification across layers averaged over ω values in $\{0, 0.1, \ldots, 49.9, 50\}$ are fairly consistent between LR and LRW for the AssetClasses data set.

African rand), “IDRUSD” (Indonesian rupiah), “AUDUSD” (Australian dollar), “NZDUSD” (New Zealand dollar), “CADUSD” (Canadian dollar) “XAU” (Gold), “XAG” (Silver), and “XPT” (Platinum). The highest co-classification index in this block is between the 6 the european currencies (it is approximately 0.95) and the lowest co-classifiaction is between Platinum and the rest of the assets in the block (it is approximately 0.6).

The largest diagonal block (i.e., nodes 44–79 in Fig. 6.2(h)) contains all equity assets in Table A.2 except for “NZSE” (New Zealand All Ordinaries Index), 4 currencies, and 5 metals. In the order in which they appear in the block, we list these 36 assets in Table 6.1. The highest co-classification index in this block is between the first 24 equities of the list (it is approximately equal to 1) and the lowest co-classification index is between the Philippines peso and all other assets in the block (it is approximately equal to 0.7). Finally, by magnifying the bottow right corner of Fig. 6.2(h), one can also observe small blocks of commodities with a high co-classification index. For example nodes 83–85 correspond to “S 1” (Soybean), BO1 (Soybeanoil), and “‘SM1” (Soybean meal), and nodes 81–82 correspond to “CC1” (Cacao) and “KC1” (Coffee).

Ticker	Asset class	Description
GDUEEGFL	Equities	Emerging markets: Latin America
GDUEEGFA	Equities	Emerging markets: Europe, Middle East, Africa
SPX	Equities	Standard and Poors 500
INDU	Equities	Dow Jones Industrial Average Index
NDX	Equities	NASDAQ 100 Index
RTY	Equities	Russell 2000 Index
NKY	Equities	Nikkei 225 Index
HSI	Equities	Hang Seng Index
AS30	Equities	Australian All Ordinaries Index
SPTSX	Equities	S&P/Toronto Stock Exchange Index
UKX	Equities	FTSE 100 Index
DAX	Equities	German DAX Index
CAC	Equities	French CAC 40 Index
IBEX	Equities	Spanish IBEX 35 Index
FTSEMIB	Equities	Italian FTSE MIB Index
SMI	Equities	Swiss Market Index
BVLX	Equities	PSI General Index
ISEQ	Equities	Irish Overall Index
BEL20	Equities	Belgium BEL 20 Index
KFX	Equities	OMX Copenhagen 20 Index
HEX	Equities	Helsinki Stock Exchange General Index
OBX	Equities	Norwegian OBX Stock Index
OMX	Equities	OMX Stockholm 30 Index
ATX	Equities	Austrian Traded Index
ASE	Equities	Athens Stock Exchange General Index
AEX	Equities	AEX Index
MXNUSD	Currencies	Mexican peso
GDDUEMEA	Equities	Emerging markets: Asia
NZSE	Currencies	New Zealand dollar
HG1	Metals	Copper
LT1	Metals	Tin
LA1	Metals	Aluminium
LN1	Metals	Nickel
LL1	Metals	Lead
KRWUSD	Currencies	Korean won
PHPUSD	Currencies	Philippines peso

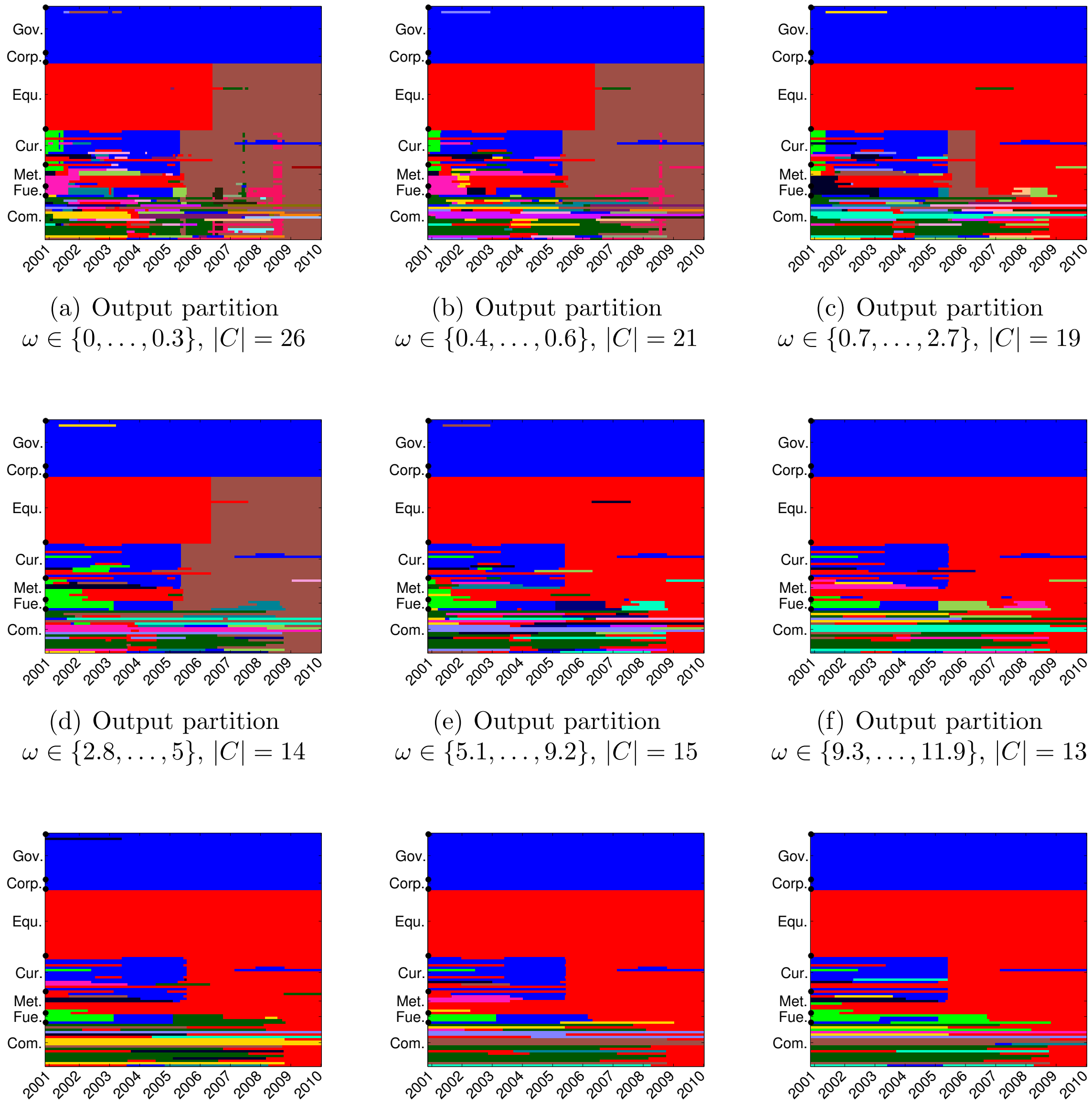
Table 6.1: Set of assets in the largest diagonal block (i.e., nodes 44–79) of Fig. 6.2(h).

6.1.2 Output multilayer partitions with increasing coupling

In Fig. 6.3, we show nine distinct multilayer partitions. Each partition corresponds to one of the LRW Cv output partitions that yields the largest value of multilayer modularity for some range of $\omega \in \{0, 0.1, \ldots, 49.9, 50\}$. In particular, we examine the nine multilayer partitions that correspond to the nine plateaus in Fig. 6.1(g,h). For each output partition, we specify the number of communities that it contains and the range of ω values in $\{0, 0.1, \ldots, 49.9, 50\}$ for which it yields the largest value of multilayer modularity.

Observe that the largest communities in each partition of Fig. 6.3 tend to be the one that contains bonds (blue community) and the one that contains equities (red community or brown community). The clear separation into diagonal blocks between 2005 and 2006 in Figs 6.2(a,c,e,g) manifests in all partitions of Fig. 6.3. In Figs 6.3(a–d), many currencies, metals, and fuels leave the bond community (blue) to form their own community (brown) between 2005 and 2006. In Figs 6.3(e–i), many currencies and metals leave the bond community (blue) to join the equity community (red) between 2005 and 2006. The Lehman bankruptcy (15 Sept 08) manifests in each panel at the end of 2008 and is marked by an increase in the size of the equity community in all partitions. Moreover, the four fuels join the bond community (blue) for the first time in all partitions at the beginning of 2003. In particular, this occurs in the layer that corresponds to the time interval 13 Apr 01–07 Mar 05. This is the last layer preceding the Iraq war of 21 Mar 05. (The increase between fuels and government bonds is visible in the correlation matrix of Fig. 3.4(c).) In Fig. 6.3(a,b), we observe that a set of assets separate from the equity community just before the Lehman bankruptcy to form their own community (see the pink streak at the end of 2008). These assets join the equity community (brown) after the Lehman bankruptcy and remain in it in all subsequent layers. The pink streak in Fig. 6.3(a,b) contains the following 28 assets : “EURUSD” (Euro), “GBPUSD” (Pounds sterling), “CHFUSD” (Swiss franc), “NOKUSD” (Norwegian krone), “SEKUSD” (Swedish krona), “CZKUSD” (Czech koruna), “XAU” (Gold), “XAG” (Silver), “XPT” (Platinum), “XPD” (Palladium), “LA1” (Aluminium), “LT1” (Tin), “LN1” (Lead), “CL1”(Crude oil, WTI), “CO1” (Crude oil, brent), “HO1” (Heating oil), “NG1” (Natural gas), “C 1” (Corn), “CC1” (Cocoa), “CT1” (Cotton), “KC1” (Coffee), “RR1” (Rough rice), “S 1” (Soybean), “QW1” (Sugar), “SM1” (Soybean meal), “W 1” (Wheat), “O 1” (Oats), and “BO1” (Soybean oil).

We end this section with a comment on three instances where — for a given asset class — most assets from the class belong to the same community but a couple do not. The government bond asset that stands out in Fig. 6.3(a–e,g) is “JNGATR” (Japanese government bond). The equity asset that stands out between the middle of 2006 and the middle of 2007 in Fig. 6.3(a–e) is “NZSE” (New Zealand All Ordinaries



(a) Output partition $\omega \in \{0, \ldots, 0.3\}$, $|C| = 26$

(b) Output partition $\omega \in \{0.4, \ldots, 0.6\}$, $|C| = 21$

(c) Output partition $\omega \in \{0.7, \ldots, 2.7\}$, $|C| = 19$

(d) Output partition $\omega \in \{2.8, \ldots, 5\}$, $|C| = 14$

(e) Output partition $\omega \in \{5.1, \ldots, 9.2\}$, $|C| = 15$

(f) Output partition $\omega \in \{9.3, \ldots, 11.9\}$, $|C| = 13$

(g) Output partition $\omega \in \{12, \ldots, 24.1\}$, $|C| = 11$

(h) Output partition $\omega \in \{24.2, \ldots, 31.1\}$, $|C| = 11$

(i) Output partition $\omega \in \{31.2, \ldots, 50\}$, $|C| = 11$

Figure 6.3: Multilayer Cv output partitions with highest multilayer modularity for the values of ω in the caption. We use LRW and the uniform null network (i.e., $P_{ijs} = \langle \boldsymbol{A}_s \rangle$) with $\gamma = 1$. Recall that we consider $\omega \in \{0, 0.1, \ldots, 49.9, 50\}$. Colors correspond to community assignments, and dates correspond to the first layer that contains a data point from the specified year.

Index). Finally, the currencies that stand out in all panels between 2007 and 2010 are "CHFUSD" (Swiss franc) and "JPYUSD" (Japanese Yen) when there are two currencies and "JPYUSD" when there is only one. In both cases, these currencies leave the equity community to join the bond community. (One can clearly see in the correlation matrix of Fig. 3.4(e,f) that "JPYUSD" is correlated strongly to bonds and correlated weakly to equities during the time intervals 25 Aug 06–18 Jul 08 and 08 Feb 08–01 Jan 10.)

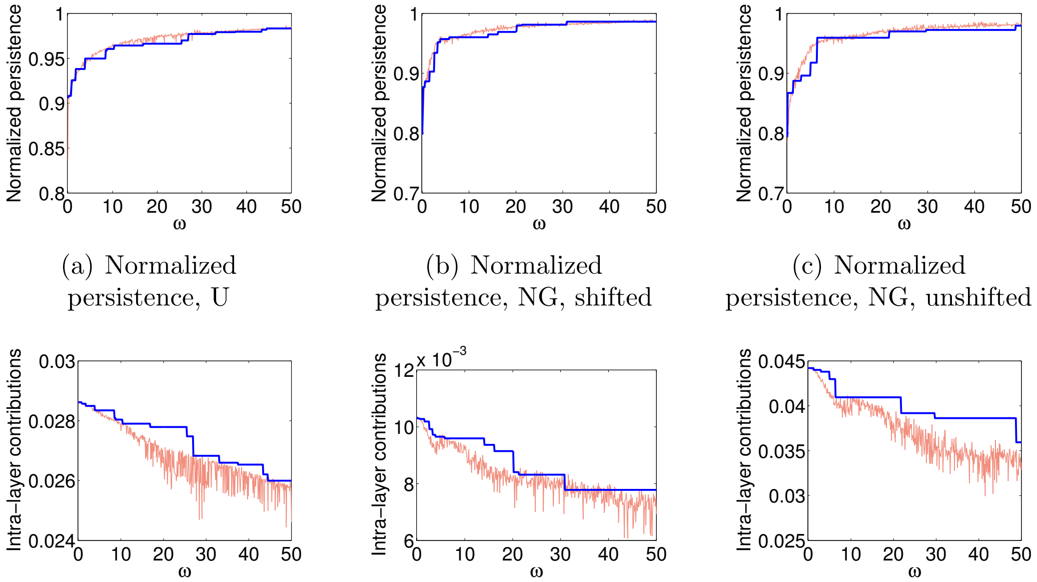
6.2 Numerical experiments with SingleAssetClass

In this section, we use LRW to identify Cv output partitions and we vary the null network for the SingleAssetClass data set. We illustrated in Chapter 4 that one needs to be cautious when interpreting communities that one obtains with a null network in which the distribution of edge weights is sample-dependent. In particular, we stated in equation (2.9) that edge weights in the NG null network are given by $P_{ij} = \text{corr}(\hat{z}_i, \hat{z}_{\text{tot}})\text{corr}(\hat{z}_j, \hat{z}_{\text{tot}})$ for a correlation matrix.[3] In the case of AssetClasses, assets in the data set belong to different classes. The quantity $\hat{z}_{\text{tot}}$ thus depends on the number and type of assets that one chooses to include in each class. As illustrated in Fig. 4.2, this can lead to misleading results when one interprets the communities that one obtains for AssetClasses with the NG null network. In the case of SingleAssetClass, assets belong to a single asset class (i.e., they are all equity assets) and constitute the S&P 1500. The quantity $\hat{z}_{\text{tot}}$ in this case represents the mean standardized time series of assets in S&P 1500. Whether or not $\text{cov}(\hat{z}_i, \hat{z}_{\text{tot}})$ (i.e., k_i) is a feature that one may wish to "control for" in a null network is less clear in this case. Accordingly, we compare some features of the partitions that we obtain with the U null network and the NG null network (which we apply to shifted and unshifted correlation matrices) for SingleAssetClass.

6.2.1 Comparisons with different null networks

In Fig. 6.4, we show plots of persistence (respectively, intra-layer modularity) for the Cv output partitions obtained with the U null network, the NG null network on a shifted correlation matrix, and the NG null network on an unshifted correlation matrix in panels (a)–(c) (respectively, panels (d)–(f)). We show two curves in all panels. Each point on the red curve is computed with the corresponding Cv output partition, and each point on the blue curve is computed with the Cv output partition that yields that largest value of multilayer modularity for the corresponding ω value. The number of iterations until convergence to the Cv output partition for all three null networks and all values of ω varies between 2 and 5 for the SingleAssetClass

[3] The edge weight of the NG null network becomes $P_{ij} = [\text{cov}(\hat{z}_i, \hat{z}_{tot}) + N][\text{cov}(\hat{z}_j, \hat{z}_{tot}) + N]/[\text{cov}(\hat{z}_{tot}, \hat{z}_{tot}) + N^2]$ for a correlation matrix that is linearly shifted to $[0, 1]$.



(a) Normalized persistence, U

(b) Normalized persistence, NG, shifted

(c) Normalized persistence, NG, unshifted

(d) Intra-layer modularity, U

(e) Intra-layer modularity, NG, shifted

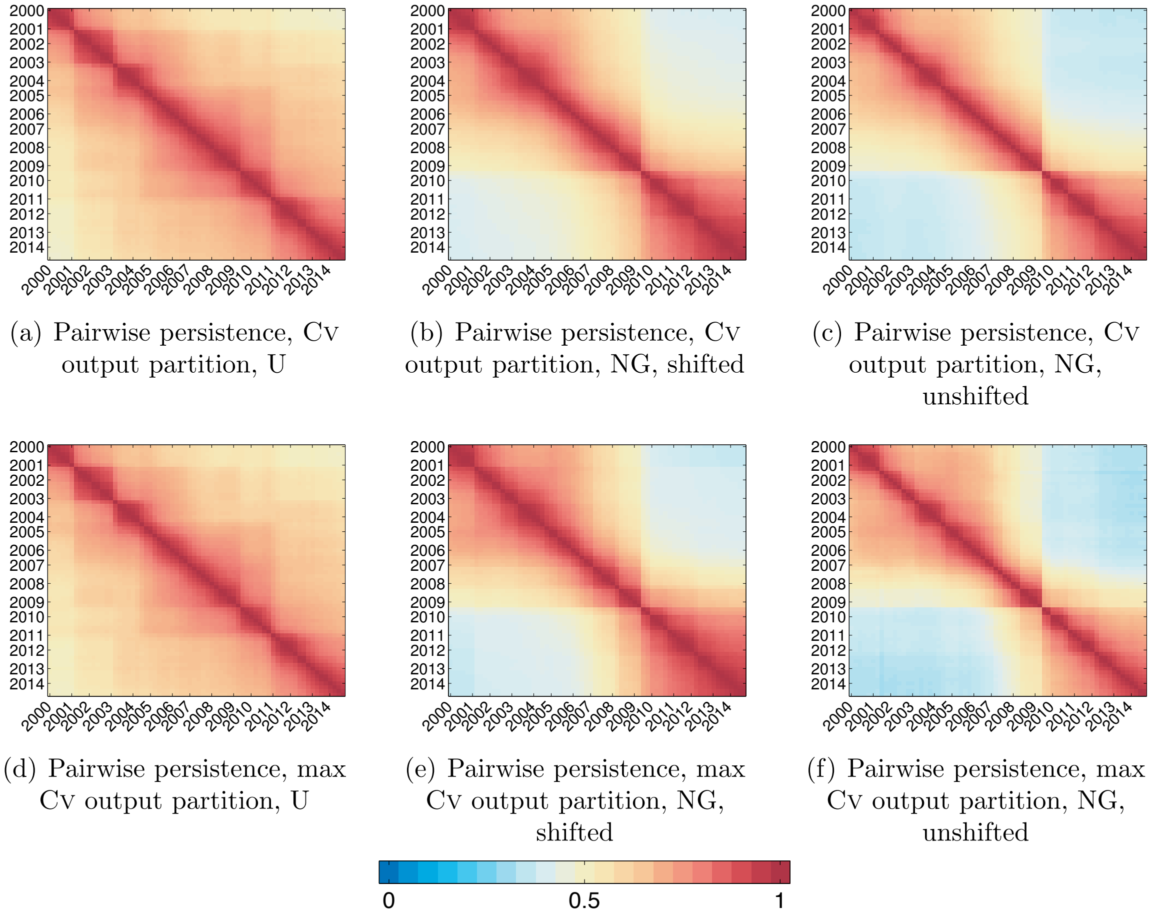
(f) Intra-layer modularity, NG, unshifted

Figure 6.4: Persistence for Cv output partitions with (a) the U null network, (b) the NG null network on a linearly shifted correlation matrix, (c) the NG null network on an unshifted correlation matrix. Intra-layer modularity for Cv output partitions with LRW for (e) the U null network, (e) the NG null network on a linearly shifted correlation matrix, and (f) the NG null network on an unshifted correlation matrix. Each point on the red curve is computed with the identified partition, and each point on the blue curve is computed with the Cv output partition that yields the largest value of multilayer modularity for the corresponding ω value amongst all identified Cv output partitions. Our sample of ω values is $\{0, 0.1, \ldots, 49.9, 50\}$, and we use $\gamma = 1$ in all cases. We normalize persistence by $N(|\mathcal{T}| - 1)$ and intra-layer modularity by $\sum_{s=1}^{|\mathcal{T}|} \sum_{i,j=1}^{N} A_{ijs}$ in all panels. We use LRW to identify all Cv output partitions.

data set. We apply our post-processing procedure from Section 4.2.3.1 to all Cv output partitions. With each null network, we store one Cv multilayer partition for each value of ω. This gives a total of 501 multilayer partitions per panel.

As we pointed out for the experiments in Fig. 6.1, the blue curves that correspond to persistence are non-decreasing functions of ω and the blue curves that correspond to intra-layer modularity are non-increasing functions of ω. The increase of persistence and the decrease of intra-layer modularity is not monotonic for the red curve, because there is a single local optimum for each ω value.

