Networks, Communities, and Consumer Behaviour

Lucas G. S. Jeub
Somerville College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity 2015

Abstract

Networks are an abstract representation of connections (the "edges") between entities (the "nodes"). One can represent many different types of data in this way, including many social, biological, technological and physical systems. Examples we discuss in this thesis include networks of friendship ties between individuals on Facebook, coauthorship networks between scientists, and similarities in voting patterns between members of the US Congress. Analysing intermediate-sized (or "meso-scale") features often reveals insights about a network's structure and function. A particular type of meso-scale feature are "communities", where one typically thinks of a community as a set of nodes that is particularly "well-connected" internally but has "few" connections to other nodes in a network. A complementary interpretation of a community is as a set of nodes that "trap" a diffusion-like dynamical process for a "long" time. Based on this dynamical interpretation, we investigate "size-resolved community structure" in networks by identifying bottlenecks of locally-biased dynamical processes that start at seed sets of nodes. By sampling many different local communities for different seeds and different strengths of the locality bias of the dynamical process, we obtain a picture of the way communities at different size scales compare in a network. This "size-resolved community structure" provides a signature of community structure in a network and its qualitative features are related to the way local communities combine to form the larger scale structure of a network. For many data sets, ordinary networks are not sufficient to represent the detailed connectivity patterns. For example, connections often evolve over time and one may have different types of connections between the same entities. Multilayer networks provide a framework to represent these different types of situations. The perspective of communities as bottlenecks to dynamical processes extends in a natural way to multilayer networks and we use it to illustrate that two types of random walk on a multilayer network that have been used as the basis for identifying communities in a multilayer network correspond to very different notions of what it means for a set of nodes to be a good multilayer community. This exemplifies the need for multilayer benchmark networks with known community structure to compare the ability of different methods to identify intuitive community structure. We propose a method for generating benchmark networks with general multilayer structure and use it as the basis for a preliminary comparison of different multilayer community detection methods. Finally, we use multilayer community detection to analyse survey data about people's perception of their hair. One key advantage of this type of data compared to most traditional network data sets is that we have a large number of potential explanatory variables that we can use to interpret the results of identifying communities which allows us to identify some potentially interesting hypothesis.

Acknowledgements

I would like to thank my supervisor Dr. Mason Porter for his support and guidance without which none of this work would have been possible.

I am grateful to EPSRC and Unilever for supporting this project under the EPSRC CASE studentship No. BK/10/039.

Publications and Software

Publications

- Jeub, L. G. S., Balachandran, P., Porter, M. A., Mucha, P. J., and Mahoney, M. W. (2015). Think locally, act locally: Detection of small, medium-sized, and large communities in large networks. *Physical Review E*, 91:012821.
- Jeub, L. G. S., Mahoney, M. W., Mucha, P. J., and Porter, M. A. (2015). A local perspective on communities in multilayer networks. arXiv:1510.05185 [cs.SI]
- Bazzi, M., Jeub, L. G. S., Arenas, A., Porter, M. A., and Howison, S. D. (in prep.). Multilayer benchmark networks for community detection.

Software

- GenLouvain: Jutla, I. S., Jeub, L. G. S., and Mucha, P. J., A generalized Louvain method for community detection implemented in MATLAB, available at `http://netwiki.amath.unc.edu/GenLouvain`
- LocalCommunities: Jeub, L. G. S., available at `https://github.com/LJeub/LocalCommunities`
- SpringVisCom: Jeub, L. G. S., available at `https://github.com/LJeub/SpringVisCom`

Statement of Originality

The research in this thesis is a result of collaboration between myself and my coauthors on the listed publications. 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.

List of Definitions

Contents

Chapter 1

Introduction

Many physical, technological, biological, and social systems can be modelled as networks, which in their simplest form are represented by graphs. A (static and single-layer) graph consists of a set of entities (called "vertices" or "nodes") and pairwise interactions (called "edges" or "links") between those vertices [56, 91, 153]. Graphical representations of data have led to numerous insights in the natural, social, and information sciences; and the study of networks has in turn borrowed ideas from all of these areas [14].