In Fig. 6.5, we show the $|\mathcal{T}| \times |\mathcal{T}|$ pairwise persistence matrix whose entries are $\big(\text{Pers}(C)|_{s,r}\big)/N$ averaged over the ω values. We compute this matrix in panels (a)–(c) with the set of Cv output partitions and in panels (d)–(f) with the set of Cv output partitions in which the k^{th} partition yields the highest value of multilayer modularity



(a) Pairwise persistence, Cv output partition, U

(b) Pairwise persistence, Cv output partition, NG, shifted

(c) Pairwise persistence, Cv output partition, NG, unshifted

(d) Pairwise persistence, max Cv output partition, U

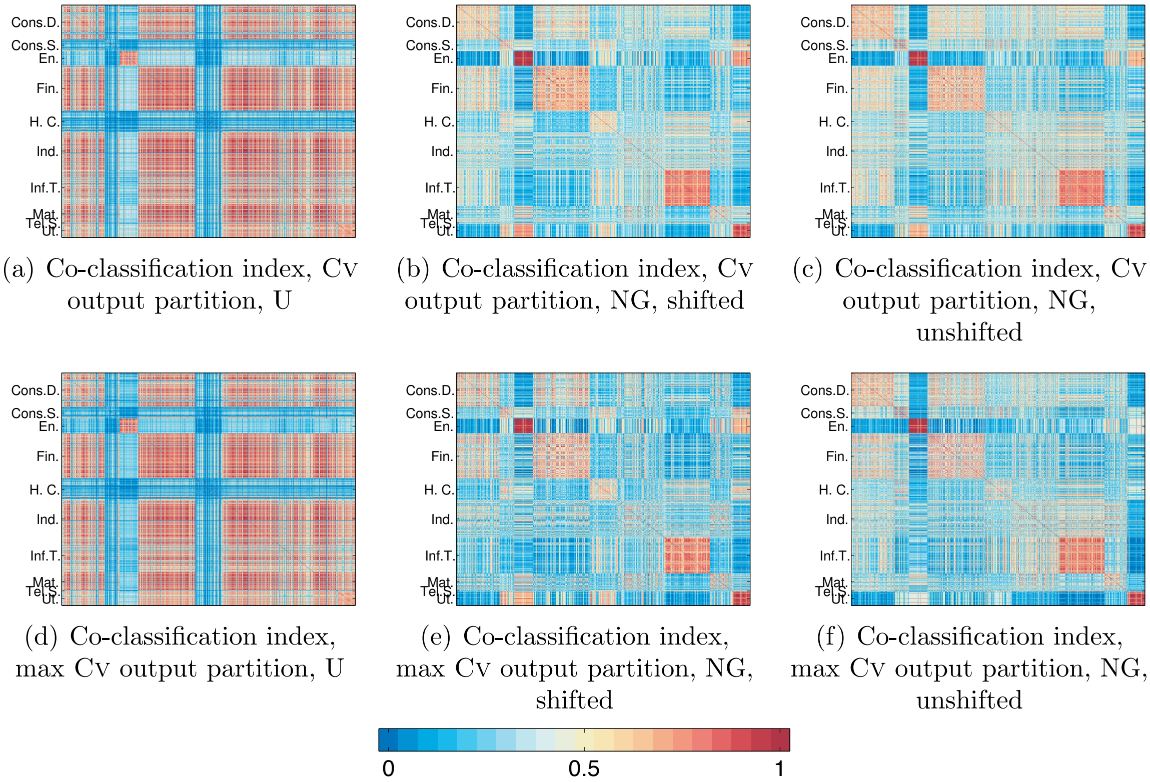
(e) Pairwise persistence, max Cv output partition, NG, shifted

(f) Pairwise persistence, max Cv output partition, NG, unshifted

Figure 6.5: We show normalized pairwise persistence (i.e., $\text{Pers}(C)|_{s,r} = \sum_{i=1}^{N} \delta(c_{i_s}, c_{i_r})/N$), averaged over $\omega \in \{0, 0.1, \ldots, 49.9, 50\}$, for Cv output partitions with (a) the U null network, (b) the NG null network on a shifted correlation matrix, and (c) the NG null network on an unshifted correlation matrix, respectively; and for Cv output partitions with highest multilayer modularity with (d) the U null network, (e) the NG null network on a shifted correlation matrix, and (f) the NG null network on an unshifted correlation matrix, respectively. We use LRW with $\gamma = 1$ in all cases. Dates correspond to the first layer that contains a data point from the specified year.

for the k^{th} value of ω in $\{0, 0.1, \ldots, 49.9, 50\}$ amongst all Cv output partitions. Note that the second set of partitions contains duplicates. In particular, the number of unique partitions is 16 in panel (d), 14 in panel (e), and 9 in panel (f).

Observe that panels (a)–(c) are visually similar to panels (d)–(f) in Fig. 6.5. This suggests that pairwise persistence computed from the set of 501 Cv output partitions is fairly consistent with pairwise persistence computed from the set of Cv output partitions with highest multilayer modularity. Moreover, panels (b,c,e,f) are also visually similar. This suggests that pairwise persistence computed with the NG null network on a shifted correlation matrix is relatively consistent with pairwise persis-



(a) Co-classification index, Cv output partition, U

(b) Co-classification index, Cv output partition, NG, shifted

(c) Co-classification index, Cv output partition, NG, unshifted

(d) Co-classification index, max Cv output partition, U

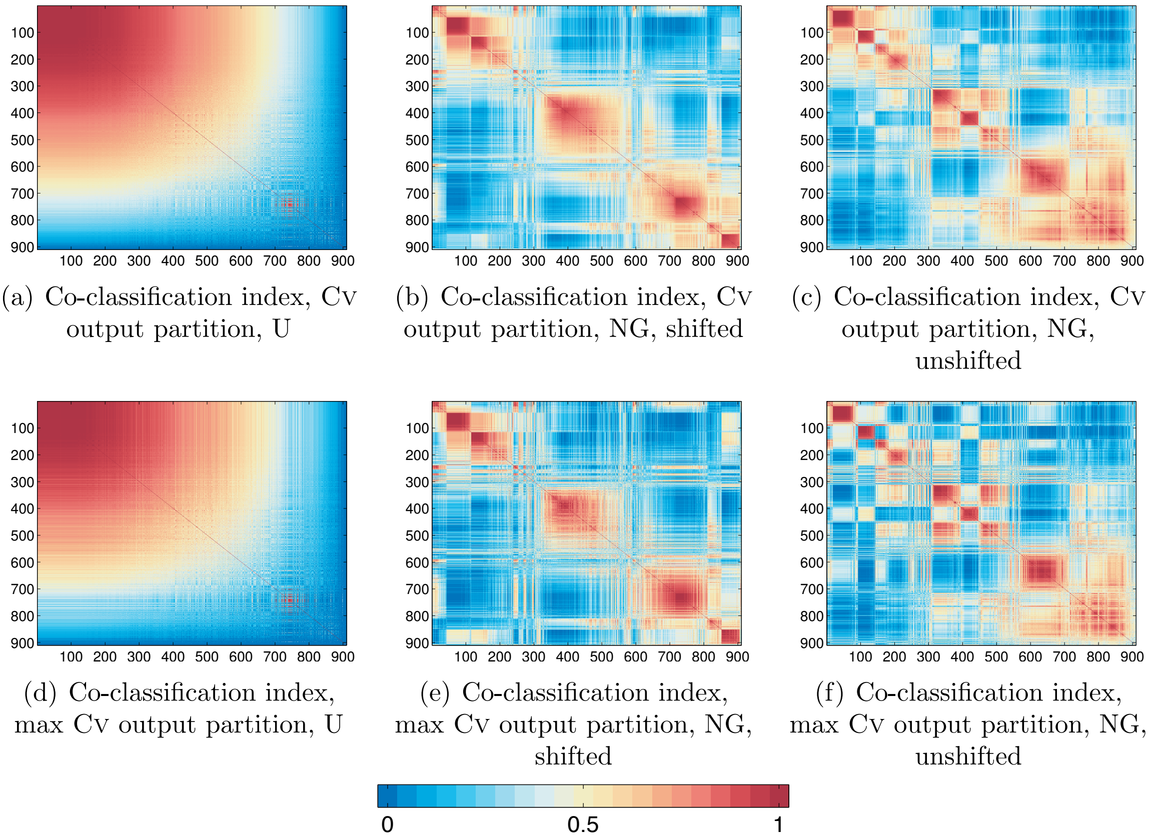
(e) Co-classification index, max Cv output partition, NG, shifted

(f) Co-classification index, max Cv output partition, NG, unshifted

Figure 6.6: We show the pairwise co-classification of nodes across layers (i.e., $\left(\sum_{s=1}^{|\mathcal{T}|} \delta(c_{i_s}, c_{j_s})\right)/|\mathcal{T}|$) averaged over $\omega \in \{0, 0.1, \ldots, 49.9, 50\}$ for Cv output partitions with (a) the U null network, (b) the NG null network on a shifted correlation matrix, and (c) the NG null network on an unshifted correlation matrix; and for the Cv output partitions with highest multilayer modularity with (d) the U null network, (e) the NG null network on a shifted correlation matrix, and (f) the NG null network on an unshifted correlation matrix. We use LRW with $\gamma = 1$ in all cases. We reorder nodes in all matrices by sector classification.

tence computed with the NG null network on an unshifted correlation matrix with SingleAssetClass. In panels (a,d) obtained with the U null network, pairwise persistence is relatively high between all pairs of layers. The highest value in these panels 1 and the lowest value is 0.5. In panels (b,c,e,f), the most noticeable separation between diagonal blocks occurs between 2009 and 2010, near the layer that corresponds to the time interval 18 Aug 08–27 Aug 09 (i.e., the first layer that contains a data point near the middle of 2009).

In Figs 6.6 and 6.7, we show the pairwise co-classification index $\left(\sum_{s=1}^{|\mathcal{T}|} \delta(c_{i_s}, c_{j_s})\right)/|\mathcal{T}|$ of nodes across layers averaged over ω. In both figures, we compute the top three panels with the set of Cv output partitions and the bottom three panels with the set of Cv output partitions in which the k^{th} partition yields the highest value of multi-



(a) Co-classification index, Cv output partition, U

(b) Co-classification index, Cv output partition, NG, shifted

(c) Co-classification index, Cv output partition, NG, unshifted

(d) Co-classification index, max Cv output partition, U

(e) Co-classification index, max Cv output partition, NG, shifted

(f) Co-classification index, max Cv output partition, NG, unshifted

Figure 6.7: We show the pairwise co-classification of nodes across layers (i.e., $\left(\sum_{s=1}^{|\mathcal{T}|} \delta(c_{i_s}, c_{j_s})\right) / |\mathcal{T}|$), averaged over $\omega \in \{0, 0.1, \ldots, 49.9, 50\}$, for Cv output partitions with (a) the U null network, (b) the NG null network on a shifted correlation matrix, and (c) the NG null network on an unshifted correlation matrix; and for the Cv output partitions with highest multilayer modularity with (d) the U null network, (e) the NG null network on a shifted correlation matrix, and (f) the NG null network on an unshifted correlation matrix. We use LRW with $\gamma = 1$ in all cases. We use a node order that emphasizes block-diagonal structure in each matrix. We use the same ordering in each of the three pairs (a,d), (b,e), and (c,d) (but a different ordering for different pairs).

layer modularity for the k^{th} value of ω in $\{0, 0.1, \ldots, 49.9, 50\}$ amongst all Cv output partitions. In Fig. 6.6, we reorder all matrices using the sector classification of nodes. In Fig. 6.7, we reorder all matrices using a node ordering that we obtain by applying the U null network to each matrix. This ordering can emphasize block diagonal structure in a matrix. We use the same node order in Figs 6.7(a,d), Figs 6.7(b,e), and Figs 6.7(c,f) (but a different node order for different pairs). We observe that matrices in the top three panels of Fig. 6.6 (respectively, Fig. 6.7) are visually similar to matrices in the bottom three panels of Fig. 6.6 (respectively, Fig. 6.7).

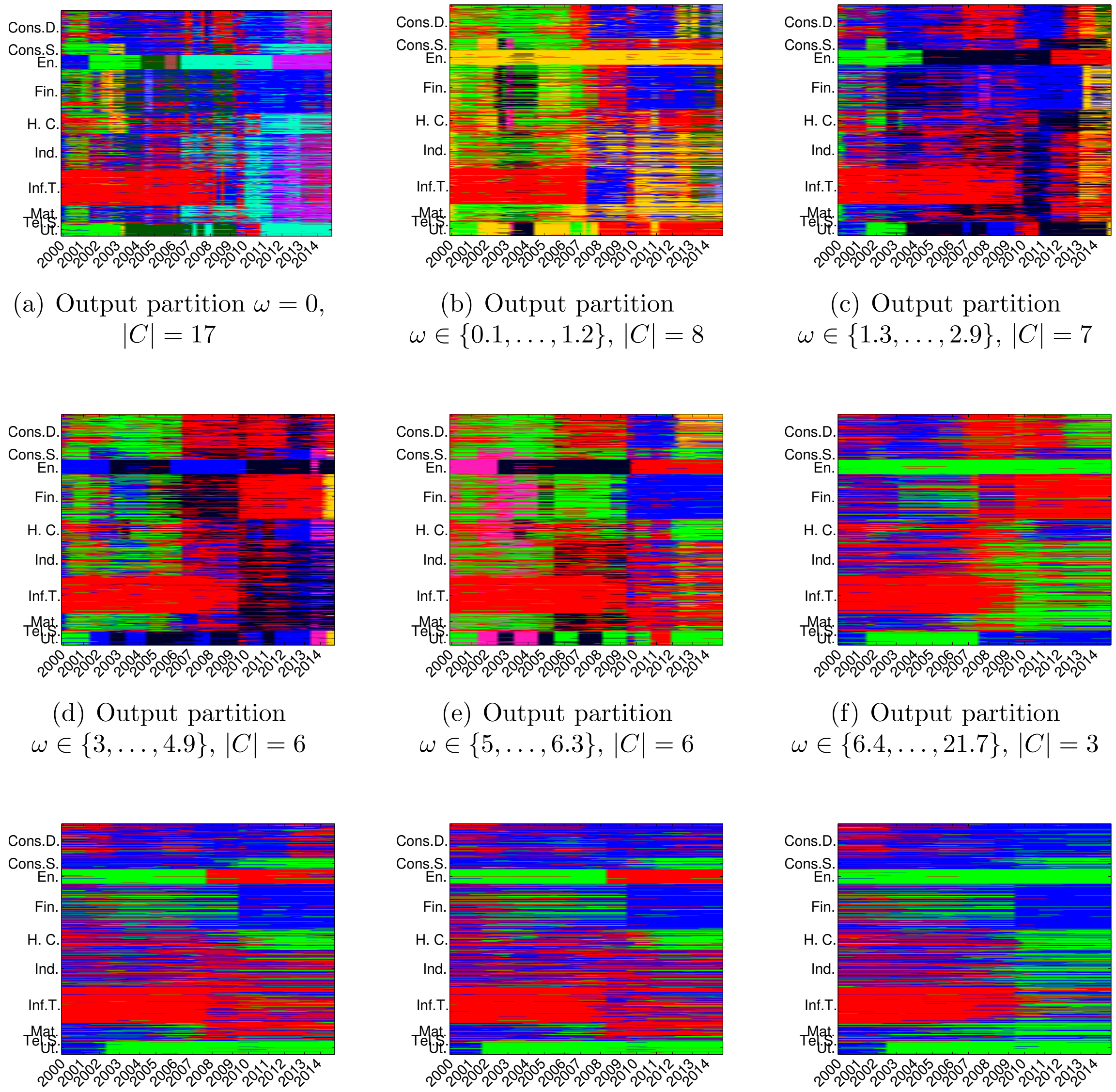
In Figs 6.6(a,d), my matching diagonal and off-diagonal blocks to their corresponding sector(s), we observe that both inter-sector and intra-sector average co-classification indices seem to be high for all sectors except for consumer staples (Cons. S.), energy (En.), and healthcare (H.C.). For the energy sector, only intra-sector co-classification indices seem to be high. For consumer staples and health care, neither intra-sector nor inter-sector co-classification indices seem to be high. (See Figs. 3.6(b–f) for example correlation matrices where inter-sector correlation is lower for consumer staples, energy, and healthcare than it is for other sectors.) In Figs 6.6(b,c,e,f), we observe that assets from the six following sectors have a relatively high co-classification index: consumer discretionary (Cons. D), consumer staples (Cons. S.), energy (En.), financials (Fin.), information technology (Inf. T.), and utilities (Ut.) The four sectors with highest intra-sector average co-classification index in Figs 6.6(b,c,e,f) are visibly energy, utilities, information technology, and financials.

In Fig. 6.7, we clearly see features in each panel that do not appear in its corresponding panel of Fig. 6.6. (Recall that we use the same set of matrices to generate Figs. 6.6 and 6.7.) For example, there are more red diagonal blocks in Fig. 6.7(b,c,e,f) than there are in Fig. 6.6(b,c,e,f). This suggests that community membership of nodes is not based solely on sector classification. We briefly comment on Fig. 6.7(a,d). Observe that Fig. 6.7(a,d) has a large "core" set of assets that often belong to the same community (i.e., roughly nodes 1–600), many assets that sometimes belong to the same community as the core (i.e., roughly nodes 600–750), and many assets that do not have a high co-classification index with the core (i.e., roughly nodes 750–900). In particular, the number of communities in the set of Cv partitions obtained with the U null network is of the order of 10^2 for all values of ω. In the multilayer partition that corresponds to $\omega = 0$ (recall that we maximize persistence for each Cv output partition with our post-processing procedure), the number of communities is 821, the largest community contains approximately 66.6% of the $N|\mathcal{T}|$ nodes, and the second largest community contains approximately 4.4% of the $N|\mathcal{T}|$ nodes. For the partition that we obtain with $\omega = 50$ (the largest value in our sample), the number of communities is 196, the largest community contains approximately 72.9% of the $N|\mathcal{T}|$ nodes, and the second largest community contains approximately 1.5% of the $N|\mathcal{T}|$ nodes. This suggests that multilayer partitions with the U null network for SingleAssetClass contain one large community, to which most nodes in the multilayer network belong, and many smaller communities. (This observation is consistent with the correlation values in Figs 3.6(e,f) and Fig. 3.3(b), where we observed that many of the equity assets are highly correlated.)

6.2.2 Output multilayer partitions with increasing coupling

As example output multilayer partitions, we show Cv output partitions that we obtain with the NG null network on an unshifted correlation matrix in Fig. 6.8. That is, we show the nine Cv output partitions that correspond to the nine "plateaus" in Figs 6.4(c,f).

The Cv output partition that yields the largest value of multilayer modularity (amongst identified partitions) for $\omega = 0$ does not yield the largest value of multilayer modularity for any other value of ω in our sample. This suggests that there is a trade-off between persistence $\text{Pers}(C)$ and intra-layer modularity $Q(C|\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{|\mathcal{T}|}; 0)$ starting at $\omega = 0.1$ in our sample of ω values. This was not the case for AssetClasses in Fig. 6.3. Moreover, most of the assets from the energy, utilities, information technology, and financials sectors belong to the same community in several layers of all output partitions. This is consistent with our observations in Section 6.2.1, where we noted that these sectors had the highest mean inter-sector co-classification index in Figs 6.6(b,c,e,f). (Recall that we average the co-classification index over time and ω values in these figures.) The clear diagonal-block separation between 2009 and 2010 in Figs 6.5(b,c,e,f) manifests in all panels of Fig. 6.8. The largest number of assets from the financials sector that belong to the same community seems to increase between 2009 and 2010. Additionally, the largest number of assets from the information technology sector that belong to the same community seems to decrease between 2009 and 2010. We also observe that the largest number of assets from the health care sector that belong to the same community seems to increase after 2010. (A notable event that took place on 23 March 2010 is Obama's Health Care Reform.) We show the largest number of assets from each of the three sectors that belong to the same community across time in Fig. 6.9. In each panel, we normalize the number of assets by the size of each sector.



(a) Output partition $\omega = 0$, $|C| = 17$

(b) Output partition $\omega \in \{0.1, \ldots, 1.2\}$, $|C| = 8$

(c) Output partition $\omega \in \{1.3, \ldots, 2.9\}$, $|C| = 7$

(d) Output partition $\omega \in \{3, \ldots, 4.9\}$, $|C| = 6$

(e) Output partition $\omega \in \{5, \ldots, 6.3\}$, $|C| = 6$

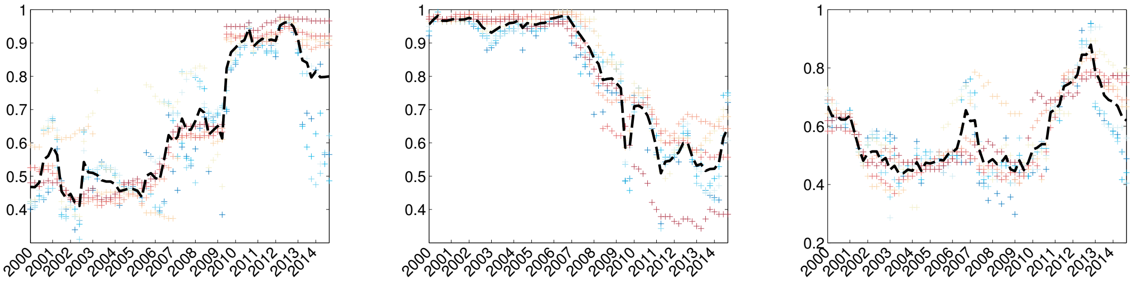
(f) Output partition $\omega \in \{6.4, \ldots, 21.7\}$, $|C| = 3$

(g) Output partition $\omega \in \{21.8, \ldots, 29.6\}$, $|C| = 3$

(h) Output partition $\omega \in \{29.7, \ldots, 48.7\}$, $|C| = 3$

(i) Output partition $\omega \in \{48.8, \ldots, 50\}$, $|C| = 3$

Figure 6.8: Multilayer Cv output partitions with highest multilayer modularity for the values of ω in the caption. We use LRW and the NG null network (i.e., $P_{ijs} = (k_{i_s} k_{j_s})/(2m_s)$) on an unshifted correlation matrix (i.e., $\boldsymbol{A}_s \in [-1, 1]^{N \times N}$) with $\gamma = 1$. Recall that $\omega \in \{0, 0.1, \ldots, 49.9, 50\}$ is our sample of ω values. Colors correspond to community assignments, and dates correspond to the first layer that contains a data point from the specified year.



(a) Largest fraction of Fin. that belong to the same community in each layer

(b) Largest fraction of Inf.T. that belong to the same community in each layer

(c) Largest fraction of H.C. that belong to the same community in each layer

Figure 6.9: In each panel, we show nine scatter plots (one for each of the nine Cv output partitions of Fig. 6.8) of the largest fraction of assets from a given sector that belong to the same community in a given layer, and we show the mean value of the nine scatter plots computed at each layer (black dashed curve). We show these plots for (a) financials (Fin.), (b) information technology (Inf. T.), and (c) healthcare (H.C.). We normalize the values in each panel by the size of the corresponding sector. Recall from Section 3.3.2 that there are 177 Fin., 140 Inf. T., and 84 H.C assets. Colors of the scatter plots in each panel scale from deep blue to deep red with the range of ω values for which the corresponding Cv output partition yields the largest value of multilayer modularity. That is, the scatter plot that corresponds to the range with lowest values (i.e., $\omega = 0$ in Fig. 6.8) is in dark blue, and the scatter plot that corresponds to the range with highest values (i.e., $\omega \in \{48.8, 48.9, \ldots, 49.9, 50\}$ in Fig. 6.8) is in bright red. Dates correspond to the first layer that contains a data point from the specified year.

6.3 Summary

We carried out various experiments with the data sets AssetClasses and SingleAssetClass in which we examined persistence (see (4.3)), intra-layer modularity (see (4.4)), pairwise persistence (see (4.5)) averaged over increasing values of inter-layer coupling, co-classification indices (see Section 2.3.2) averaged over time and increasing values of inter-layer coupling, and example multilayer output partitions. In our experiments, we tried to illustrate three main points: (1) communities that we identified with multilayer modularity maximization were not based solely on class or sector classification, (2) we obtained fairly consistent results in the experiments that we carried out with different variants of the same heuristics and with different sets of output partitions for a given variant of a heuristic, and (3) the output partitions that we obtained with higher values of inter-layer coupling differed from the partitions that we obtained with 0 inter-layer coupling (reflecting a trade-off between persistence and intra-layer modularity). From a practical viewpoint, some features of the output partitions obtained with AssetClasses are consistent with observations drawn from the same data set with other techniques (e.g., post-Lehman "bi-modal structure" [65,200]), whereas other features seem to highlight periods in time that are different from those previously noticed and that may potentially provide additional insight (e.g., the change in community structure between 2005 and 2006). We also illustrated with SingleAssetClass that the ability to explicitly specify the choice of null network can enable one to extract higher-order structure from a correlation matrix in which, *a priori*, most assets are highly correlated. For both data sets, we pointed out various instances in which a change in community structure in an output partition corresponds to a change in connectivity patterns in the correlation matrices.