Although numerous tools have been developed for the analysis of networks [153], most of them concentrate on static networks with only a single type of tie between entities. Such ordinary networks are often unable to capture the complex interactions among entities in the real world. In general, interactions (and the entities themselves) can change over time, and there can also be different types of interactions between the same pair of entities. Temporal networks allow one to consider the former situation [87], and multiplex networks allow one to consider the latter [204]. By using "multilayer networks" [24, 103], it is possible to represent either temporal networks or multiplex networks, as well as other more general types of structures. In the former case, each layer represents a time or a time window. In the latter case, each layer represents a type of interaction. (Indeed, one can represent a multiplex temporal network in the multilayer framework.)

In general, networks can be described using a combination of local, global, and

"meso-scale" perspectives. Local properties are functions of single nodes and their immediate neighbourhood and include properties such as a nodes "degree", i.e., how many neighbours it has, and local "clustering coefficient" which measure how many neighbours of a node are also neighbours of each other. Global properties are concerned with the network as a whole. They include different measures of the size of a network, such as the number of nodes, the number of edges, and the "diameter", i.e., the longest distance between any pair of nodes in a network. Another important global property of a network is its "density", where one is often interested in sparse networks which have few edges compared to the total number of pairs of nodes.

In this thesis we are particularly interested in meso-scale properties of networks. Meso-scale properties arise at intermediate size scales in a network and are the result of interactions between local neighbourhoods of different nodes. One type of meso-scale structure are "communities", i.e. groups of nodes that are relatively densely connected to other nodes in the same group but have relatively few connections to nodes outside the group. Communities are probably the type of meso-scale structure that has received the most attention in the networks literature and many different methods have been developed to detect communities in networks [61, 69, 167]. Perhaps the main reason that community structure has received such a large amount of attention compared to other meso-scale structures, such as "core-periphery structure" [25, 44, 174] where we have a set of "core" nodes which are responsible for most of the connections in the network and a set of "peripheral" nodes which are sparsely connected and mostly interact via the core, is that the presence of strong community structure in a network has some highly desirable intuitive properties. As nodes in a good community have little interactions with nodes in other communities, one might hope to be able to analyse communities individually without considering other parts of a network. From a dynamics perspective, this corresponds to the idea of "time scale separation" [160], where at short time scales a dynamical process remains trapped within a community and one can analyse its behaviour within the community while ignoring the other parts of the network, whereas at long time scales one can assume that the process is essentially well-mixed within communities, such

that one can approximate the behaviour of the process by only considering transitions between communities. This link between the behaviour of dynamical processes and community structure is the basis for many popular methods to detect communities [61, 111, 176], where one often uses a random walk as a parsimonious model for information diffusion.

One can investigate community structure in a network both from a global perspective and a local perspective. From a global perspective, community structure corresponds to partitioning the nodes of a network into (potentially overlapping) sets such that most of the edges in a network should be between nodes in the same set. This perspective, often implicitly, entails the assumption that the adjacency matrix of a network can be reasonably approximated by a block-diagonal matrix, i.e., a "block-diagonal view of the world". From a local perspective on community structure, one tries to identify communities that are centred around a particular seed node. From the dynamical interpretation of community structure this corresponds to trying to identify a set of nodes that traps a dynamical process that *starts at the seed node* at short time scales before it explores other parts of a network. This local perspective can accommodate pervasive overlaps between communities in a natural way as local communities for different seed nodes can share many nodes even if the two seed nodes do not belong to each others local community. As a result, the large scale structure one obtains by amalgamating local communities does not necessarily resemble a global block-diagonal structure even in networks that have good local communities. Moreover, taking a local perspective on community structure is also consistent with real-world experience of individuals, such as users of Facebook [70, 201], who experience the global organisation of a social network only through their personal neighbourhood.

Viewed from this perspective, the computation of network community structure depends fundamentally on three things: (1) actual network structure, (2) the dynamics or application of interest, and (3) the initial conditions or network region of interest. The perspective in point (2) contrasts with the prevalent view of community structure as arising simply from network structure [61, 167], but it is consistent with

the notion of dynamical systems depending fundamentally on their initial conditions, and it is crucial in many applications (e.g., both social [35, 36] and biological contagions [27, 71, 108]). In Chapter 3 we use the behaviour of random walks and geodesic spreading processes to investigate local community structure in different types of networks. We focus in particular on the way local communities of different sizes compare in a given network. The main tool we use for this comparison is the "network community profile" introduced in [127, 128], which measures the community quality of the best community of a given size as a function of community size. As we see in Chapter 3, size-resolved community structure differs substantially between different types of networks and is qualitatively consistent between networks of the same type. Furthermore, the qualitative features of the network community profile reveal how local communities in a network combine to form the larger structure of a network. Using "conductance" (which is intimately related to the problem of characterising the mixing rates of random walks [93]) as the measure of community quality,we roughly distinguish three types of qualitative behaviour which we link to different types of large-scale organisation of a network:

- Small and large communities are of comparable quality, indicating either essentially uniform connections between all nodes in the case where there are no good communities or uniform connections between communities if there are good communities.

- Community quality increases with size, indicating roughly nested community structure which can arise in particular as a result of low-dimensional spatial constraints.

- Community quality decreases with size, indicating a nested core-periphery structure.

Given the direct link between conductance and mixing of random walks [147] these different situations have direct implications for information diffusion on these different types of networks (given a random walk is a reasonable approximation to

the more complicated processes taking place on real networks). One can also apply a similar procedure for other types of dynamical processes [68] which may be a more appropriate model of information diffusion in some situation.

Given that mathematically tractable definitions of community quality tend to be too restrictive to be useful in practical applications [167], experimental validation of community detection methods is extremely important. One possible means of validation is to compare algorithmically-obtained communities with known "ground truth" communities. However, notions of ground truth can be weak in large networks [128, 199, 210], and one rarely possesses even a weak notion of ground truth for most networks. Indeed, in many cases, one should not expect a real (or realistic) large network to possess a single feature that (to leading order) dominates large-scale latent structure in a network. Thus, comparing the output of community-detection algorithms to "ground truth" in practise is most appropriate for obtaining coarse insights into how a network might be organised into social or functional groups of nodes [200]. Instead, validation of community detection algorithms relies mostly on artificial benchmark networks with known structure. We show that the popular LFR benchmark networks [114, 118] have a simplified community structure that does not reproduce the more intricate features of community structure in real networks. This is an important point to keep in mind when translating results from benchmark networks as good performance on the simplified structure of the benchmark networks may not translate to good performance on the more intricate structure of real networks.

The perspective of communities as bottlenecks to dynamical processes on networks extends in a natural way to multilayer networks by considering dynamical processes that traverse both intralayer and interlayer edges. The rich structure of multilayer networks allows one to generalise diffusion-like dynamics in different ways which differ in the way they spread between layers. On the same network, these different diffusion-like processes can show radically different behaviour and hence lead to different conclusions about community structure. We illustrate this effect using two different definitions of a random walk on a multilayer network that were used to

motivate multilayer community detection algorithms [48, 149] in Section 3.6.

To probe this type of behaviour in more detail, artificial benchmark networks with known structural properties are an important tool. For the single-layer case, a popular family of benchmark networks with known community structure are the LFR networks [114, 118]. LFR networks reflect some stylised properties of real networks, namely heterogeneous degree and community-size distributions. Even though, as we show in Section 3.5, they do not reflect the intricate structure of real networks, they are nonetheless an invaluable tool for comparing the behaviour of different community detection methods. For multilayer networks, similar benchmarks only exist for special cases (temporal networks [81, 183] and multiplex networks with blocks of layers that have identical community structure [48]). We introduce an approach to generate multilayer benchmark networks with more general correlations between community structure in different layers in Chapter 4. We use a modular approach, where we assume that the community assignment for a node in a layer is the result of a noisy copying process, where at each iteration a node in a given layer has some probability to copy the community assignment of the same node in an adjacent layer. Iterating this process, we can generate community structure where we control both the amount of correlation between communities in different layers as well as the structure of the underlying dependencies. We then use this community structure to sample edges between nodes in each layer using a generative network model that has the prescribed community structure. The model we discuss in Chapter 4 produces networks that have very similar structure to the LFR networks, though we plan to use this approach with different network models in the future. We use this benchmark to compare multilayer modularity optimisation [149] and multilayer InfoMap [48] and find that multilayer modularity optimisation can exploit correlations between layers successfully to identify the planted partition, whereas multilayer InfoMap in its current implementation does not.