Chapter 7

Discussion and Future Work

In this thesis, we investigated various issues associated with community detection in temporal networks. We began the thesis with an overview of community-detection methods for time-independent and temporal networks in Chapter 2. Most community-detection methods are designed for time-independent networks, and multilayer networks is a recent framework for representing a sequence of networks that includes connections at different points in time and/or different types of connections at a given point in time [31,109]. We highlighted recent efforts towards developing a community-detection method for multilayer networks based on the optimization of a quality function. In particular, we examined multilayer modularity maximization [140] and multilayer map minimization [56]. Two key differences between these methods and their single-layer counterpart are that they produce a multilayer partition (as opposed to a single-layer partition) and, more importantly, that communities identified in individual layers depend on connectivity patterns in other layers. A practical advantage of methods based on the optimization of a quality function (in contrast to, say, multilayer block modeling approaches [158, 160, 186]) is that they do not require a model of the observed network and they require no *a priori* estimate of the number of communities in the final partition. A disadvantage, however, is that they have less theoretical grounding. Coupled with the many other existing issues in community detection, this can render the interpretation of communities that one obtains with these methods all the more difficult.

In Chapter 4, we tried to better understand how one can interpret communities with multilayer modularity maximization. We examined two main issues: (1) the role of null networks and (2) the role of inter-layer coupling. In section 4.1, we illustrated with financial asset-correlation networks that one needs to choose a null network with caution because it ultimately determines what one regards as densely connected in a network. We showed that this choice can lead to misleading conclusions when one uses a null network in which the distribution of edge weights is sample-dependent. In the following paragraph, we will make a few observations on the potential benefits associated with the flexibility of this choice. The ability to explicitly specify what is anticipated (e.g., by setting the mean expected edge weight in a null network equal to the mean edge weight in an observed network) is an appealing feature of modularity maximization. One can adapt it to different applications and to different questions for a given application. This feature of modularity maximization has been explored in the context of spatial correlation networks to take into account prior knowledge that neighboring countries tend to be highly correlated with respect to the spread of a disease [61, 175]. More recently, the authors in [131] suggested a null network for correlation networks (though we note that this is not how the authors phrase it) that only depends on the leading eigenvector and leading eigenvalue of a sample correlation matrix (i.e., the "market mode" in random matrix theory [162, 164]). The intuition behind this null network is that it controls for a "common factor influencing all stocks within a given market" [131]. It would be interesting to compare the partitions that one obtains with this null network to those one obtains with the Newman–Girvan null network on the equity data set (i.e., SingleAssetClass), especially in light of the interpretation of asset strength as the covariance between a standardized time series and the mean standardized time series. One could also try to incorporate "ground truth" about a data set into a null network. For example, one could include prior knowledge that assets from the same asset class tend to be highly correlated by taking the expected edge weight between two assets to be the mean computed from the corresponding assets classes (instead of the full correlation matrix).

In the context of correlation networks, for any choice of null network that one makes, it is important that one's estimate of a correlation matrix is reliable. In this thesis, we used Pearson correlation because it is a simple and commonly used estimate of pairwise correlation [63, 65, 134, 154]. Estimating a correlation matrix from a set of empirical time series is an active (and unresolved) research area that warrants further investigation (e.g., one can try to address issues of non-stationarity [178], one can estimate the true covariance matrix from a sample covariance matrix by solving a maximum likelihood problem [51], and one can explore the issue of choosing an appropriate time window length relative to the number of assets in a data set [65, 153, 164]).

Given a choice of observed network and null network (or more generally, given a choice of matrices $\{\boldsymbol{B}_1, \dots, \boldsymbol{B}_{|\mathcal{T}|}\}$), we explored the effect of inter-layer edges on an optimal multilayer partition in Section 4.2. We showed that multilayer modularity maximization reflects a trade-off between intra-layer modularity (i.e., time-independent community structure) and temporal persistence. The largest achievable value of persistence for an optimal partition obtained with a given value of inter-layer coupling is a non-decreasing function in ω. Similarly, the largest achievable value of intra-layer modularity for an optimal partition obtained with a given value of inter-layer coupling is a non-increasing function in ω. Moreover, we showed that there exists a value $\omega_0 > 0$ below which an optimal multilayer partition is also optimal for $\omega = 0$. Similarly, there exists a value ω_∞ above which persistence is maximal and multilayer modularity maximization reduces to single-layer modularity maximization on the mean single-layer modularity matrix. These results provide some information on an optimal multilayer partition in the extreme cases $\omega < \omega_0$ and $\omega > \omega_\infty$, but it would be useful to further explore whether something can be said about how intra-layer modularity decreases and how persistence increases between these bounds. Although this seems nontrivial (for example, one can easily design toy examples in which these quantities vary very differently across different examples), it would be useful to gain some insight on this point that either applies to any choice of single-layer modularity matrices, or that is at least applicable to a large set of (non-pathological) single-layer modularity matrices. Such insight could shed light on whether there exists an interval (or multiple intervals) of ω values between ω_0 and ω_∞ in which the trade-off between persistence and intra-layer modularity yields the most insights.

Another issue in multilayer community-detection that warrants further investigation is that of interpretability. As we mentioned in Chapter 5, multilayer community-detection inherits all the issues of single-layer community detection with (at least) one important additional one: what does "community" mean intuitively in a multilayer network? In particular, should one interpret nodes in the same community from the same layer differently from nodes that are in the same community but belong to different layers? One would think that an answer to this question should depend, at least partly, on how one defines inter-layer edges in a multilayer network. We make a few observations to illustrate that this point is rather nuanced. Suppose that one uses cross-correlation matrices (with lags $1, \ldots, |\mathcal{T}| - 1$) on the off-diagonal blocks of the multilayer adjacency matrix in (3.8) instead of uniform, ordinal, and diagonal inter-layer coupling. It would be sensible, for such a choice, to use a null network on the off-diagonal blocks (much in the same way as one does for the diagonal blocks) and potentially to interpret communities within and between layers in similar ways. In this case, both an inter-layer edge weight and an intra-layer edge weight reflect a correlation computed from a pair of time series. Furthermore, for this nonuniform choice of coupling, one cannot infer the inter-layer community assignments of an optimal partition knowing only the set of partitions that it induces on individual layers. Given an optimal partition's set of intra-layer partitions, its inter-layer assignments depend on inter-layer edge weights (here given by cross-correlations). In contrast, if one uses uniform, ordinal, and diagonal inter-layer coupling, one can infer inter-layer community assignments in an optimal partition knowing only the set of partitions it induces on individual layers. That is, given the set of intra-layer partitions of an optimal multilayer partition (for some choice of ω), inter-layer community assignments can be inferred by maximizing persistence (which, as defined, is independent of inter-layer edge weights). In some sense, the information in a multilayer partition obtained with uniform, ordinal, and diagonal inter-layer coupling is fully contained in the sequence of intra-layer partitions induced on individual layers (i.e., one can infer the former from the latter). When $\omega = 0$, elements of this sequence are uncorrelated, and when $\omega \neq 0$, elements of this sequence are correlated and reflect a trade-off between intra-layer modularity and persistence. Both the non-uniform choice and the uniform choice of inter-layer coupling that we described in this paragraph yield multilayer partitions, but one's interpretation of multilayer communities (although clear in neither case) seems like it should differ for each choice.

We proposed a simple benchmark that generates a sequence of correlated single-layer partitions in Chapter 5. In this benchmark, nodes in each layer are independent and every node keeps the same community assignment between consecutive layers with probability p. Nodes that are unassigned in each layer get assigned to a community at random via a null distribution specified for that layer. When $p = 0$, this benchmark generates a set of uncorrelated single-layer partitions. When $p \neq 0$, this benchmark generates a set of correlated single-layer partitions. We generate a benchmark network using a stochastic block model for the set of partitions induced on the set of layers. We used a simple model in Chapter 5, but there are various next steps that one could explore to make the benchmark more realistic. For example, one could consider non-uniform choices for the persistence parameter p and define a $(|\mathcal{T}| - 1)$-dimensional vector $\boldsymbol{p}$. Instead of having multilayer benchmark partitions in which the expected number of nodes that change communities is (potentially) nonzero between any pair of consecutive layers, this can produce benchmark partitions with chains of consecutive layers within which structure does not change but between which structure does change (similarly to the benchmark in [56]). One could also introduce memory effects (e.g., a node in layer s keeps its community assignment from layer $s-1$ with probability p_1 and its community assignment from layer $s-2$ with probability p_2) and introduce dependencies between edges in different layers (e.g., the probability of an edge between i_{s+1} and j_{s+1} depends on whether there is an edge between i_s and j_s).

In our experiments, we compared the behaviour of different heuristics for solving modularity maximization on our benchmark and carried out preliminary experiments with multilayer Infomap [1]. We showed that a simple benchmark can highlight features of heuristics and methods that do not manifest as clearly when one performs experiments with real data. Such observations can provide useful guidelines regarding what choices (e.g., choice of a heuristic or parameter choices for a given method) may be more or less appropriate in different situations. They also emphasize a need for further theoretical grounding of both methods and heuristics in community detection. From a computational viewpoint, it seems that what one wants is a heuristic that (ideally) satisfies three conditions: (1) it scales well with the size of a network, (2) it produces sensible ouput partitions, and (3) is has some theoretical guarantee on how closely an identified partition resembles an optimal partition. Potentially interesting directions to pursue on this front is whether it is possible to modify the Louvain algorithm in such a way that it allows modularity-decreasing moves (to mitigate its tendency to get trapped in local optima) without substantially increasing its computational run-time. One could also investigate whether using more eigenvectors in a spectral heuristic improves the quality of a multilayer output partition [209]. Our observations in Section 4.2.3.2 suggest that one may need at least one eigenvector for every distinct layer. There are also other heuristics in the literature that would be interesting to investigate and that one might be able to extend to multilayer modularity maximization [92, 112, 149, 206].

The issues discussed in this thesis constitute a small part of many unresolved questions in community detection, a sub-area of network analysis that itself falls under the broad umbrella of "data analysis". Someone once told me that "network analysis is about giving fuzzy answers to fuzzy questions, but these answers can be useful". In some sense, this is a fairly accurate description of much of the work in the field. Network analysis is a blend of ideas that come from physicists, computer scientists, mathematicians, sociologists, engineers, and others. It is not surprising that a relatively recent interdisciplinary research area still lacks precision and rigour. Moreover, any area that deals primarily with data analysis is bound to be interdisciplinary: "analyzing data" involves different steps (e.g., developing methods, developing algorithms, and developing visualization tools) and "interpreting data" involves diverse expertise (e.g., sociology, finance, and biology). In a day and age where technology — and thus data — pervades most aspects of our lives, one can expect interdisciplinary research to continue to grow. It is clear that such research would benefit greatly from further industry input and from the input of related and more established fields in mathematics. The sooner we bridge these gaps, the sooner interdisciplinary research can fulfill its potential.

Appendix A

Details of Financial Assets

Ticker	Asset class	Description
AUGATR	Govt. bonds	Australia
CAGATR	Govt. bonds	Canada
JNGATR	Govt. bonds	Japan
NZGATR	Govt. bonds	New Zealand
USGATR	Govt. bonds	U.S.
ATGATR	Govt. bonds	Austria
BEGATR	Govt. bonds	Belgium
DEGATR	Govt. bonds	Denmark
FIGATR	Govt. bonds	Finland
FRGATR	Govt. bonds	France
GRGATR	Govt. bonds	Germany
IEGATR	Govt. bonds	Ireland
ITGATR	Govt. bonds	Italy
NEGATR	Govt. bonds	Netherlands
NOGATR	Govt. bonds	Norway
PTGATR	Govt. bonds	Portugal
SPGATR	Govt. bonds	Spain
SWGATR	Govt. bonds	Sweden
SZGATR	Govt. bonds	Switzerland
UKGATR	Govt. bonds	U.K.
MOODCBAA	Corp. bonds	Moody's BAA rated
MOODCAAA	Corp. bonds	Moody's AAA rated
MOODCAA	Corp. bonds	Moody's AA rated
MOODCA	Corp. bonds	Moody's A rated

Table A.1: Government bonds and corporate bonds in AssetClasses data set.

Ticker	Asset class	Description
GDUEEGFL	Equities	Emerging markets: Latin America
GDDUEMEA	Equities	Emerging markets: Europe, Middle East, Africa
GDUEEGFA	Equities	Emerging markets: Asia
SPX	Equities	Standard and Poors 500 (U.S.)
INDU	Equities	Dow Jones Industrial Average Index (U.S.)
NDX	Equities	NASDAQ 100 Index (U.S.)
RTY	Equities	Russell 2000 Index (U.S.)
NKY	Equities	Nikkei 225 Index (Japan)
HSI	Equities	Hang Seng Index (Hong Kong)
AS30	Equities	Australian All Ordinaries Index
NZSE	Equities	New Zealand All Ordinaries Index
SPTSX	Equities	S&P/Toronto Stock Exchange Index
UKX	Equities	FTSE 100 Index (U.K.)
DAX	Equities	DAX Index (Germany)
CAC	Equities	CAC 40 Index (France)
IBEX	Equities	IBEX 35 Index (Spain)
FTSEMIB	Equities	FTSE MIB Index (Italy)
SMI	Equities	Swiss Market Index
BVLX	Equities	PSI General Index (Portugal)
ISEQ	Equities	Irish Overall Index
BEL20	Equities	BEL 20 Index (Belgium)
KFX	Equities	OMX Copenhagen 20 Index
HEX	Equities	Helsinki Stock Exchange General Index
OBX	Equities	OBX Stock Index (Norway)
OMX	Equities	OMX Stockholm 30 Index
ATX	Equities	Austrian Traded Index
ASE	Equities	Athens Stock Exchange General Index
AEX	Equities	AEX Index (Netherlands)
AUDUSD	Currencies	Australian dollar
EURUSD	Currencies	Euro
GBPUSD	Currencies	Pounds sterling
CADUSD	Currencies	Canadian dollar
CHFUSD	Currencies	Swiss franc
JPYUSD	Currencies	Japanese yen
NOKUSD	Currencies	Norwegian krone
SEKUSD	Currencies	Swedish krona
NZDUSD	Currencies	New Zealand dollar
ZARUSD	Currencies	South African rand
IDRUSD	Currencies	Indonesian rupiah
PHPUSD	Currencies	Philippines peso
MXNUSD	Currencies	Mexican peso
CZKUSD	Currencies	Czech koruna
KRWUSD	Currencies	Korean won

Table A.2: Equities and currencies in AssetClasses data set.

Ticker	Asset class	Description
XAU	Metals	Gold
XAG	Metals	Silver
XPT	Metals	Platinum
XPD	Metals	Palladium
LA1	Metals	Aluminium
HG1	Metals	Copper
LT1	Metals	Tin
LL1	Metals	Lead
LN1	Metals	Nickel
CL1	Fuels	Crude oil, WTI
CO1	Fuels	Crude oil, brent
HO1	Fuels	Heating oil
NG1	Fuels	Natural gas
C 1	Commodities	Corn
CC1	Commodities	Cocoa
CT1	Commodities	Cotton
JO1	Commodities	Orange juice
KC1	Commodities	Coffee
LB1	Commodities	Lumber
LC1	Commodities	Live cattle
LH1	Commodities	Lean hogs
PB1	Commodities	Frozen pork bellies
RR1	Commodities	Rough rice
S 1	Commodities	Soybean
QW1	Commodities	Sugar
SM1	Commodities	Soybean meal
W 1	Commodities	Wheat
O 1	Commodities	Oats
BO1	Commodities	Soybean oil
FC1	Commodities	Feeder cattle
JN1	Commodities	Rubber

Table A.3: Metals, fuels, and commodities in AssetClasses data set.

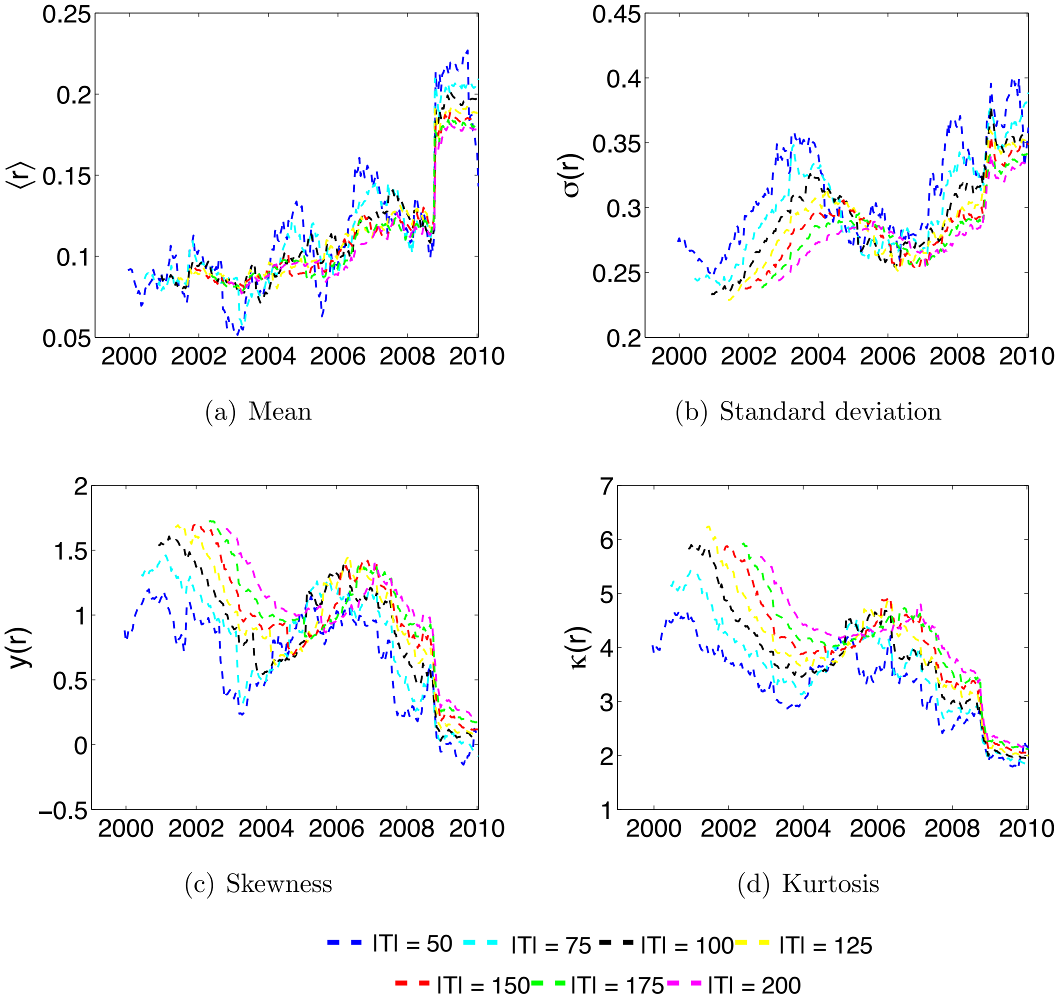
Appendix B

Robustness of Correlations to Time Window Length and Overlap

In this section we carry out numerical experiments to investigate the effect of window length (denoted $|T|$) and step size between consecutive time windows (denoted δt) on the sequence of sample correlation matrices $\{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_s | s = 1, \ldots, |\mathcal{T}|\}$ defined in Section 3.1.3. Figure B.1 shows the mean, standard deviation, kurtosis, and skewness of each correlation matrix in the sequence for time window lengths between 50 weeks (roughly one year of data) and 200 weeks (roughly four years of data). Diagonal entries are excluded from the computation, and each point on each curve in panels (a,b,c,d) of Fig. B.1 corresponds to one of the statistics computed for the first time window that contains a data point from the year on the horizontal axis. For every choice of time window length, we observe that correlations undergo various changes in time. The most notable change is around the end of 2008, where the mean correlation significantly increases for all window lengths considered. We show a frequency plot of all correlation values in the sequence in Fig. B.2(a) and a table of summary statistics averaged over all time windows in Fig. B.2(b). The average mean and variance appear to be relatively robust to the values of time window length considered. The maximum of the absolute value of the differences is of the order of 10^{-3} for the former and 10^{-2} for the latter.

We repeat the same experiments for values of δt ranging from 1 week (the smallest possible step size for this data set) to 16 weeks (roughly 4 months). We observe in Figs. B.3 and B.4 that statistics computed at each point in time and statistics averaged over time seem robust to values of the step size in this range of δt.

We perform similar experiments for the SingleAssetClass data set for values of $|T|$ ranging from 200 days (ten months excluding weekends) to 450 days (slightly under two years excluding weekends) in Figs. B.5 and B.6 and for values of δt ranging from



(a) Mean

(b) Standard deviation

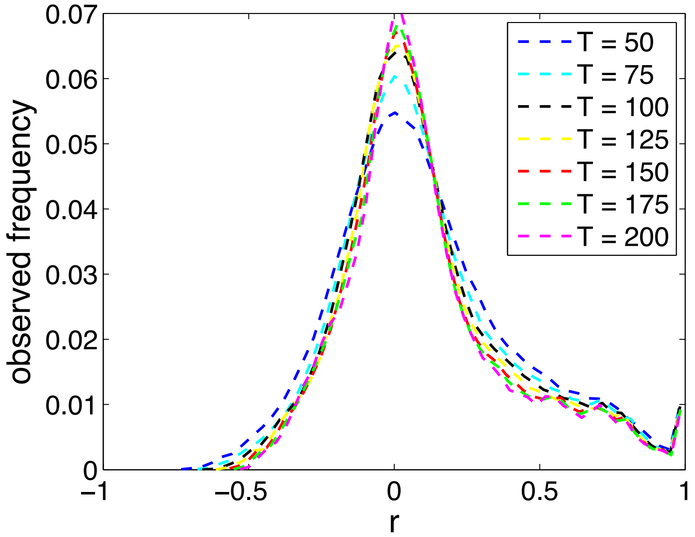
(c) Skewness

(d) Kurtosis

Figure B.1: Comparison of basic correlation statistics in AssetClasses data set over time for different time window lengths.

22 days (one month excluding weekeds) to 132 days (six months excluding weekends) in Figs. B.7 and B.8. We obtain similar quantitative results regarding robustness for the range of values considered.

This results in Figs. B.1–B.8 suggests that various features of the correlation matrices are relatively robust to the different samples of window length $|T|$ and step size δt that we consider in this Section.