Community detection is intimately related to the problem of data clustering, where one tries to identify groups of data points that are close in some sense [92]. For single-layer networks this connection has been exploited to successfully identify

clusters in artificial and real-world data sets [45, 47, 82]. We use a similar approach to explore clusters in survey data on hair characteristics, using multilayer networks to obtain a richer representation of the underlying data in Chapter 5. The connections in each layer are based on computing similarities between participants based on a related subset of questions. We discuss different types of similarity measures and compare the resulting networks using both the local methods we discuss in Chapter 3 and multilayer modularity optimisation to identify communities. However, we obtain broadly comparable results for different similarity measures. We combine the results from different similarity measures using a consensus clustering approach (see Section 2.5.2) to obtain a consensus partition of the multilayer networks which we analyse in more detail. One of the advantages of this type of data set compared to most traditional network data sets that are available, is that we have a large number of potential explanatory variables to interpret the results of the community detection procedure. In Chapter 5 we use multinomial logistic regression with a lasso penalty [63, 195] to relate community structure to the explanatory variables. This effectively converts the task of interpreting community structure into a classification problem, where we try to predict community assignment based on the available data. This approach should also be applicable to other data sets where we have a weak notion of "ground truth". For the survey data it suggests some interesting hypothesis about peoples perception of their hair.

Chapter 2

Background

This chapter introduces key concepts and notation that we use throughout this thesis.

2.1 Networks

A network consists of a set of entities and the connections between them. The concept of a network is useful for describing many different systems in a variety of fields. Examples include social networks, where the entities can be people or animals and connections could for example represent friendship. Other examples include food webs in ecology, networks of neurons in the brain, road networks, the internet, and many more [153].

2.1.1 Definitions and Notation

The *topology* of a network refers to the pattern of its underlying connections, where nodes are unlabelled and only the presence or absence of a connection for each pair of nodes is recorded. The *structure* of a network additionally accounts for any other information about the nodes and connections that may be available, e.g. varied strength of connections or different types of nodes. Its topology associates a network with a corresponding graph. A *graph* $G(V, E)$ consists of a set of *nodes* (also called vertices) V and a set of *edges* $E \subset V \times V$ encoding the connections between nodes. Edges are directed, where the ordered pair $(i, j) \in E$ indicates the presence of a

directed edge from i (the *initial node* of the edge) to j (the *terminal node*), and we say that node i is *adjacent* to node j if $(i,j) \in E$. A graph is undirected if $(i,j) \in E \Rightarrow (j,i) \in E$. For undirected graphs one usually replaces the pair of directed edges (i,j) and (j,i) by a single undirected edge $\{i,j\}$, such that $E \subset 2^V$. One would often like to include more features of the structure of a network in its analysis, rather than just its topology. One of the most common extensions is to allow edges to have an associated *weight*. We define edge weights using a weight function $w : E \to \mathbb{R}$, where $w(i,j)$ is the weight of the edge (i,j). For unweighted networks we define $w(i,j) = 1$ for all $(i,j) \in E$ for convenience.

An alternative representation of a graph is an adjacency matrix. The *adjacency matrix* of a graph $G = (V,E)$ with vertex set $V = \{1,\dots,n\}$, $n = |V|$, is given by the $n \times n$ matrix $\boldsymbol{A}$ with entries

$$\boldsymbol{A}_{ij} = \begin{cases} w(j,i), & (j,i) \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

The above definition allows edges to have a direction, where the ordered pair $(i,j) \in E$ indicates the presence of a directed edge from i to j, and hence $\boldsymbol{A}_{ij} \neq 0$ indicates the presence of an edge directed from j to i.

Degree and Strength

The degree and strength of a node are two of the most fundamental local properties in a network. The *degree* of a node is the number of edges incident at the node. For an undirected network with unweighted adjacency matrix $\boldsymbol{A}$, the degree d_i of node i is given by

$$d_i = \sum_{j=1}^{n} \boldsymbol{A}_{ij} = \sum_{j=1}^{n} \boldsymbol{A}_{ji}.$$

In a directed network, a node has distinct in-degree d_i^{in} and out-degree d_i^{out} given, respectively, by the number of inbound and outbound edges at the node,

$$d_i^{\text{in}} = \sum_{j=1}^{n} \boldsymbol{A}_{ij}, \qquad d_i^{\text{out}} = \sum_{j=1}^{n} \boldsymbol{A}_{ji}.$$

The *strength* of a node in a weighted network is the total weight of the edges incident at the node. In a directed network, each node has separate in- and out-strength, corresponding to the total weight of incoming and outgoing edges, respectively. The strength of a node can be calculated using the same formulas as for degree by replacing the unweighted adjacency matrix by its weighted counterpart. We denote the strength, in-strength, and out-strength of a node by k, k^{in}, and k^{out} respectively.