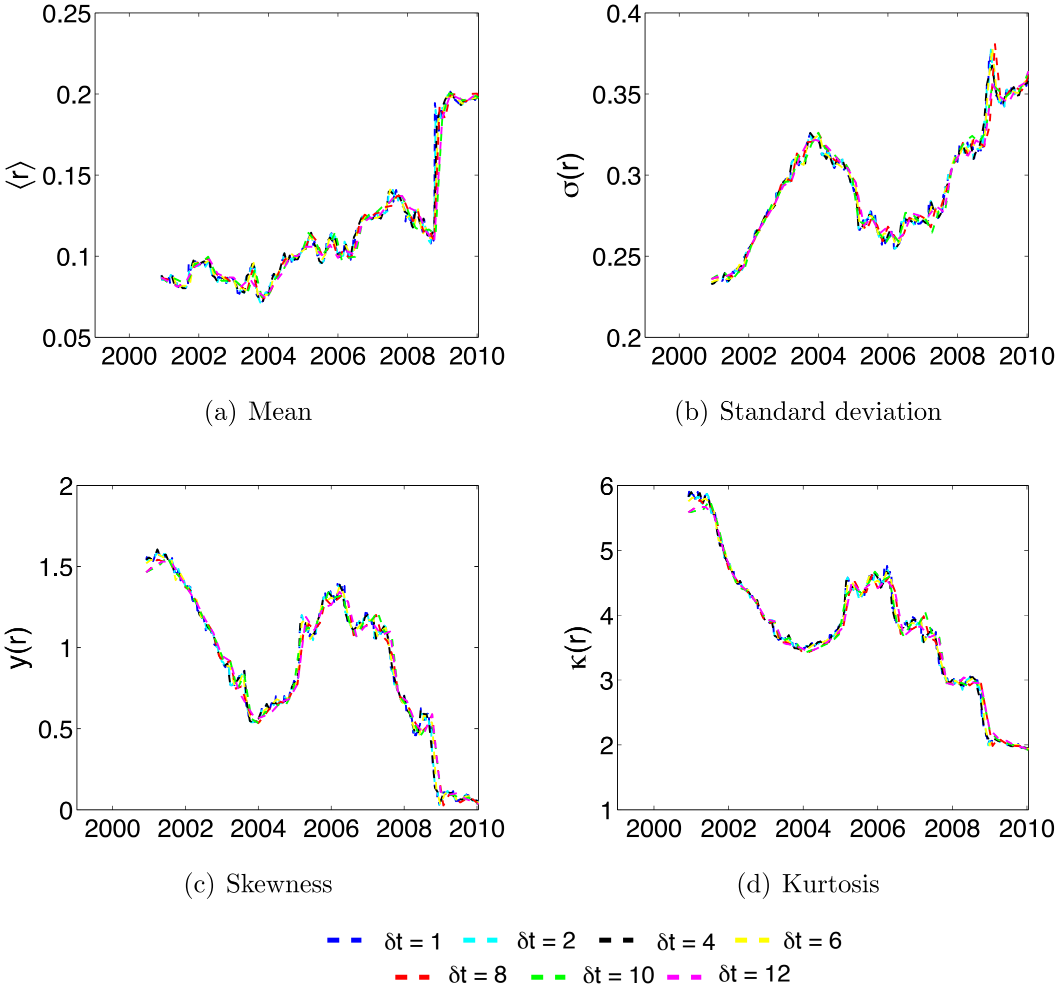


(a) Correlation over all time windows for AssetClasses data set

Window length	**Mean**	**Std. Dev.**	**Skewness**	**Kurtosis**
50	0.1151	0.3108	0.6698	3.3223
75	0.1159	0.2992	0.8087	3.6105
100	0.1148	0.2925	0.8868	3.7837
125	0.1146	0.2891	0.9285	3.8750
150	0.1137	0.2866	0.9575	3.9323
175	0.1135	0.2847	0.9750	3.9647
200	0.1132	0.2829	0.9880	3.9964

(b) First four moments averaged over all time windows

Figure B.2: Comparison of basic correlation statistics in AssetClasses data set averaged over time for different time window lengths.



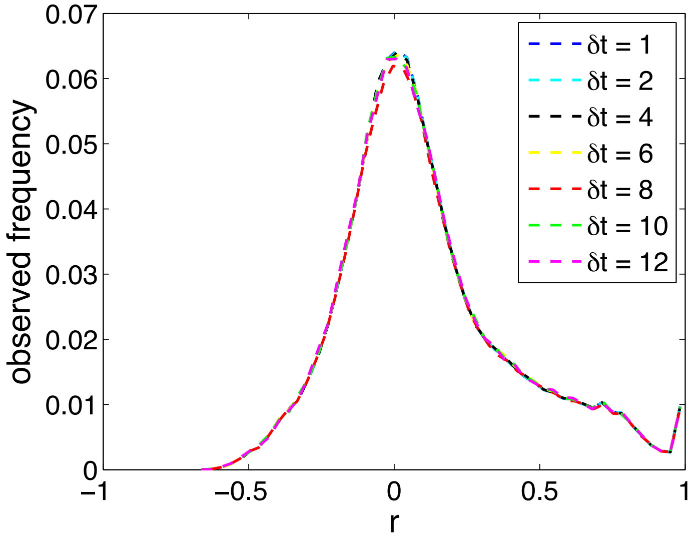
(a) Mean

(b) Standard deviation

(c) Skewness

(d) Kurtosis

Figure B.3: Comparison of basic correlation statistics in AssetClasses over time for different values of the step size between consecutive time windows.

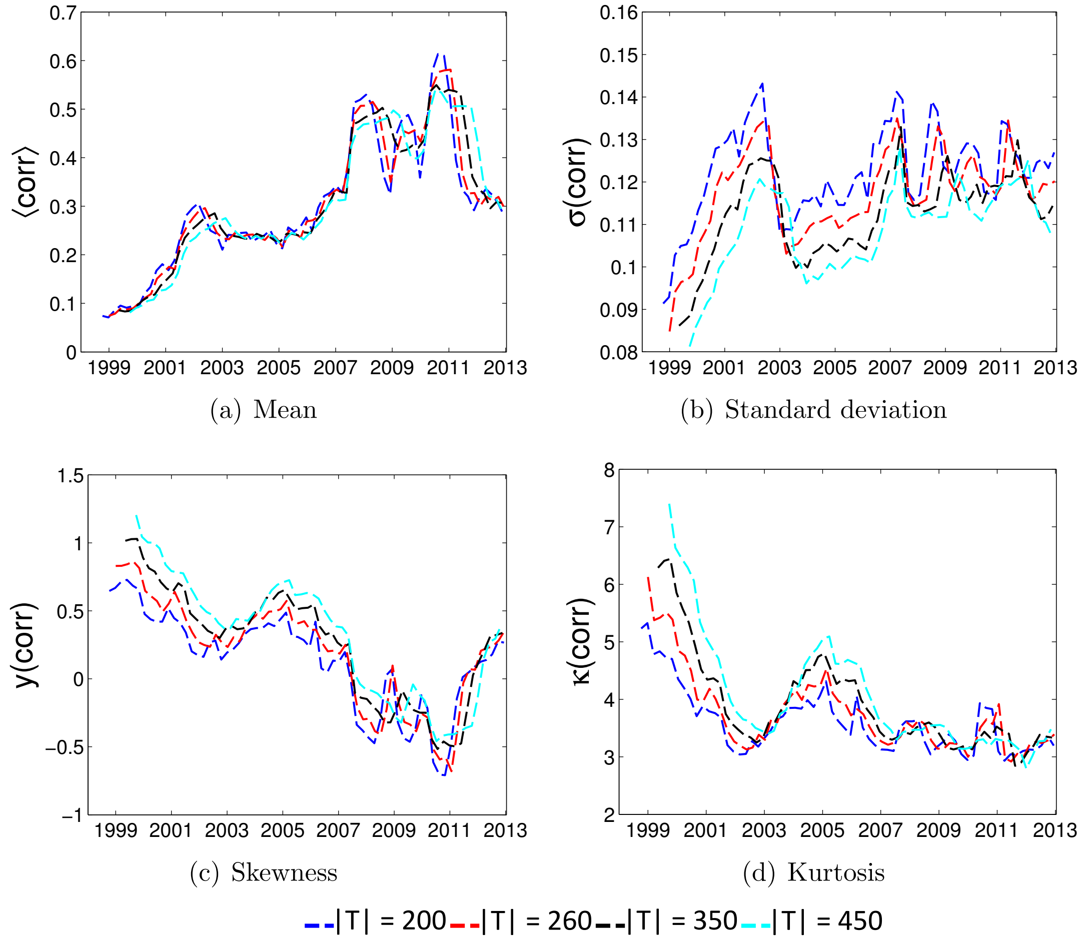


(a) Correlation over all time windows

Step	**Mean**	**Std. Dev.**	**Skewness**	**Kurtosis**
1	0.1148	0.2925	0.8873	3.7850
2	0.1148	0.2925	0.8868	3.7837
4	0.1152	0.2928	0.8839	3.7763
6	0.1155	0.2935	0.8770	3.7648
8	0.1158	0.2937	0.8704	3.7502
10	0.1154	0.2934	0.8778	3.7640
12	0.1153	0.2936	0.8743	3.7561

(b) First four moments averaged over all time windows

Figure B.4: Comparison of basic correlation statistics in AssetClasses averaged over time for different values of the step size between consecutive time windows.



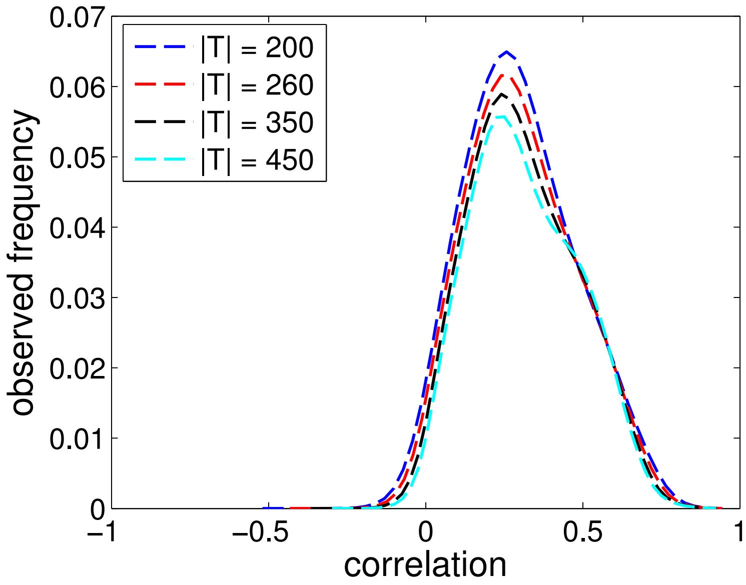
(a) Mean

(b) Standard deviation

(c) Skewness

(d) Kurtosis

Figure B.5: Comparison of basic correlation statistics in SINGLEASSETCLASS over time for different time window lengths.

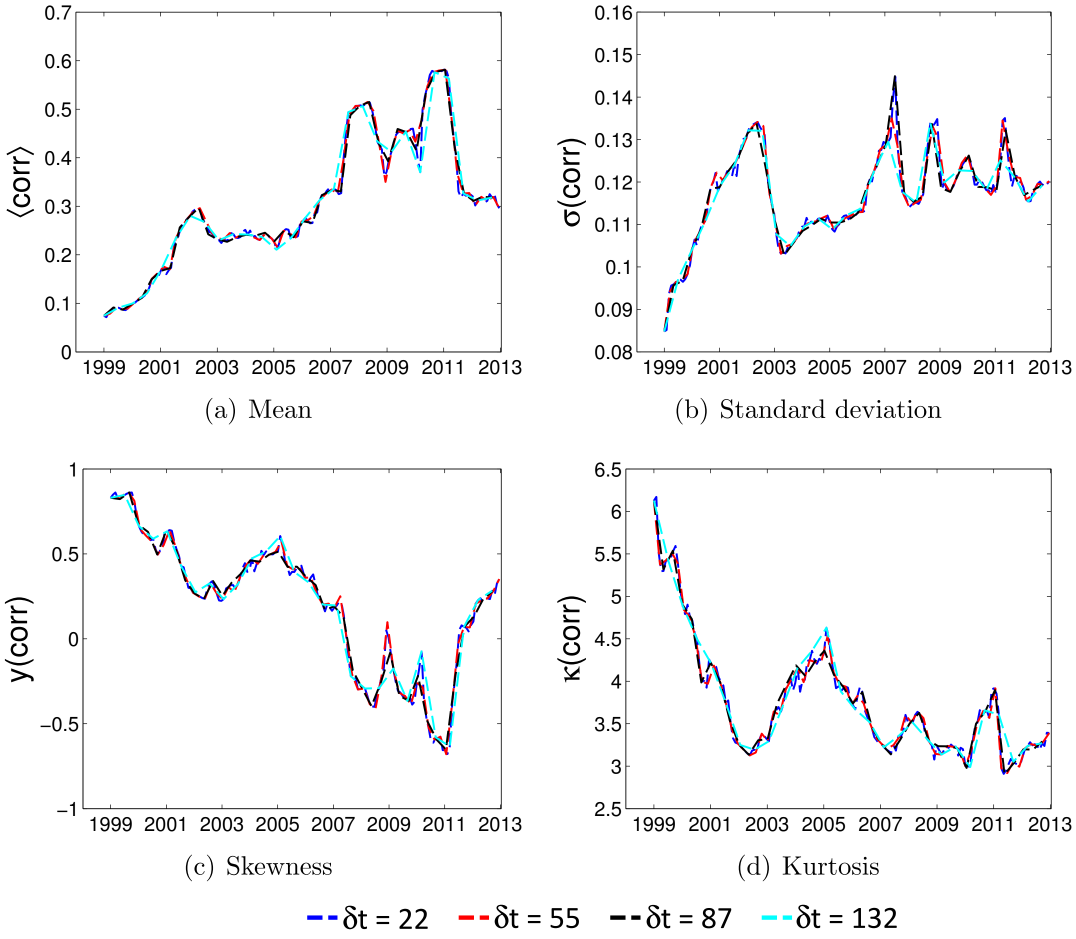


(a) Correlation over all time windows

Window length	**Mean**	**Std. Dev.**	**Skewness**	**Kurtosis**
200	0.3005	0.1221	0.1375	3.6101
260	0.3050	0.1171	0.1954	3.7551
350	0.3100	0.1123	0.2550	3.9149
450	0.3151	0.1091	0.3036	4.0441

(b) First four moments averaged over all time windows

Figure B.6: Comparison of basic correlation statistics in SingleAssetClass averaged over time for different time window lengths.



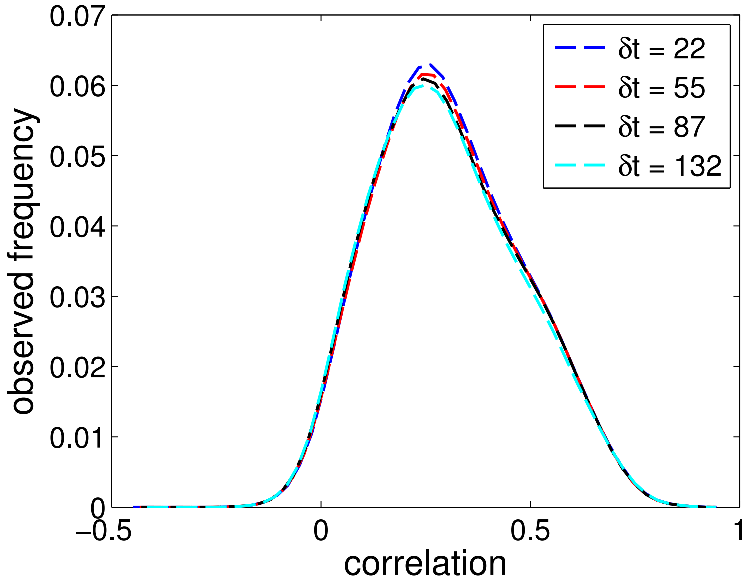
(a) Mean

(b) Standard deviation

(c) Skewness

(d) Kurtosis

Figure B.7: Comparison of basic correlation statistics in SingleAssetClass over time for different values of the step size between consecutive time windows.



(a) Correlation over all time windows

Step	Mean	Std. Dev.	Skewness	Kurtosis
22	0.3051	0.1171	0.1924	3.7496
55	0.3050	0.1171	0.1954	3.7551
87	0.3028	0.1167	0.1935	3.7848
132	0.3003	0.1162	0.2048	3.8106

(b) First four moments averaged over all time windows

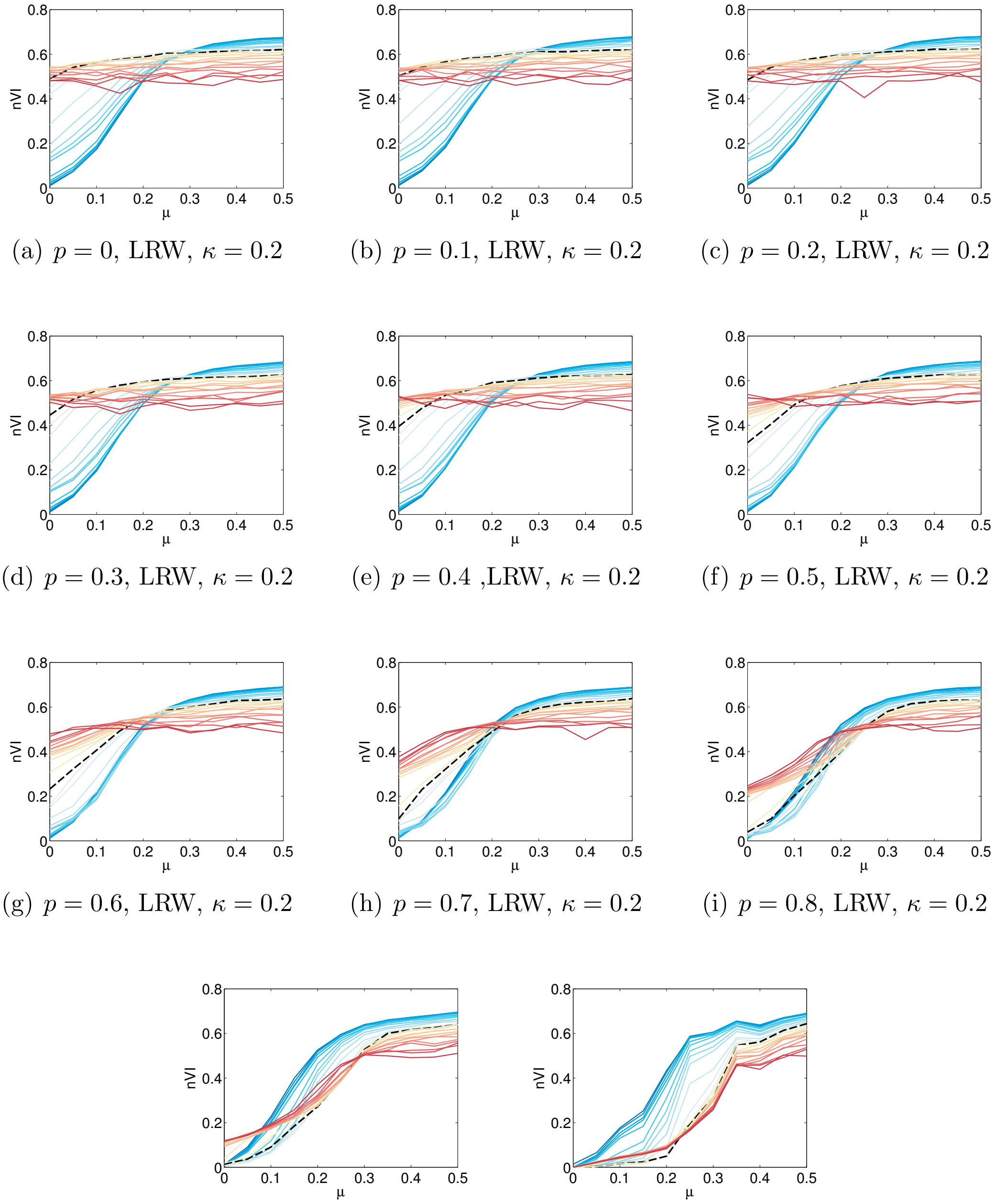
Figure B.8: Comparison of basic correlation statistics in SingleAssetClass averaged over time for different values of the step size between consecutive time windows.

Appendix C

Robustness of Qualitative Benchmark Results

In this section, we first show that the qualitative results observed with Louvain-RandWeighted (LRW) in Fig. 5.13 is relatively robust to the choice of similarity measure (see Figs C.1–C.5) and to a different choice of null distribution (see Figs C.6–C.7). We show plots for output partitions obtained after one iteration of the heuristic and after iterating the heuristic on a post-processed output multilayer partition until the partition no longer changes. The number of iterations is no more than six in all cases and we observe that the results from multiple iterations outperform those obtained after one iteration in all cases (this is particularly noticeable for $p \in \{0.8, 0.9, 1\}$). Furthermore, we repeat the experiment in Fig. 5.14 and show plots of persistence, intra-layer modularity, and nVI with LR and LRW for $\omega \in \{0, 0.1, \ldots, 9.9, 10\}$ instead of $\omega \in \{0, 0.1, \ldots, 2.9, 3\}$ (see Figs C.8–C.9). We observe that the results after multiple iterations outperform the results obtained after one iteration for both LR and LRW, and that the variability of the output partition seems to increase for $\omega \gtrapprox 3$. Finally, we show the experiments with Infomap in Figs 5.15 and 5.16 for all values $p \in \{0, 0.1, \ldots, 0.91\}$ instead of only $p \in \{0, 0.9, 1\}$ (see Figs C.10–C.13), and for larger values of N and $|\mathcal{T}|$ (see Figs C.14–C.15). We observe that the results are qualitatively similar to those of Figs 5.15 and 5.16.

In Figs C.1–C.5, we fix $(N, \mathcal{T}) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \ldots, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.05, \ldots, 4.45, 0.5\}$ and $\kappa = 0.2$. We omit this information from the caption of Figs C.1–C.5 and only specify the heuristic that we use, the sample of ω values, and the measure of similarity. In Figs C.6–C.15, we include the complete details of the figure in the caption.



(a) $p = 0$, LRW, $\kappa = 0.2$

(b) $p = 0.1$, LRW, $\kappa = 0.2$

(c) $p = 0.2$, LRW, $\kappa = 0.2$

(d) $p = 0.3$, LRW, $\kappa = 0.2$

(e) $p = 0.4$,LRW, $\kappa = 0.2$

(f) $p = 0.5$, LRW, $\kappa = 0.2$

(g) $p = 0.6$, LRW, $\kappa = 0.2$

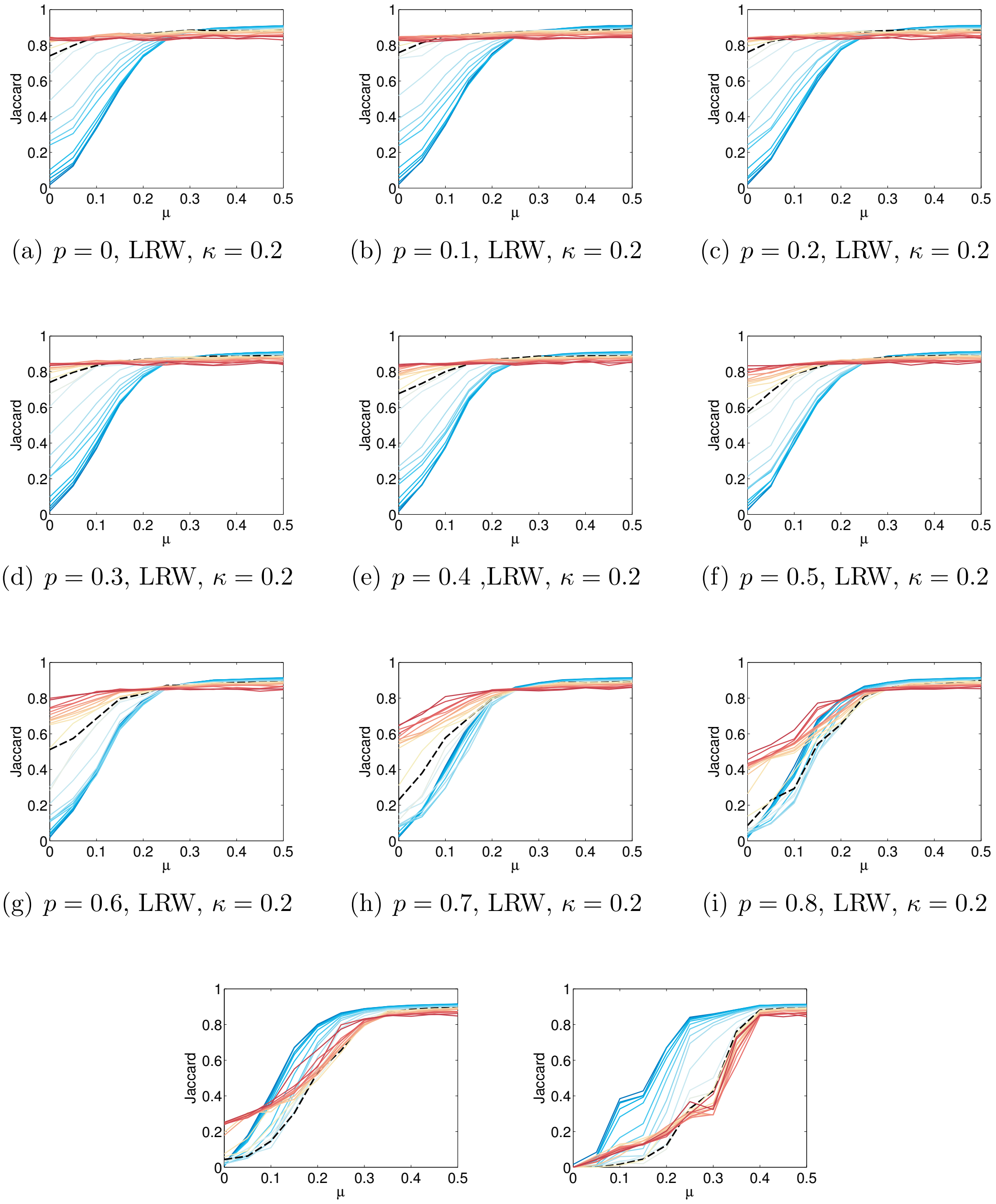
(h) $p = 0.7$, LRW, $\kappa = 0.2$

(i) $p = 0.8$, LRW, $\kappa = 0.2$

(j) $p = 0.9$, LRW, $\kappa = 0.2$

(k) $p = 1$, LRW, $\kappa = 0.2$

Figure C.1: We compute the nVI after one iteration of LouvainRandWeighted. The sample of ω values is $\{0, 0.2, \ldots, 4.8, 5\}$. We show plots for $\omega < 2.4$ with solid blue curves, those for $\omega > 2.4$ with solid red curves, and that of $\omega = 2.4$ with a black dashed curve. Colors scale with the value of ω from deep blue ($\omega = 0$) to deep red ($\omega = 5$) (a,b,c,d,e,f,g,h,i,j,k) correspond to $p \in \{0, 0.1, \ldots, 0.9, 1\}$, respectively.



(a) $p = 0$, LRW, $\kappa = 0.2$

(b) $p = 0.1$, LRW, $\kappa = 0.2$

(c) $p = 0.2$, LRW, $\kappa = 0.2$

(d) $p = 0.3$, LRW, $\kappa = 0.2$

(e) $p = 0.4$,LRW, $\kappa = 0.2$

(f) $p = 0.5$, LRW, $\kappa = 0.2$

(g) $p = 0.6$, LRW, $\kappa = 0.2$

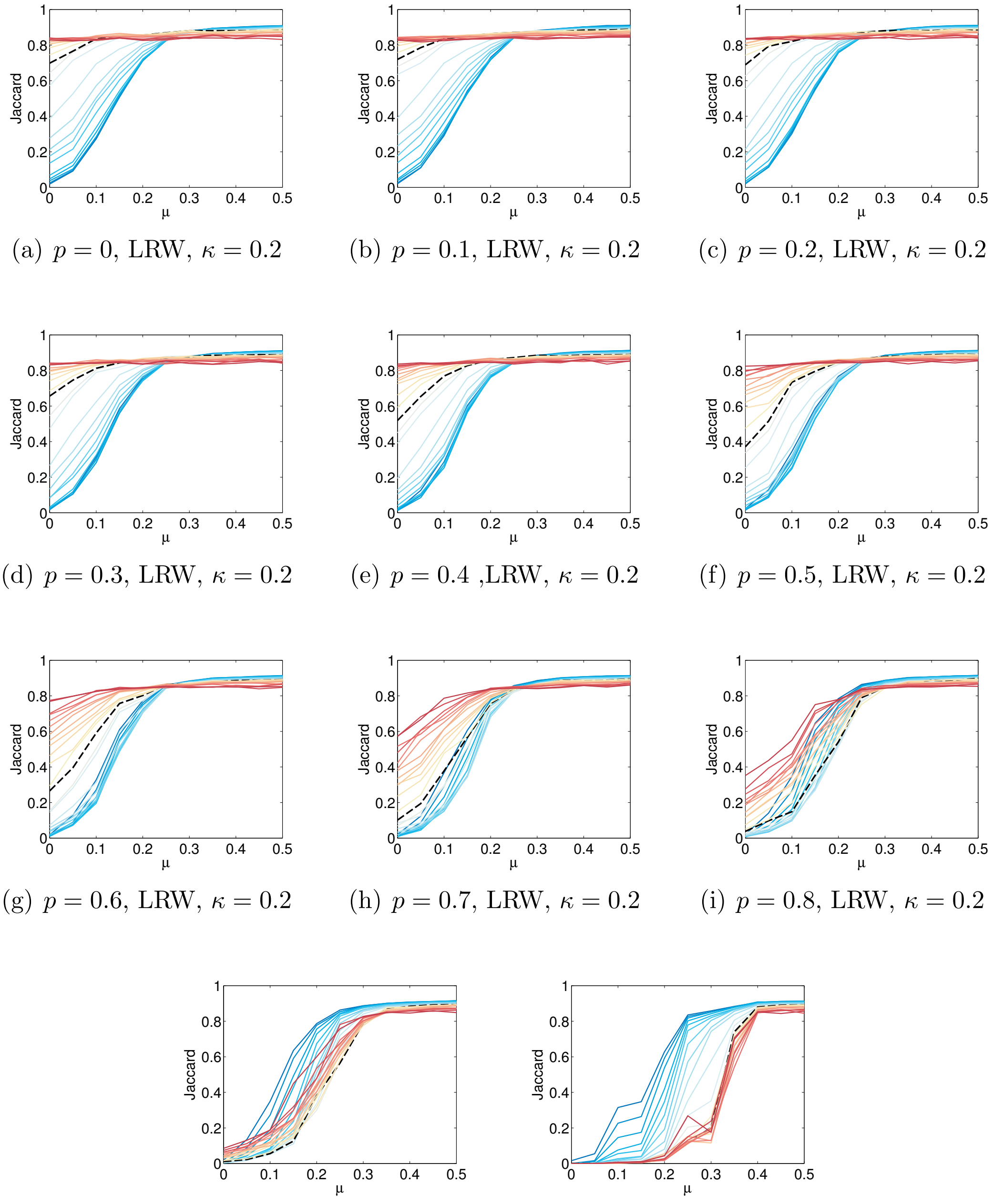
(h) $p = 0.7$, LRW, $\kappa = 0.2$

(i) $p = 0.8$, LRW, $\kappa = 0.2$

(j) $p = 0.9$, LRW, $\kappa = 0.2$

(k) $p = 1$, LRW, $\kappa = 0.2$

Figure C.2: We compute the Jaccard coefficient after one iteration of Louvain-RandWeighted. The sample of ω values is $\{0, 0.2, \ldots, 4.8, 5\}$. We show plots for $\omega < 2.4$ with solid blue curves, those for $\omega > 2.4$ with solid red curves, and that of $\omega = 2.4$ with a black dashed curve. Colors scale with the value of ω from deep blue ($\omega = 0$) to deep red ($\omega = 5$) (a,b,c,d,e,f,g,h,i,j,k) correspond to $p \in \{0, 0.1, \ldots, 0.9, 1\}$, respectively.



(a) $p = 0$, LRW, $\kappa = 0.2$

(b) $p = 0.1$, LRW, $\kappa = 0.2$

(c) $p = 0.2$, LRW, $\kappa = 0.2$

(d) $p = 0.3$, LRW, $\kappa = 0.2$

(e) $p = 0.4$,LRW, $\kappa = 0.2$

(f) $p = 0.5$, LRW, $\kappa = 0.2$

(g) $p = 0.6$, LRW, $\kappa = 0.2$

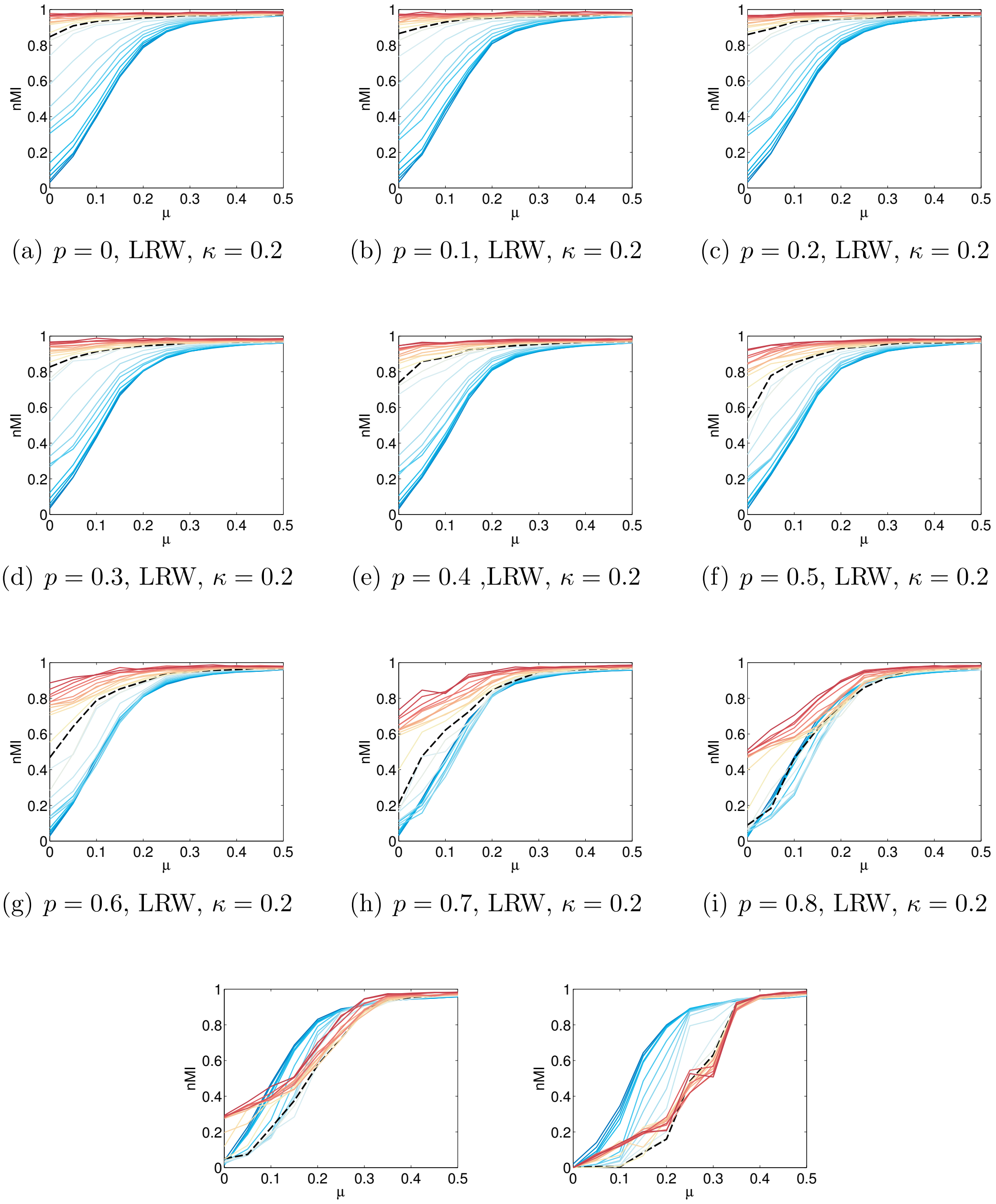
(h) $p = 0.7$, LRW, $\kappa = 0.2$

(i) $p = 0.8$, LRW, $\kappa = 0.2$

(j) $p = 0.9$, LRW, $\kappa = 0.2$

(k) $p = 1$, LRW, $\kappa = 0.2$

Figure C.3: We compute the Jaccard coefficient after iterating LouvainRandWeighted on the post-processed output partition until it no longer changes. The sample of ω values is $\{0, 0.2, \ldots, 4.8, 5\}$. We show plots for $\omega < 2.4$ with solid blue curves, those for $\omega > 2.4$ with solid red curves, and that of $\omega = 2.4$ with a black dashed curve. Colors scale with the value of ω from deep blue ($\omega = 0$) to deep red ($\omega = 5$) (a,b,c,d,e,f,g,h,i,j,k) correspond to $p \in \{0, 0.1, \ldots, 0.9, 1\}$, respectively.



(a) $p = 0$, LRW, $\kappa = 0.2$

(b) $p = 0.1$, LRW, $\kappa = 0.2$

(c) $p = 0.2$, LRW, $\kappa = 0.2$

(d) $p = 0.3$, LRW, $\kappa = 0.2$

(e) $p = 0.4$,LRW, $\kappa = 0.2$

(f) $p = 0.5$, LRW, $\kappa = 0.2$

(g) $p = 0.6$, LRW, $\kappa = 0.2$

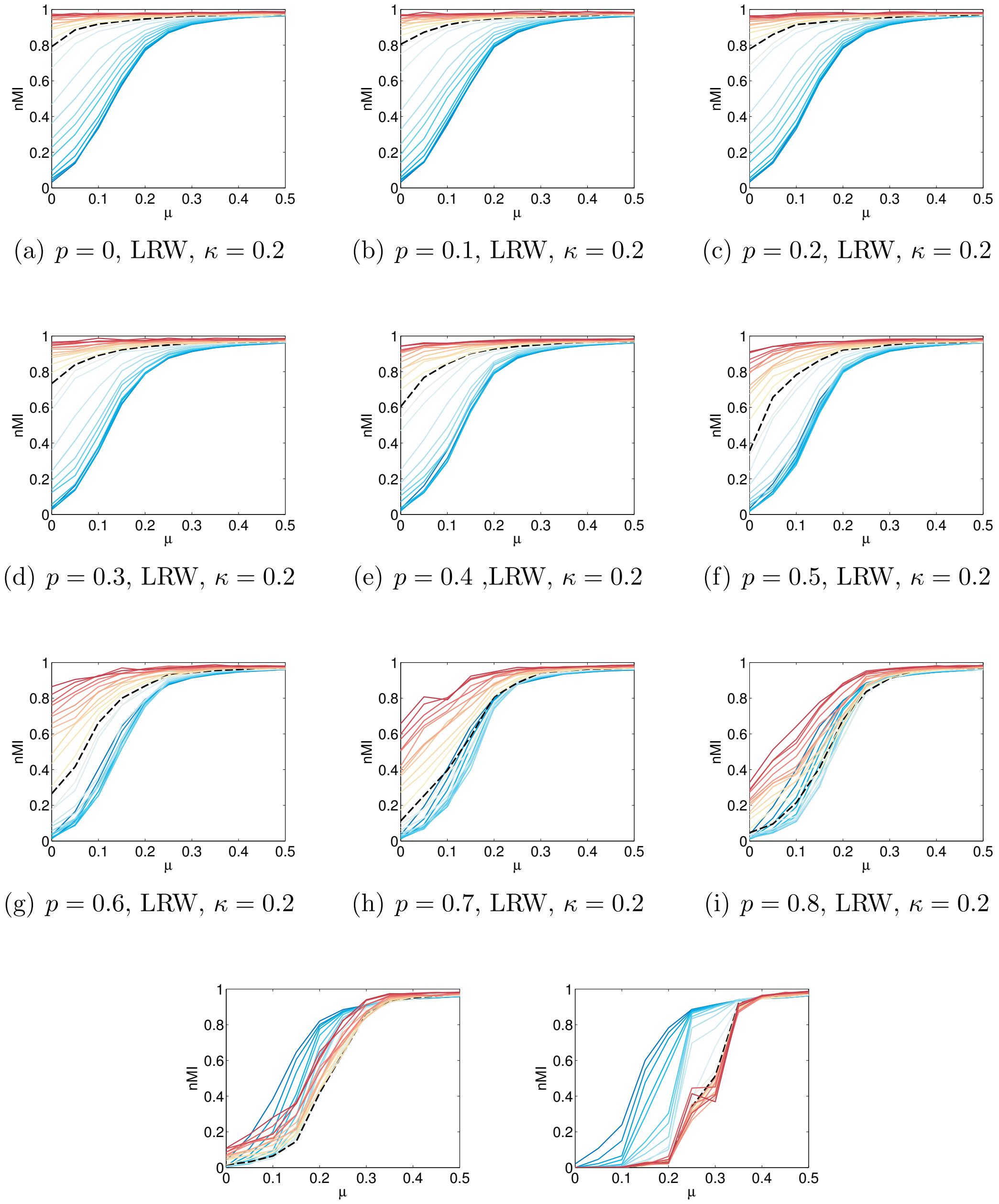
(h) $p = 0.7$, LRW, $\kappa = 0.2$

(i) $p = 0.8$, LRW, $\kappa = 0.2$

(j) $p = 0.9$, LRW, $\kappa = 0.2$

(k) $p = 1$, LRW, $\kappa = 0.2$

Figure C.4: We compute the nMI after one iteration of LouvainRandWeighted (the expression for nMI is given by 1−nVI in (2.33) with a normalization of VI in (2.31) by the joint entropy instead of $\log N$ [110, 128]). We plot 1−nMI for consistency with previous figures. The sample of ω values is $\{0, 0.2, \ldots, 4.8, 5\}$. We show plots for $\omega < 2.4$ with solid blue curves, those for $\omega > 2.4$ with solid red curves, and that of $\omega = 2.4$ with a black dashed curve. Colors scale with the value of ω from deep blue ($\omega = 0$) to deep red ($\omega = 5$) (a,b,c,d,e,f,g,h,i,j,k) correspond to $p \in \{0, 0.1, \ldots, 0.9, 1\}$, respectively.



(a) $p = 0$, LRW, $\kappa = 0.2$

(b) $p = 0.1$, LRW, $\kappa = 0.2$

(c) $p = 0.2$, LRW, $\kappa = 0.2$

(d) $p = 0.3$, LRW, $\kappa = 0.2$

(e) $p = 0.4$,LRW, $\kappa = 0.2$

(f) $p = 0.5$, LRW, $\kappa = 0.2$

(g) $p = 0.6$, LRW, $\kappa = 0.2$

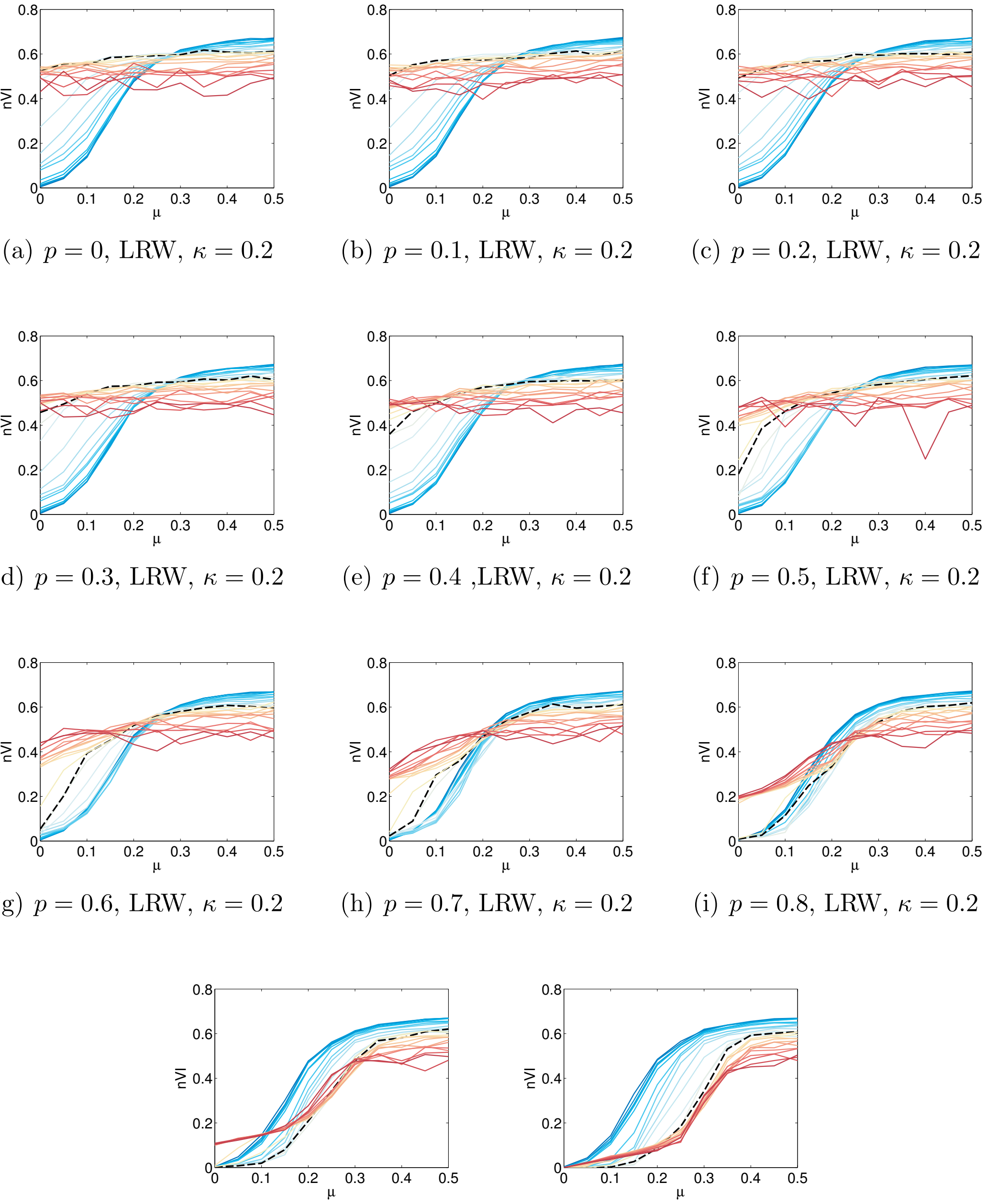
(h) $p = 0.7$, LRW, $\kappa = 0.2$

(i) $p = 0.8$, LRW, $\kappa = 0.2$

(j) $p = 0.9$, LRW, $\kappa = 0.2$

(k) $p = 1$, LRW, $\kappa = 0.2$

Figure C.5: We compute the nMI after iterating LouvainRandWeighted on the post-processed output partition until it no longer changes (the expression for nMI is given by 1−nVI in (2.33) with a normalization of VI in (2.31) by the joint entropy instead of $\log N$ [110, 128]). We plot 1−nMI for consistency with previous figures. The sample of ω values is $\{0, 0.2, \ldots, 4.8, 5\}$. We show plots for $\omega < 2.4$ with solid blue curves, those for $\omega > 2.4$ with solid red curves, and that of $\omega = 2.4$ with a black dashed curve. Colors scale with the value of ω from deep blue ($\omega = 0$) to deep red ($\omega = 5$) (a,b,c,d,e,f,g,h,i,j,k) correspond to $p \in \{0, 0.1, \ldots, 0.9, 1\}$, respectively..