Laplacian and Normalised Laplacian

Other than the adjacency matrix, we frequently use two other matrices to encode the connectivity structure of a network, the *combinatorial Laplacian* $\boldsymbol{L}$ with entries

$$\boldsymbol{L}_{ij} = k_i \delta(i,j) - \boldsymbol{A}_{ij}, \tag{2.2}$$

and the *normalised Laplacian* $\mathcal{L}$ with entries

$$\mathcal{L}_{ij} = 1 - \frac{\boldsymbol{A}_{ij}}{\sqrt{k_i k_j}}. \tag{2.3}$$

Note that δ is the *delta function (Kronecker delta)*, where

$$\delta(i,j) = \begin{cases} 1, & i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

Using the diagonal *degree matrix* $\boldsymbol{D}$ with entries $\boldsymbol{D}_{ij} = \delta(i,j)k_i$, we can write $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ and $\mathcal{L} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}}$.

Paths, Cliques, and Components

A *path* P in G is a sequence $P = \{(i_t, , j_t)\}_{t=1}^{s} \subset E$ of edges , such that $i_{t+1} = j_t$ for $t \in \{1, \ldots, s-1\}$, and $i_t \neq i_{t'}$ for all $t \neq t' \in \{1, \ldots, s\}$ (i.e., a path does not visit a node more than once). The length of path P is $|P| = \sum_{(i,j)\in P} l((i,j))$, where $l : E \to \mathbb{R}_{\geq 0}$ is a length function, such that $l((i,j))$ is the length of the edge (i,j). For an unweighted network, $l((i,j)) = 1$ for all edges. For weighted networks, where $w((i,j))$ is conventionally a measure of closeness of the tie between nodes i and j, we define $l((i,j)) = \frac{1}{|w((i,j))|}$. We use $\mathcal{P}_{ij}$ to denote the set of all paths between nodes i and j, i.e., all paths such that $i_1 = i$ and $j_s = j$.

For a graph $G = (V, E)$, a *clique* is a subgraph $K \subset G$, such that every pair of nodes in K is adjacent in G. A *k-clique* is a clique that contains k nodes, and a *maximal clique* is a clique that cannot be enlarged by adding nodes[1].

Two nodes i and j are *weakly connected* if there exists a path from i to j in the underlying undirected network and *strongly connected* if there exists a path from i to j and from j to i that respects edge directions. A graph is weakly (strongly) connected if each pair of nodes is weakly (strongly) connected.

The *weakly (strongly) connected component* associated with a node a is the set of all nodes that are weakly (strongly) connected to a. Weak and strong connectedness are equivalence relations, so a graph can be decomposed into disjoint equivalence classes of weakly and strongly connected components. In the case of undirected graphs, weak and strong connectedness are equivalent, and one simply speaks of "connected" nodes and components.

Geodesic Distance, Neighbourhoods, and Ego Networks

The *geodesic distance* $\Delta(i,j) = \min_{P\in\mathcal{P}_{ij}} |P|$ between two nodes i and j is the length of a shortest path between i and j. The *k-neighbourhood* $N_k(i) = \{j \in V : \Delta(i,j) \leq k\}$

[1]The original definition of a clique [136, 137] includes maximality as part of the definition. Dropping the maximality condition from the definition is often convenient and common practise. Ref. [136] also introduces the notion of an n-clique as a maximal set of nodes such that each pair of nodes in the set is connected by a paths of length at most n. This should not be confused with the common use of k-clique as shorthand for a clique of size k which we adopt in this thesis.

of i is the set of all nodes that are at most a distance k away from i. The *k-neighbourhood* of a set of nodes S is $N_k(S) = \bigcup_{i \in S} N_k(i)$. The *egocentric network* (i.e., *ego network* or *ego-net*) [204] of a node (the ego) is the subgraph induced by the neighbourhood of the ego. That is, the ego-net is the network that consists of all nodes that are in the ego's 1-neighbourhood and all edges between those nodes. We use the term *k-ego-net* for the subgraph induced by the k-neighbourhood of a node. The traditional definition of an ego-net excludes the ego and its edges, but we specifically include them.