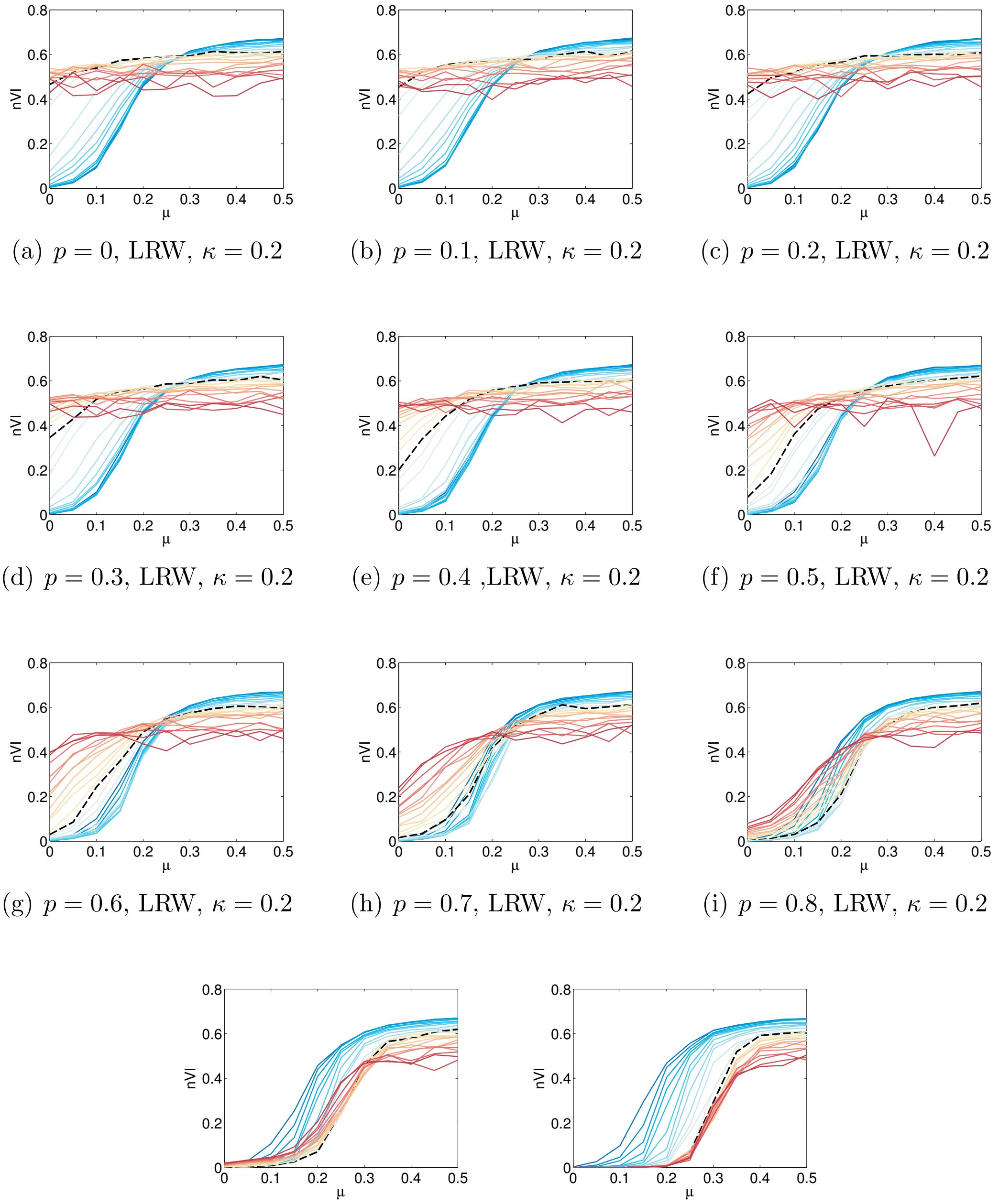
(a) $p = 0$, LRW, $\kappa = 0.2$ (b) $p = 0.1$, LRW, $\kappa = 0.2$ (c) $p = 0.2$, LRW, $\kappa = 0.2$

(d) $p = 0.3$, LRW, $\kappa = 0.2$ (e) $p = 0.4$,LRW, $\kappa = 0.2$ (f) $p = 0.5$, LRW, $\kappa = 0.2$

(g) $p = 0.6$, LRW, $\kappa = 0.2$ (h) $p = 0.7$, LRW, $\kappa = 0.2$ (i) $p = 0.8$, LRW, $\kappa = 0.2$

(j) $p = 0.9$, LRW, $\kappa = 0.2$ (k) $p = 1$, LRW, $\kappa = 0.2$

Figure C.6: We compute the nVI after one iteration of LouvainRandWeighted. We fix $(N, \mathcal{T}) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \dots, 0.9, 1\}$ and we generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.05, \dots, 4.45, 0.5\}$ and $\kappa = 0.2$. We fix a null distribution that is uniform within and between layers (i.e., $\boldsymbol{p}_s = \boldsymbol{p}$ for all s and $p_i = p_j$ for all $i, j \in \{1, \dots, |p|\}$). We fix $p_i = 0.2N$ for all i (and thus $|\boldsymbol{p}| = 5$). The sample of ω values is $\{0, 0.2, \dots, 4.8, 5\}$. We show plots for $\omega < 2.4$ with solid blue curves, those for $\omega > 2.4$ with solid red curves, and that of $\omega = 2.4$ with a black dashed curve. Colors scale with the value of ω from deep blue ($\omega = 0$) to deep red ($\omega = 5$) (a,b,c,d,e,f,g,h,i,j,k) correspond to $p \in \{0, 0.1, \dots, 0.9, 1\}$, respectively.



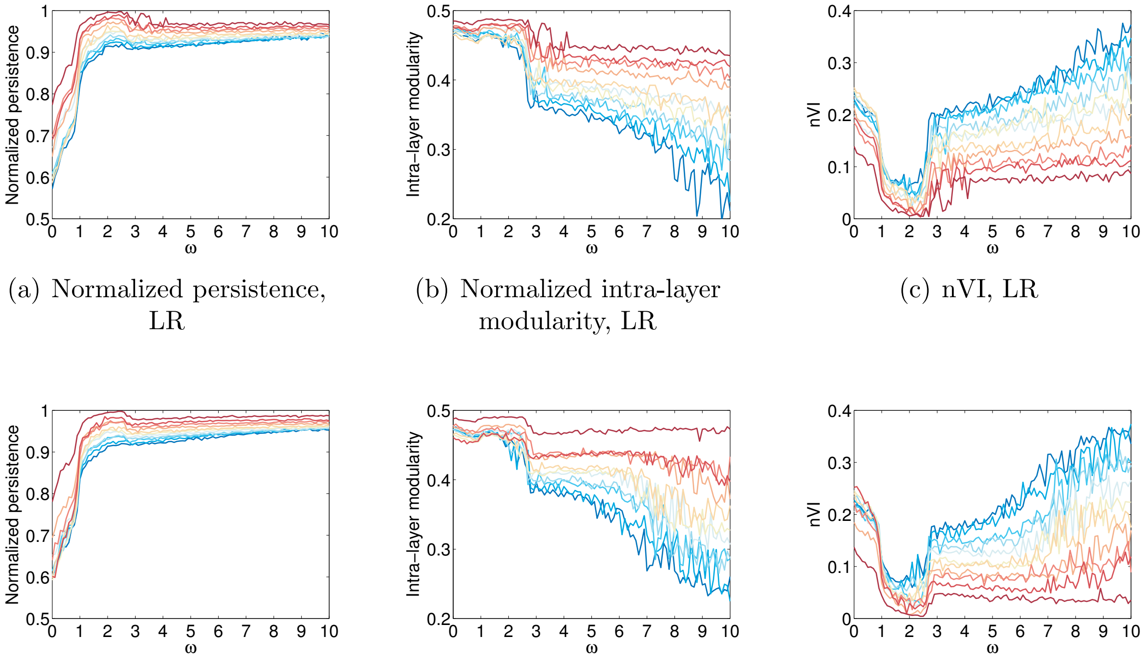
(a) $p = 0$, LRW, $\kappa = 0.2$ (b) $p = 0.1$, LRW, $\kappa = 0.2$ (c) $p = 0.2$, LRW, $\kappa = 0.2$

(d) $p = 0.3$, LRW, $\kappa = 0.2$ (e) $p = 0.4$,LRW, $\kappa = 0.2$ (f) $p = 0.5$, LRW, $\kappa = 0.2$

(g) $p = 0.6$, LRW, $\kappa = 0.2$ (h) $p = 0.7$, LRW, $\kappa = 0.2$ (i) $p = 0.8$, LRW, $\kappa = 0.2$

(j) $p = 0.9$, LRW, $\kappa = 0.2$ (k) $p = 1$, LRW, $\kappa = 0.2$

Figure C.7: We compute the nVI after iterating LouvainRandWeighted on the post-processed output partition until it no longer changes. We fix $(N, \mathcal{T}) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \ldots, 0.9, 1\}$ and we generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.05, \ldots, 4.45, 0.5\}$ and $\kappa = 0.2$. We fix a null distribution that is uniform within and between layers (i.e., $\boldsymbol{p}_s = \boldsymbol{p}$ for all s and $p_i = p_j$ for all $i, j \in \{1, \ldots, |p|\}$). We fix $p_i = 0.2N$ for all i (and thus $|\boldsymbol{p}| = 5$). The sample of ω values is $\{0, 0.2, \ldots, 4.8, 5\}$. We show plots for $\omega < 2.4$ with solid blue curves, those for $\omega > 2.4$ with solid red curves, and that of $\omega = 2.4$ with a black dashed curve. Colors scale with the value of ω from deep blue ($\omega = 0$) to deep red ($\omega = 5$) (a,b,c,d,e,f,g,h,i,j,k) correspond to $p \in \{0, 0.1, \ldots, 0.9, 1\}$, respectively.



(a) Normalized persistence, LR

(b) Normalized intra-layer modularity, LR

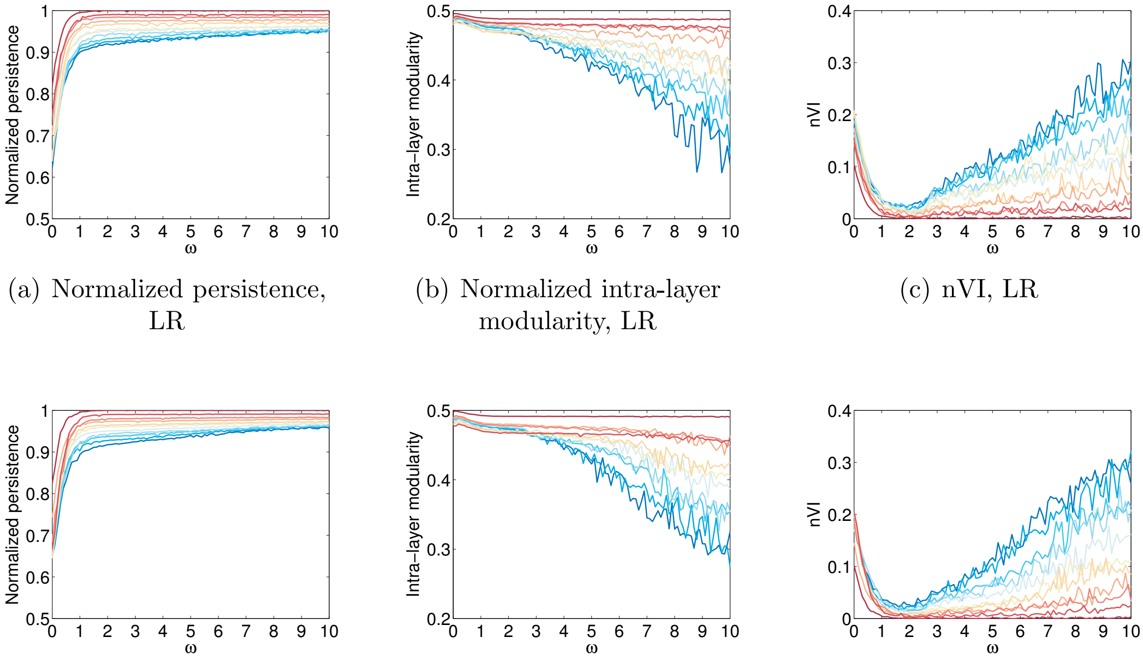
(c) nVI, LR

(d) Normalized persistence, LRW

(e) Normalized intra-layer modularity, LRW

(f) nVI, LRW

Figure C.8: We fix $(N, \mathcal{T}) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0.9, 0.91, \ldots, 0.99, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu = 0.1$ and $\kappa = 0.2$. The sample of ω values is $\{0, 0.1, \ldots, 9.9, 10\}$. We show plots of normalized persistence, normalized intra-layer modularity, and nVI between the output and planted multilayer partitions for (a,b,c) LouvainRand, and (d,e,f) LouvainRandWeighted. Each curve corresponds to a value of p in our sample. Colors scale with the value of p from deep blue ($p = 0$) to deep red ($p = 1$). We normalize persistence by $(N|\mathcal{T}| - 1)$ and intra-layer modularity by $\sum_{s=1}^{|\mathcal{T}|}(\mathbf{1}^T \boldsymbol{A}_s \mathbf{1})$ for each value of ω. The output partitions that we use in these panels are obtained after one run of the heusistic.



(a) Normalized persistence, LR

(b) Normalized intra-layer modularity, LR

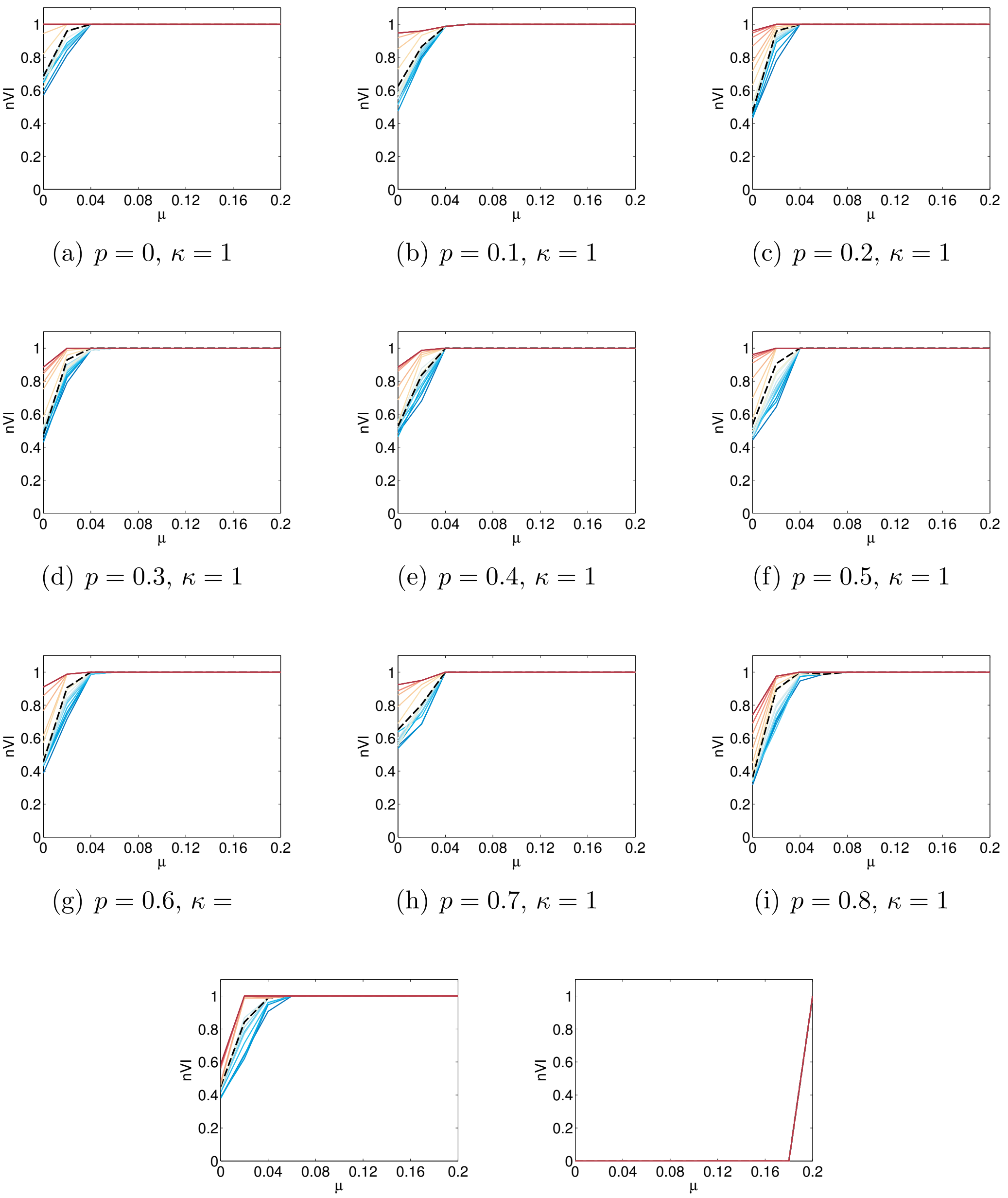
(c) nVI, LR

(d) Normalized persistence, LRW

(e) Normalized intra-layer modularity, LRW

(f) nVI, LRW

Figure C.9: We fix $(N, \mathcal{T}) = (150, 100)$ and use multilayer modularity maximization with the uniform null network and $\gamma = 1$. We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0.9, 0.91, \ldots, 0.99, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu = 0.1$ and $\kappa = 0.2$. The sample of ω values is $\{0, 0.1, \ldots, 9.9, 10\}$. We show plots of normalized persistence, normalized intra-layer modularity, and nVI between the output and planted multilayer partitions for (a,b,c) LouvainRand, and (d,e,f) LouvainRandWeighted. Each curve corresponds to a value of p in our sample. Colors scale with the value of p from deep blue ($p = 0$) to deep red ($p = 1$). We normalize persistence by $(N|\mathcal{T}| - 1)$ and intra-layer modularity by $\sum_{s=1}^{|\mathcal{T}|}(\mathbf{1}^T \boldsymbol{A}_s \mathbf{1})$ for each value of ω. The output partitions that we use in these panels are obtained after iterating the heuristic on a post-processed output partition until it no longer changes.



(a) $p = 0$, $\kappa = 1$

(b) $p = 0.1$, $\kappa = 1$

(c) $p = 0.2$, $\kappa = 1$

(d) $p = 0.3$, $\kappa = 1$

(e) $p = 0.4$, $\kappa = 1$

(f) $p = 0.5$, $\kappa = 1$

(g) $p = 0.6$, $\kappa =$

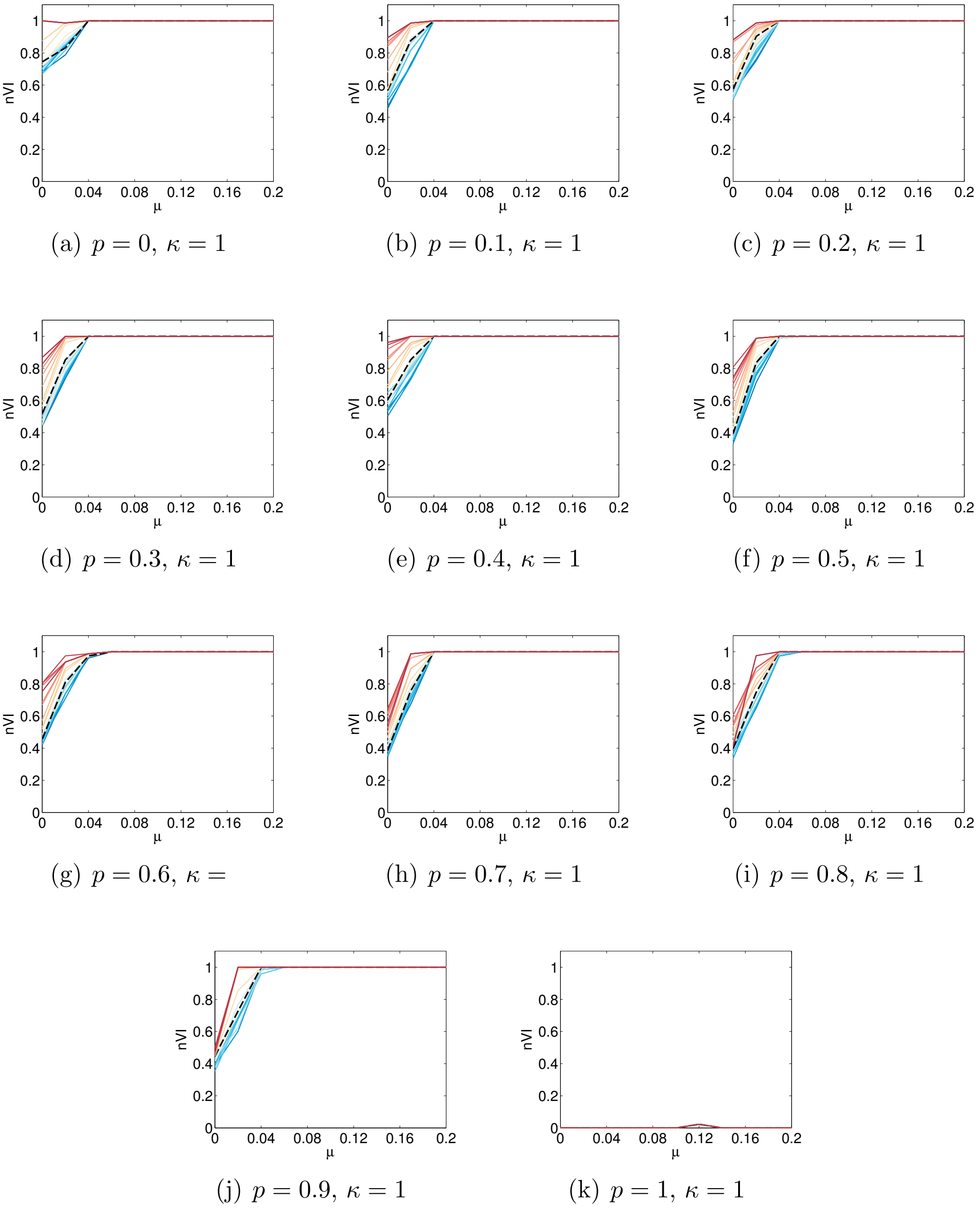
(h) $p = 0.7$, $\kappa = 1$

(i) $p = 0.8$, $\kappa = 1$

(j) $p = 0.9$, $\kappa = 1$

(k) $p = 1$, $\kappa = 1$

Figure C.10: We fix $(N, \mathcal{T}) = (100, 50)$ and use multiplex-case Infomap (i.e., the random walker in equation (2.26) can "relax" to any layer). The sample of "relax-rate" values is $r \in \{0, 0.01, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1\}$. We show plots for $r < 0.1$ with solid blue curves, those for $r > 0.1$ with solid red curves, and that of $r = 0.1$ with a black dashed curve. Colors scale with the value of r from deep blue ($r = 0$) to deep red ($r = 1$). We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \ldots, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.02, \ldots, 0.18, 0.2\}$ and $\kappa = 1$. For each pair (p, μ) we generate a multilayer partition and a multilayer network. We perform one run of Infomap for each value of r on each multilayer network. (a,b,c,d,e,f,g,h,i,j,k) Infomap and $p \in \{0, 0.1, \ldots, 0.9, 1\}$.



(a) $p = 0$, $\kappa = 1$

(b) $p = 0.1$, $\kappa = 1$

(c) $p = 0.2$, $\kappa = 1$

(d) $p = 0.3$, $\kappa = 1$

(e) $p = 0.4$, $\kappa = 1$

(f) $p = 0.5$, $\kappa = 1$

(g) $p = 0.6$, $\kappa =$

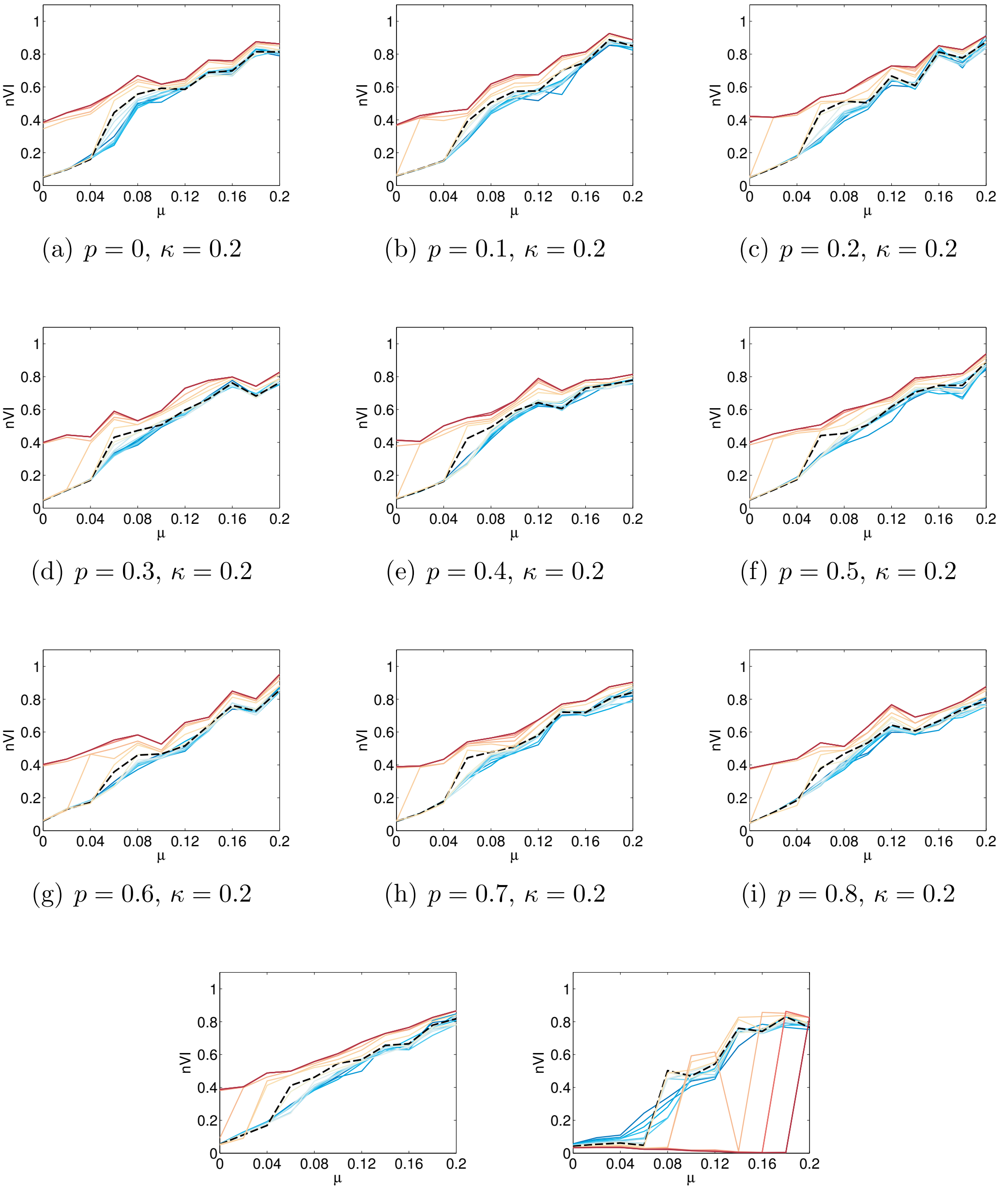
(h) $p = 0.7$, $\kappa = 1$

(i) $p = 0.8$, $\kappa = 1$

(j) $p = 0.9$, $\kappa = 1$

(k) $p = 1$, $\kappa = 1$

Figure C.11: We fix $(N, \mathcal{T}) = (100, 50)$ and use temporal-case Infomap (i.e., the random walker in equation (2.26) can only "relax" to adjacent layers). The sample of "relax-rate" values is $r \in \{0, 0.01, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1\}$. We show plots for $r < 0.1$ with solid blue curves, those for $r > 0.1$ with solid red curves, and that of $r = 0.1$ with a black dashed curve. Colors scale with the value of r from deep blue ($r = 0$) to deep red ($r = 1$). We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \ldots, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.02, \ldots, 0.18, 0.2\}$ and $\kappa = 1$. For each pair (p, μ) we generate a multilayer partition and a multilayer network. We perform one run of Infomap for each value of r on each multilayer network. (a,b,c,d,e,f,g,h,i,j,k) Infomap and $p \in \{0, 0.1, \ldots, 0.9, 1\}$.



(a) $p = 0$, $\kappa = 0.2$

(b) $p = 0.1$, $\kappa = 0.2$

(c) $p = 0.2$, $\kappa = 0.2$

(d) $p = 0.3$, $\kappa = 0.2$

(e) $p = 0.4$, $\kappa = 0.2$

(f) $p = 0.5$, $\kappa = 0.2$

(g) $p = 0.6$, $\kappa = 0.2$

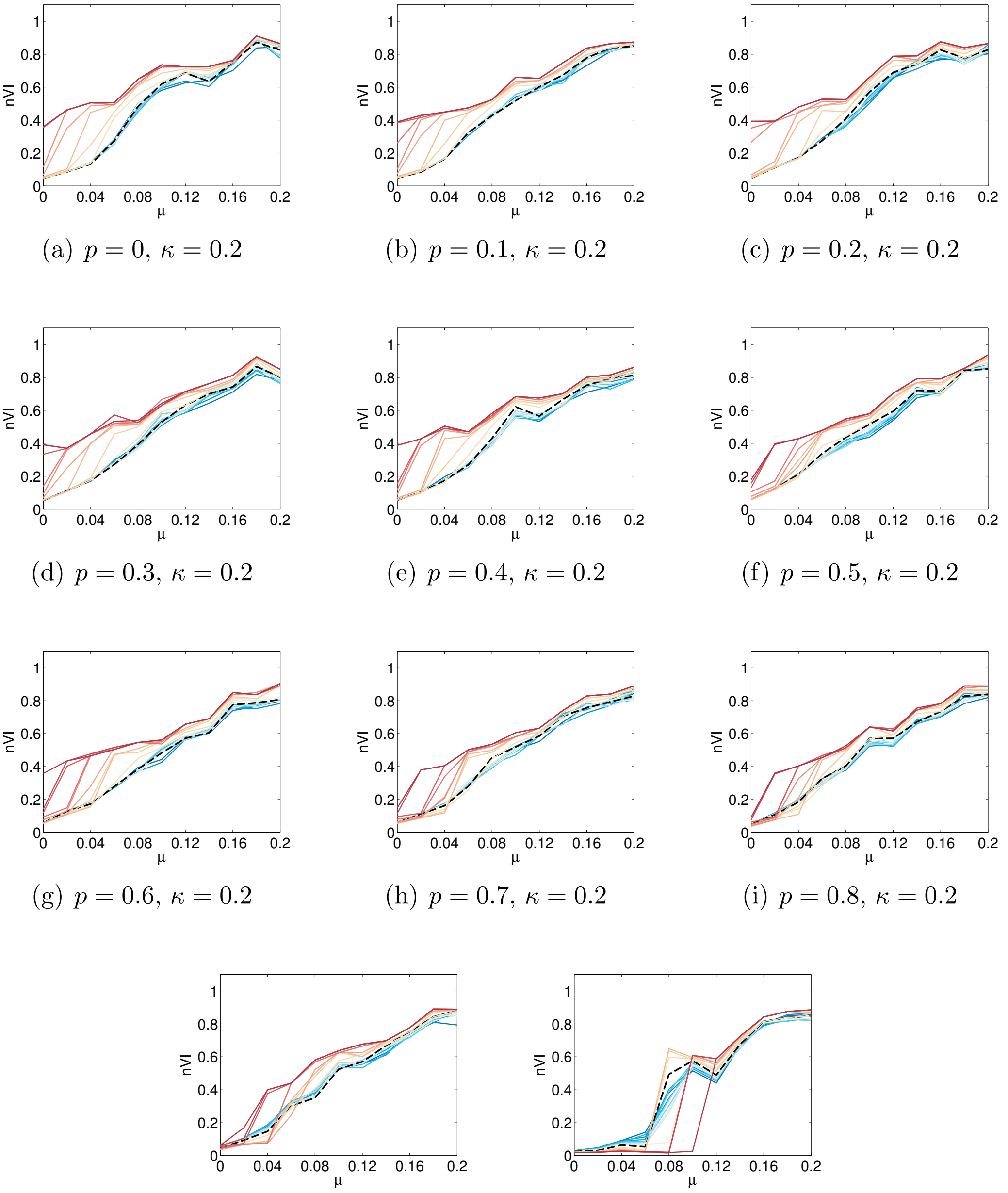
(h) $p = 0.7$, $\kappa = 0.2$

(i) $p = 0.8$, $\kappa = 0.2$

(j) $p = 0.9$, $\kappa = 0.2$

(k) $p = 1$, $\kappa = 0.2$

Figure C.12: We fix $(N, \mathcal{T}) = (100, 50)$ and use multiplex-case Infomap (i.e., the random walker in equation (2.26) can "relax" to any layer). The sample of "relax-rate" values is $r \in \{0, 0.01, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1\}$. We show plots for $r < 0.1$ with solid blue curves, those for $r > 0.1$ with solid red curves, and that of $r = 0.1$ with a black dashed curve. Colors scale with the value of r from deep blue ($r = 0$) to deep red ($r = 1$). We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \ldots, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.02, \ldots, 0.18, 0.2\}$ and $\kappa = 2$. For each pair (p, μ) we generate a multilayer partition and a multilayer network. We perform one run of Infomap for each value of r on each multilayer network. (a,b,c,d,e,f,g,h,i,j,k) Infomap and $p \in \{0, 0.1, \ldots, 0.9, 1\}$.



(a) $p = 0$, $\kappa = 0.2$

(b) $p = 0.1$, $\kappa = 0.2$

(c) $p = 0.2$, $\kappa = 0.2$

(d) $p = 0.3$, $\kappa = 0.2$

(e) $p = 0.4$, $\kappa = 0.2$

(f) $p = 0.5$, $\kappa = 0.2$

(g) $p = 0.6$, $\kappa = 0.2$

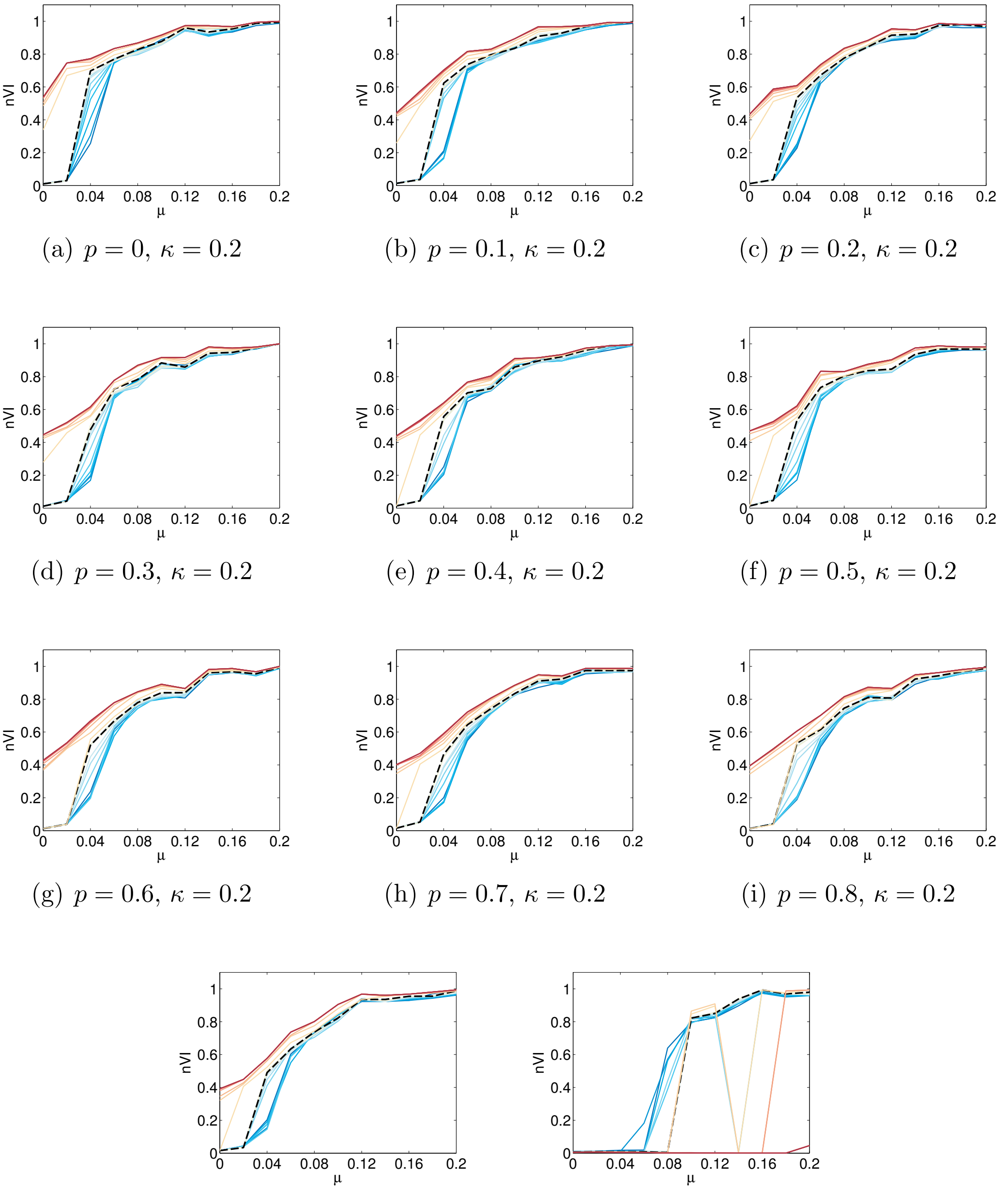
(h) $p = 0.7$, $\kappa = 0.2$

(i) $p = 0.8$, $\kappa = 0.2$

(j) $p = 0.9$, $\kappa = 0.2$

(k) $p = 1$, $\kappa = 0.2$

Figure C.13: We fix $(N, \mathcal{T}) = (100, 50)$ and use temporal-case Infomap (i.e., the random walker in equation (2.26) can only "relax" to adjacent layers). The sample of "relax-rate" values is $r \in \{0, 0.01, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1\}$. We show plots for $r < 0.1$ with solid blue curves, those for $r > 0.1$ with solid red curves, and that of $r = 0.1$ with a black dashed curve. Colors scale with the value of r from deep blue ($r = 0$) to deep red ($r = 1$). We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \ldots, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.02, \ldots, 0.18, 0.2\}$ and $\kappa = 0.2$. For each pair (p, μ) we generate a multilayer partition and a multilayer network. We perform one run of Infomap for each value of r on each multilayer network. (a,b,c,d,e,f,g,h,i,j,k) Infomap and $p \in \{0, 0.1, \ldots, 0.9, 1\}$.



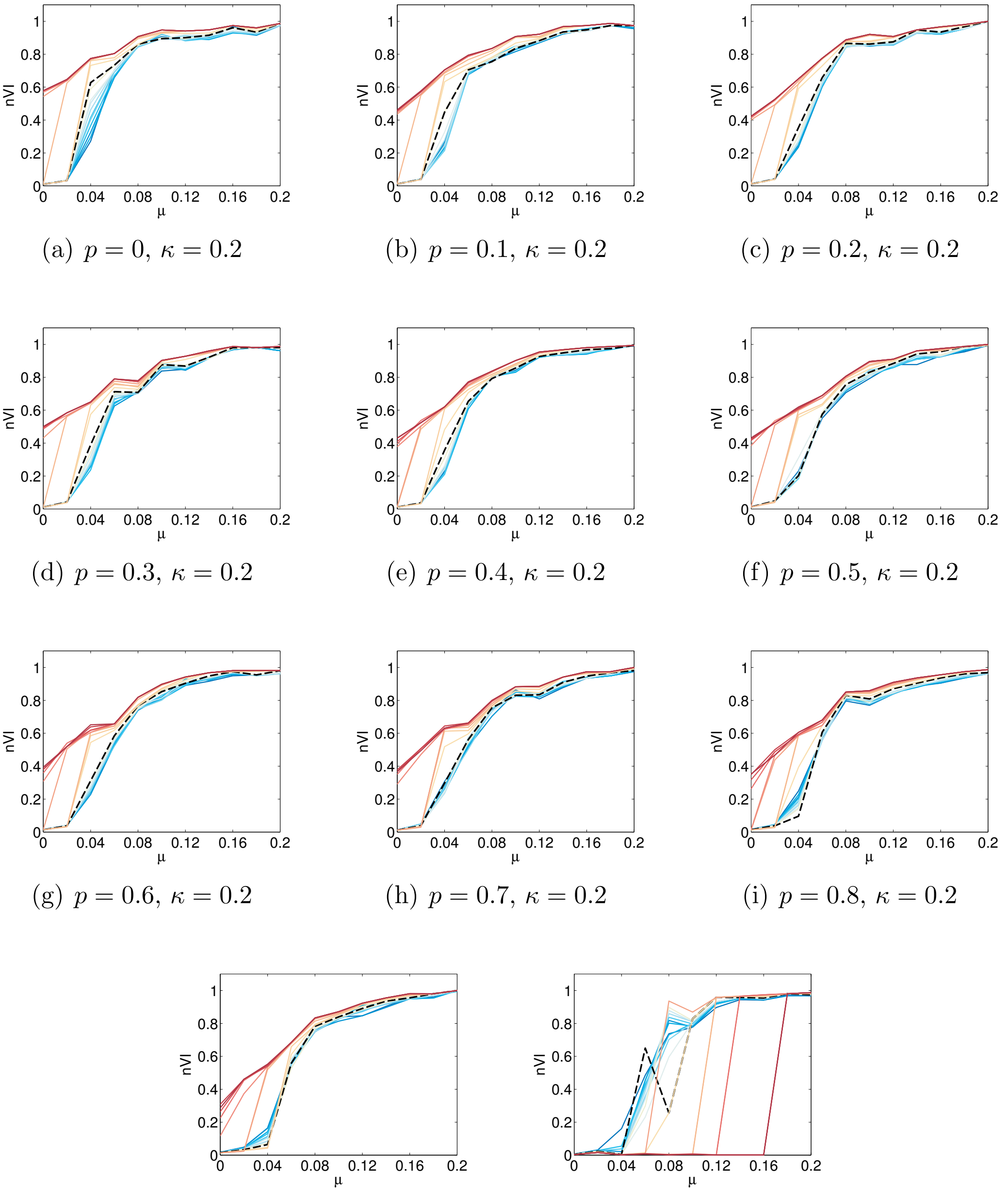
(a) $p = 0$, $\kappa = 0.2$ (b) $p = 0.1$, $\kappa = 0.2$ (c) $p = 0.2$, $\kappa = 0.2$

(d) $p = 0.3$, $\kappa = 0.2$ (e) $p = 0.4$, $\kappa = 0.2$ (f) $p = 0.5$, $\kappa = 0.2$

(g) $p = 0.6$, $\kappa = 0.2$ (h) $p = 0.7$, $\kappa = 0.2$ (i) $p = 0.8$, $\kappa = 0.2$

(j) $p = 0.9$, $\kappa = 0.2$ (k) $p = 1$, $\kappa = 0.2$

Figure C.14: We fix $(N, \mathcal{T}) = (150, 100)$ and use multiplex-case Infomap (i.e., the random walker in equation (2.26) can "relax" to any layer). The sample of "relax-rate" values is $r \in \{0, 0.01, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1\}$. We show plots for $r < 0.1$ with solid blue curves, those for $r > 0.1$ with solid red curves, and that of $r = 0.1$ with a black dashed curve. Colors scale with the value of r from deep blue ($r = 0$) to deep red ($r = 1$). We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \ldots, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.02, \ldots, 0.18, 0.2\}$ and $\kappa = 0.2$. For each pair (p, μ) we generate a multilayer partition and a multilayer network. We perform one run of Infomap for each value of r on each multilayer network. (a,b,c,d,e,f,g,h,i,j,k) Infomap and $p \in \{0, 0.1, \ldots, 0.9, 1\}$.



(a) $p = 0$, $\kappa = 0.2$

(b) $p = 0.1$, $\kappa = 0.2$

(c) $p = 0.2$, $\kappa = 0.2$

(d) $p = 0.3$, $\kappa = 0.2$

(e) $p = 0.4$, $\kappa = 0.2$

(f) $p = 0.5$, $\kappa = 0.2$

(g) $p = 0.6$, $\kappa = 0.2$

(h) $p = 0.7$, $\kappa = 0.2$

(i) $p = 0.8$, $\kappa = 0.2$

(j) $p = 0.9$, $\kappa = 0.2$

(k) $p = 1$, $\kappa = 0.2$

Figure C.15: We fix $(N, \mathcal{T}) = (150, 100)$ and use temporal-case Infomap (i.e., the random walker in equation (2.26) can only "relax" to adjacent layers). The sample of "relax-rate" values is $r \in \{0, 0.01, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1\}$. We show plots for $r < 0.1$ with solid blue curves, those for $r > 0.1$ with solid red curves, and that of $r = 0.1$ with a black dashed curve. Colors scale with the value of r from deep blue ($r = 0$) to deep red ($r = 1$). We generate a multilayer partition using our benchmark in Table 5.1 with $p \in \{0, 0.1, \ldots, 0.9, 1\}$ and a power-law null distribution with exponent $\tau = -1$, the lower bound on the expected community size is $|C|_{\min} = 0.1N$, and the upper bound on the expected community size is $|C|_{\max} = 0.3N$. We generate multilayer networks using the stochastic block model in (5.6) with $\mu \in \{0, 0.02, \ldots, 0.18, 0.2\}$ and $\kappa = 0.2$. For each pair (p, μ) we generate a multilayer partition and a multilayer network. We perform one run of Infomap for each value of r on each multilayer network. (a,b,c,d,e,f,g,h,i,j,k) Infomap and $p \in \{0, 0.1, \ldots, 0.9, 1\}$.

Bibliography

[1] *Code available at http://www.mapequation.org.*

[2] *Code available at http://netwiki.amath.unc.edu/GenLouvain/GenLouvain.*

[3] *Code available at http://netwiki.amath.unc.edu/VisComms.*

[4] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B*, 66(3):409–418, 2008.

[5] C. C. Aggarwal and K. Subbian. Evolutionary network analysis: A survey. *ACM Computing Surveys (CSUR)*, 47(1):10–36, 2014.

[6] C. Aicher, A. Z. Jacobs, and A. Clauset. Learning block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248, 2015.

[7] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9:1981–2014, 2008.

[8] D. Aldous and J. A. Fill. *Reversible Markov Chains and Random Walks on Graphs*. University of California, Berkeley, 2002.

[9] C. J. Alpert and A. B. Kahng. Recent directions in netlist partitioning: a survey. *Integration, the Very Large Scale Integration (VLSI) Journal*, 19(1):1–81, 1995.

[10] C. J. Alpert, A. B. Kahng, and S.-Z Yao. Spectral partitioning with multiple eigenvectors. *Discrete Applied Mathematics*, 90(1-3):3–26, 1999.

[11] L. A. N. Amaral and J. M. Ottino. Complex networks. *The European Physical Journal B*, 38(2):147–162, 2004.

[12] M. Anderberg. *Cluster Analysis for Applications*. Academic Press, New York, 1973.

[13] C. J. Anderson, S. Wasserman, and K. Faust. Building stochastic blockmodels. *Social Networks*, 14:137–161, 1992.

[14] A. Arenas, A. Fernández, and S. Gòmez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10:053039, 2008.

[15] T. Aynaud and J.-L. Guillaume. Static community detection algorithms for evolving networks. *WiOpt'10: Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 84:508–514, 2010.

[16] E. Backer and A. K. Jain. A clustering performance measure based on fuzzy set decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(1):66–75, 1981.

[17] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Boston, 1999.

[18] E.R. Barnes. An algorithm for partitioning the nodes of a graph. *SIAM Journal on Algebraic Discrete Methods*, 3(4):541–550, 1982.

[19] I. Barnett and J.-P. Onnela. Change point detection in correlation networks. *arXiv preprint:1410.0761*, 2014.

[20] D. S. Bassett, E. T. Owens, M. A. Porter, M. L. Manning, and K. E. Daniels. Extraction of force-chain network architecture in granular materials using community detection. *Soft Matter*, 11(14):2731–2744, 2015.

[21] D. S. Bassett, M. A. Porter, N. F. Wymbs, S. T. Grafton, J. M. Carlson, and P. J. Mucha. Robust detection of dynamic community structure in networks. *Chaos*, 23(1:013142), 2013.

[22] D. S. Bassett, N. F. Wymbs, M. A. Porter, P. J. Mucha, J. M. Carlson, and S. T. Grafton. Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences*, 108:7641–7646, 2011.

[23] S. Battiston, J. B. Glattfelder, D. Garlaschelli, F. Lillo, and G. Caldarelli. *The structure of financial networks*. Springer, London, UK, 2010, Chapter 7.

[24] E. T. Bell. Exponential numbers. *The American Mathematical Monthly*, 41(7):411–419, 1934.

[25] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing*, 7(6):6–17, 2002.

[26] M. Berlingerio, F. Pinelli, and F. Calabrese. ABACUS: frequent pAttern mining-BAsed Community discovery in mUltidimensional networkS. *Data Mining and Knowledge Discovery*, 27(3):294–320, 2013.

[27] A. L. Bertozzi and A. Flenner. Diffuse interface models on graphs for classification of high dimensional data. *SIAM Journal on Multiscale Modeling and Simulation*, 10(3):1090–1118, 2012.

[28] P. J. Bickel and A. Chen. A nonparametric view of network models and newman-girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009.

[29] 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*, 10:P10008, 2008.

[30] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308, 2003.

[31] S. Boccaletti, G. Bianconi, R. Criado, C.I. Del Genio, J. Gómez-Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, and M. Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[32] S. Boettcher and A. G. Percus. Optimization with extremal dynamics. *Physical Review Letters*, 86(23):5211–5214, 2001.

[33] L. Bohlin, D. Edler, A. Lancichinetti, and M. Rosvall. *Measuring Scholarly Impact: Methods and Practice.* Springer, Switzerland, 2014.

[34] B. Bollobàs. *Modern Graph Theory.* Springer, New York, 1998.

[35] B. Bollobàs, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Structure and Algorithms*, 31(1):3–122, 2007.

[36] B. Bollobàs and O. Riordan. Random Graphs and Branching Processes, in: *Handbook of large-scale random networks.* Bolyai Society Mathematical Studies, 2009.

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

[38] M. Catanzaro and M. Buchanan. Network opportunity. *Nature Physics*, 9:121–123, 2013.

[39] A. Celisse, J.-J. Daudin, and L. Pierre. Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electronic Journal of Statistics*, 6:1847–1899, 2012.

[40] M. Chen, K. Kuzmin, and B. Szymanski. Community detection via maximization of modularity and its variants. *IEEE Transactions on Computational Social Systems*, 1(1):46–65, 2014.

[41] F. Chung and L. Lu. The average distance in a random graph with given expected degrees. *Proceedings of the National Academy of Sciences*, 99:15879–15882, 2002.

[42] F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, California, 1997.

[43] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[44] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(066111), 2004.

[45] A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structure and Algorithms*, 18(2):116–140, 2000.

[46] R. Cont. Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236, 2001.

[47] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley-Interscience, 2nd ed., New York, 2006.

[48] M. M. Dacarogna, R. Gençay, U. Müller, R. B. Olsen, and O. V. Pictet. *An Introduction to High-Frequency Finance.* Academic Press, San Diego, 2001.

[49] L. Danon, A. Diaz-Guilera, and A. Arenas. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 11:P11010, 2006.

[50] L. Danon, J. Duch, A. Arenas, and A. Diaz-Guilera. Community structure identification, in *Large Scale Structure and Dynamics of Complex Networks: From Information Technology to Finance and Natural Science*. World Scientific Publishing Company, Singapore, 2007.

[51] A. D'Aspremont, O. Banerjee, and L. El. Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(56):56–66, 2008.

[52] J.-C. Delvenne, R. Lambiotte, and L. E. C. Rocha. Diffusion on networked systems is a question of time or structure. *Nature Communication*, 6(7366), 2015.

[53] J.-C. Delvenne, M. T. Schaub, S. N. Yaliraki, and M. Barahona. The Stability of Graph Partition: A Dynamics-Based Framework for Community Detection, in: *Dynamics On and Of Complex Networks, Volume 2*. Springer, New York, 2013.

[54] J.-C. Delvenne, S. N. Yaliraki, and M. Barahona. Stability of graph communities across time scales. *Proceedings of the National Academy of Sciences*, 107(29):12755–12760, 2010.

[55] T. Dinh and M. T. Thai. Finding community structure with performance guarantees in complex networks. *arXiv preprint:1108.4034*, 2011.

[56] M. De Domenico, A. Lancichinetti, A. Arenas, and M. Rosvall. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Physical Review X*, 11(011027), 2015.

[57] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72(027104), 2005.

[58] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the Association for Computing Machinery (ACM)*, 19(2):248–264, 1972.

[59] A. V. Esquivel and M. Rosvall. Compression of flow can reveal overlapping modular organization in networks. *Physical Review X*, 1(021025), 2011.

[60] B. Everitt. *Cluster Analysis.* Wiley, 1st ed., New York, 1974.

[61] P. Expert, T. S. Evans, V. D. Blondel, and R. Lambiotte. Uncovering space-independent communities in spatial networks. *Proceedings of the National Academy of Sciences*, 108(19):7663–7668, 2011.

[62] D. J. Fenn. *Network communities and the foreign exchange market.* PhD Thesis, University of Oxford, 2010.

[63] D. J. Fenn, M. A. Porter, M. McDonald, S. Williams, N. F. Johnson, and N. S. Jones. Dynamic communities in multichannel data: An application to the foreign exchange market during the 2007–2008 credit crisis. *Chaos*, 19:3:033119, 2009.

[64] D. J. Fenn, M. A. Porter, P.J. Mucha, M. McDonald, S. Williams, N. F. Johnson, and N. S. Jones. Dynamical clustering of exchange rates. *Quantitative Finance*, 12(10):1493–1520, 2012.

[65] D. J. Fenn, M. A. Porter, S. Williams, M. McDonald, N. F. Johnson, and N. S. Jones. Temporal evolution of financial market correlations. *Physical Review E*, 84(026109), 2011.

[66] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[67] S. E. Fienberg and S. Wasserman. Categorical data analysis of single sociometric relations. *Sociological Methodology*, 12:156–192, 1981.

[68] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

[69] S. Fortunato and M. Barthelémy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2006.

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

[71] J. Frankle and R.M. Karp. Circuit placement and cost bounds by eigenvector decomposition. *Proceedings of the IEEE International Conference on Computer-Aided Design*, pages 414–417, 1986.

[72] A. Ghasemian, P. Zhang, A. Clauset, C. Moore, and L. Peel. Detectability thresholds and optimal algorithms for community structure in dynamic networks. *arXiv preprint:1506.06179*, 2015.

[73] 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.

[74] A. Goldenberg, A. X. Zheng, S. E. Feinberg, and E. M. Airoldi. A survey of statistical network structures. *Foundations and Trends in Machine Learning*, 2:1–117, 2009.

[75] S. Gómez, P. Jensen, and A. Arenas. Analysis of community structure in networks of correlated data. *Physical Review E*, 80(016114), 2009.

[76] S. González-Bailón, J. Borge-Holthoefer, A. Rivero, and Y. Moreno. The dynamics of protest recruitment through an online network. *Scientific Reports*, 1(197), 2011.

[77] B. H. Good, Y-A de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(046106), 2010.

[78] C. Granell, R. K. Darst, A. Arenas, S. Fortunato, and S. Gòmez. A benchmark model to assess community structure in evolving networks. *arXiv preprint:1501.05808*, 2015.

[79] C. Granell, S. Gòmez, and A. Arenas. Unsupervised clustering analysis: A multiscale complex networks approach. *International Journal of Bifurcation and Chaos*, 22(7), 2011.

[80] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(025101), 2004.

[81] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 443:895–900, 2005.

[82] K. M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17:219–229, 1970.

[83] Q. Han, K. S. Xu, and E. M. Airoldi. Consistent estimation of dynamic and multi-layer block models. *arXiv preprint:1410.8597*, 2015.

[84] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79:191–215, 1997.

[85] T. Hartmann, A. Kappes, and D. Wagner. Clustering evolving networks. *arXiv preprint: 1401.3516*, 2014.

[86] T. Hoffmann, M.A. Porter, and R. Lambiotte. Generalized master equations for non-poisson dynamics on networks. *Physical Review E*, 86(4), 2012.

[87] J. M. Hofman and C. H. Wiggins. A bayesian approach to network modularity. *Physical Review Letters*, 100(258701), 2008.

[88] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: first steps. *Social Networks*, 5:109–137, 1983.

[89] P. Holme. Modern temporal network theory: A colloquium. *arXiv preprint:1508.01303*, 2015.

[90] P. Holme and J. Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012.

[91] J. Hopcroft, O. Khan, B. Kulli, and B. Selman. Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences*, 101:5250–5253, 2004.

[92] H. Hu, T. Laurent, M. A. Porter, and A. L. Bertozzi. A method based on total variation for the network modularity optimization using the MBO scheme. *SIAM Journal on Applied Mathematics*, 76(6):2224–2246, 2013.

[93] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[94] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40:1098–1102, 1952.

[95] S. Hutchings. The behaviour of modularity-optimizing community detection algorithms. *University of Oxford, Master dissertation*, 2011.

[96] P. Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la societé vaudoise des sciences naturelles*, 37:547–579, 1901.

[97] A. Z. Jacobs and A. Clauset. A unified view of generative models for networks: models, methods, opportunities, and challenges. *arXiv preprint :1411.4070*, 2014.

[98] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, 1999.

[99] M. Jerrum and G.B. Sorkin. The metropolis algorithm for graph bisection. *SIAM Journal on Discrete Mathematics*, 82(1-3):155–175, 1998.

[100] L. G. S. Jeub, P. Balachandran, M. A. Porter, P. J. Mucha, and M. W. Mahoney. Think locally, act locally: The detection of small, medium-sized, and large communities in large networks. *Physical Review E*, 91:012821, 2014.

[101] I. T. Jolliffe. *Principal Component Analysis.* Springer, New York, 1986.

[102] I. S. Jutla, L. G. S. Jeub, and P. J. Mucha. A generalized Louvain method for community detection implemented in MATLAB, version 2.0. *Code available at http://netwiki.amath.unc.edu/GenLouvain/GenLouvain*, 2011–2014.

[103] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15, 1989.

[104] B. Karrer, E. Levina, and M. E. J. Newman. Robustness of community structure in networks. *Physical Review E*, 77(046119), 2008.

[105] B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(016107), 2001.

[106] T. Kawamoto and M. Rosvall. Estimating the resolution limit of the map equation in community detection. *Physical Review E*, 91(012809), 2015.

[107] A. Kehagias. Bad communities with high modularity. *arXiv preprint:1209.2678*, 2012.

[108] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[109] M. Kivelä, A. Arenas, M. Barthélemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[110] A. Kraskov, H. Stögbauer, R.G. Andrzejak, and P. Grassberger. Hierarchical clustering using mutual information. *Europhysics Letters*, 70(2):278–284, 2005.

[111] W. H. Kruskal. Ordinal measures of association. *Journal of the American Statistical Association*, 53(284):814–861, 1958.

[112] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sky, L. Zdeborová, and P. Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

[113] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*, 2(1-2):83–97, 1955.

[114] R. Lambiotte. Multi-scale modularity in complex networks. *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks(WiOpt)*, pages 546–553, 2010.

[115] R. Lambiotte. Multi-scale modularity and dynamics in complex networks, in: *Dynamics On and Of Complex Networks, Volume 2.* Springer, New York, 2013.

[116] R. Lambiotte, J.-C. Delvenne, and M. Barahona. Random walks, markov processes and the multiscale modular organization of complex networks. *Transactions on Network Science and Engineering*, 1(2):76–90, 2015 (arXiv:0812.1770, 2008).

[117] A. Lancichinetti and S. Fortunato. Benchmark graphs for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(016118), 2009.

[118] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(056117), 2009.

[119] A. Lancichinetti and S. Fortunato. Consensus clustering in complex networks. *Scientific Reports*, 2(336), 2012.

[120] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(046110), 2008.

[121] A. Lancichinetti, F. Radicchi, and J. J. Ramasco. Statistical significance of communities in networks. *Physical Review E*, 82(046110), 2010.

[122] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PLoS ONE*, 6(4):e18961, 2011.

[123] E. Langford, N. Schwertman, and M. Owens. Is the property of being positively correlated transitive? *The American Statistician*, 55(4):322–325, 2001.

[124] P. Latouche, E. Birmelé, and C. Ambroise. Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics*, 5(1):309–336, 2011.

[125] J. Leskovec, K. J. Lang, A. Dasgupta, and M. Mahoney. Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2011.

[126] Annick Lesne. Shannon entropy: A rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics. *Mathematical Structures in Computer Science*, 24(3), 2014.

[127] A. C. F. Lewis, N. S. Jones, M. A. Porter, and C. M. Deane. The function of communities in protein interaction networks at multiple scales. *BMC Systems Biol.*, 4:100, 2010.

[128] M. Li, J. H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang. An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2):149–154, 2001.

[129] T. W. Liao. Clustering of time series data — a survey. *Pattern Recognition*, 38:1857–1874, 2005.

[130] L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdős is Eighty*, 2(1):1–46, 1993.

[131] M. MacMahon and D. Garlaschelli. Community detection for correlation matrices. *Physical Review X*, 5(021006), 2015.

[132] K. T. Macon, P. J. Mucha, and M. A. Porter. Community structure in the United Nations General Assembly. *Physica A*, 391:343–361, 2012.

[133] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–297, 1967.

[134] R. N. Mantegna. Hierarchical structure in financial markets. *The European Physical Journal B*, pages 193–197, 1999.

[135] B. D. McKay and F. Skerman. Modularity in random regular graphs and lattices. *Electronic Notes in Discrete Mathematics*, 43:431–437, 2013.

[136] M. Meila. Comparing clusterings: an axiomatic view. *Proceedings of the 22nd International Conference on Machine Learning*, pages 577–584, 2005.

[137] M. Meila. Comparing clusterings — an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[138] G. Meissner. *Correlation risk modeling and management.* John Wiley and Sons, Singapore, 2014.

[139] B. G. Mirkin. *Mathematical Classification and Clustering.* Kluwer Academic Press, Dordrecht, 1996.

[140] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. Community structure in time-dependent, multiscale and multiplex networks. *Science*, 328:876–878, 2010.

[141] P.J. Mucha and M. A. Porter. Communities in multislice voting networks. *Chaos*, 20:4:041108, 2010.

[142] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[143] R. R. Nadakuditi and M. E. J. Newman. Graph spectra and the detectability of community structure in networks. *Physical Review Letters*, 108(188701), 2012.

[144] M. E. J. Newman. Analysis of weighted networks. *Physical Review E*, 70(066146), 2004.

[145] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(066133), 2004.

[146] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(036104), 2006.

[147] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, Oxford, UK, 2010.

[148] M. E. J. Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8:25–31, 2012.

[149] M. E. J. Newman. Community detection and graph partitioning. *Europhysics Letters*, 103(28003), 2013.

[150] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(026113), 2004.

[151] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *PNAS*, 104(23), 2007.

[152] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *J. Am. Stat. Assoc.*, 96:1077–1087, 2001.

[153] J.-P. Onnela. *Master's thesis: Taxonomy of Financial Assets*. Helsinky University of Technology, 2006.

[154] J.-P. Onnela, A. Chakraborti, K. Kaski, J. Kertész, and A. Kanto. Dynamics of market correlations: taxonomy and portfolio analysis. *Physical Review E*, 68:5:056110, 2003.

[155] G. Ormane, V. Labatut, and H. Cherifi. Towards realistic artificial benchmark for community detection algorithms evaluation. *Internet Journal of Web Based Communities*, 9(3):349–370, 2013.

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

[157] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[158] S. Paul and Y. Chen. Community detection in multi-relational data with restricted multi-layer stochastic blockmodel. *arXiv preprint:1506.02699*, 2015.

[159] L. Peel and A. Clauset. Detecting change points in the large-scale structure of evolving networks. *To appear in Applied Artificial Intelligence (arXiv preprint:1403.0989)*, 2015.

[160] T. P. Peixoto. Inferring the mesoscale structure of layered, edge-valued and time-varying networks. *arXiv preprint:1504.02381*, 2015.

[161] G. Pietri and P. Expert. Temporal stability of network partitions. *arXiv preprint:1404.7170*, 2014.

[162] V. Plerou, P. Gopikrishman, B. Rosenow, L. A. N. Amaral, T. Guhr, and H. E. Stanley. Random matrix approach to cross correlations in financial data. *Physical Review E*, 65(066126), 2002.

[163] M. A. Porter, J.-P Onnela, and P. J. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56:1082–1097, 1164–1166, 2009.

[164] M. Potters, J.-P. Bouchaud, and L. Laloux. Financial applications of random matrix theory: Old laces and new pieces. *Acta Physica Polonica B*, 36:2767–2784, 2005.

[165] M. Pourahmadi. Covariance estimation: The GLM and regularization perspectives. *Statistical Science*, 26(3):369–387, 2011.

[166] F. Radicchi and A. Arenas. Abrupt transition in the structural formation of interconnected networks. *Nature Physics*, 9:717–720, 2013.

[167] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(016110), 2006.

[168] J. Reichardt and D. R. White. Role models for complex networks. *The European Physical Journal B*, 60:217–224, 2007.

[169] T. Richardson, P. J. Mucha, and M. A. Porter. Spectral tripartitioning of networks. *Physical Review E*, 80(036111), 2009.

[170] M. Rosvall, D. Axelsoon, and C. T. Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009.

[171] 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.

[172] 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.

[173] M. Rosvall and C. T. Bergstrom. Multilevel compression of random walks on networks reveal hierarchical organization in large integrated systems. *PloS ONE*, 6(4):e18209, 2011.

[174] M. Sales-Pardo, R. Guimerà, A. A. Moreira, and L. A. N. Amaral. Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39):15224–15229, 2007.

[175] M. Sarzynska, E. A. Leicht, G. Chowell, and M. A. Porter. Null models for community detection in spatially-embedded, temporal networks. *To appear in: Journal of Complex Systems (arXiv preprint:1407.6297)*, 2015.

[176] E.N. Sawardecker, M. Sales-Pardo, and L.A.N. Amaral. Detection of node group membership in networks with group overlap. *The European Physical Journal B*, 67(3):277–284, 2009.

[177] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[178] R. Schafer and T. Guhr. Local normalization: Uncovering correlations in non-stationary financial time series. *Physica A*, 389:3856–3865, 2010.

[179] M. T. Schaub, R. Lambiotte, and M. Barahona. Encoding dynamics for multiscale community detection: Markov time sweeping for the map equation. *Physical Review E*, 86(026112), 2012.

[180] C. R. Shalizi. *Methods and Techniques of Complex Systems: An Overview.* Kluwer, London, 2003.

[181] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[182] J. Shi and J. Malik. Normalized cut and image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[183] H. A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 2008.

[184] S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich. Network modelling methods for FMRI. *NeuroImage*, 54(2):875–891, 2011.

[185] T. A. B. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997.

[186] N. Stanley, S. Shai, D. Taylor, and P. J. Mucha. Clustering network layers with the strata multilayer stochastic block model. *arXiv preprint:1507.01826*, 2015.

[187] M. Takaffoli, R. Rabbany, and O. R. Zaïane. Incremental local community identification in dynamic social networks. *IEEE International Conference on Advances in Social Networks Analysis and Mining*, pages 90–94, 2013.

[188] D. Taylor, S. A. Myers, A. Clauset, M. A. Porter, and P. J. Mucha. Eigenvector-based centrality measures for temporal networks. *preprint arXiv:1507.01266*, 2015.

[189] V. A. Traag and J. Bruggeman. Community detection in networks with positive and negative links. *Physical Review E*, 80(036115), 2009.

[190] V. A. Traag and P. Van Dooren. Narrow scope for resolution-limit-free community detection. *Physical Review E*, 84(1:016114), 2011.

[191] 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*, 53:526–543, 2011.

[192] L. N. Trefethen. *Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[193] T. Vallès-Catalá, F. A. Massucci, R. Guimerà, and M. Sales-Pardo. Multilayer stochastic block models reveal the multilayer structure of complex networks. *arXiv preprint:1410.8597*, 2014.

[194] R. Vershynin. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25:655–686, 2012.

[195] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks. *arXiv preprint: 0702048*, 2007.

[196] G. Wang, Y. Shen, and M. Ouyang. A vector partitioning approach to detecting community structure in complex networks. *Computers and Mathematics with Applications*, 55:2746–2752, 2008.

[197] Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

[198] S. Wasserman and C. Anderson. Stochastic a posteriori blockmodels: construction and assessment. *Social networks*, 9:1–36, 1987.

[199] S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. *Proceedings of the 5th SIAM International Conference on Data Mining*, 2005.

[200] S. Williams. Risk on risk off will test investors through 2011. *Financial Times*, 2011.

[201] P. Wilmott, S. Howison, and J. Dewynne. *The mathematics of financial derivatives: A student introduction.* Cambridge University Press, 1995.

[202] Y. Wu, C. Zhou, J. Xiao, J. Kurths, and H.J. Schellnhuber. Evidence for a bimodal distribution in human communication. *Proceedings of the National Academy of Sciences*, 107(44):18803–18808, 2010.

[203] K. Xu and A. Hero. Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal on Special Topics in Signal Processing*, 8(4):552–562, 2014.

[204] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 2005.

[205] X. Yan, J. E. Jensen, F. Krzakala, C. Moore, C. R. Shalizi, L. Zdeborovà, P. Zhang, and Y. Zhu. Model selection for degree-corrected block models. *Journal of Statistical Mechanics*, 5(P05007), 2014.

[206] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2001.

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

[208] A. Zalesky, A. Fornito, and E. Bullmore. On the use of correlation as a measure of network connectivity. *NeuroImage*, 60:2096–2106, 2012.

[209] X. Zhang and M. E. J. Newman. Multiway spectral community detection in networks. *arXiv preprint:1507.05108*, 2015.

[210] Q. Zhao, Y. Tian, Q. He, N. Oliver, R. Jin, and W.-C. Lee. Communication motifs: A tool to characterize social communications. *In Proceedings of the 19th ACM international conference on Information and knowledge management*, 84:1645–1648, 2010.