Connection Density

The *density* ρ of an unweighted network $G(V, E)$ with $n = |V|$ nodes and adjacency matrix $\boldsymbol{A}$ is

$$\rho(G) = \frac{\sum_{i,j=1}^{n} \boldsymbol{A}_{ij}}{n(n-1)}, \tag{2.5}$$

which gives the fraction of edges present in the network, compared to the maximum possible number of edges.

Bipartite Networks

A *bipartite network* $G_B(V_1, V_2, E)$ is a network with two distinct classes of nodes, where an edge always links a pair of nodes from different classes, i.e., $V_1 \cap V_2 = \emptyset$ and $E \subset V_1 \times V_2 \cup V_2 \times V_1$. As a result of the restriction on the possible edges imposed by the bipartite structure, bipartite networks often have to be treated differently from ordinary (unipartite) networks. Often, a convenient way to analyse bipartite networks is to study the one-mode projections. The *unweighted one-mode projections* of a bipartite network are $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, where

$$E_1 = \{(i_1, j_1) : i_1, j_1 \in V_1, \exists j_2 \in V_2 \text{ s.t. } (i_1, j_2), (j_2, j_1) \in E\}$$

and

$$E_2 = \{(i_2, j_2) : i_2, j_2 \in V_2, \exists j_1 \in V_1 \text{ s.t. } (i_2, j_1), (j_1, j_2) \in E\}.$$

By using one-mode projections, one always looses some information about the structure of the underlying bipartite network. In practise, one usually defines weighted projections to retain more of the information of the underlying bipartite network. The co-authorship and voting networks we discuss in Chapter 3 are examples of one-mode projections.

2.2 Multilayer Networks

Multilayer networks provide a unified framework for representing many types of relationships between entities that cannot adequately be represented using ordinary (single-layer) networks. Examples include *temporal networks* [87], where connections between nodes change across time and *multiplex networks*, where there are different types of connections between nodes, and many more [103].

Following the notation of [103], we consider a general *multilayer network* $M(V_M, E_M, V, \mathcal{L})$ with $n = |V|$ nodes and $l = |\mathcal{L}|$ *layers*, where the set of layers may be split into q different *aspects* (e.g. time and connection type), such that $\mathcal{L} = L_1 \times \ldots \times L_q$. The edges $E_M \subseteq V_M \times V_M$ in a multilayer network are between *state nodes* (or *node-layer tuples*) $V_M \subseteq V \times \mathcal{L}$. We sometimes use the term *physical nodes* instead of nodes for the elements of V to distinguish them from the state nodes V_M. A state node $i\boldsymbol{a} \in V_M$ represents the physical node $i \in V$ in layer $\boldsymbol{a} \in \mathcal{L}$, where $\boldsymbol{a} = (a_1, \ldots, a_q)$. We split the edges into *intralayer edges* $E_L = \{(i\boldsymbol{a}, j\boldsymbol{b}) \in E_M : \boldsymbol{a} = \boldsymbol{b}\}$ and *interlayer/coupling edges* $E_C = E_M \setminus E_L$. For a weighted multilayer network, we additionally define a weight function $w : E_M \to \mathbb{R}$, which encodes the edge weights. For an unweighted network we define $w(e) = 1$ for all $e \in E_M$ to simplify the discussion. We encode the connectivity of a multilayer network using an *adjacency tensor* $\boldsymbol{A}$, analogous to the adjacency matrix for single layer networks, where

$$\boldsymbol{A}_{i\boldsymbol{a}}^{j\boldsymbol{b}} = \begin{cases} w((j\boldsymbol{b}, i\boldsymbol{a})), & (j\boldsymbol{b}, i\boldsymbol{a}) \in E_M, \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

We analogously define the *intralayer adjacency tensor* $\boldsymbol{A}_L$ using only the intralayer edges E_L and the *coupling adjacency tensor* $\boldsymbol{A}_C$ using only the coupling edges E_C, such that $\boldsymbol{A} = \boldsymbol{A}_L + \boldsymbol{A}_C$. In many applications only the intralayer edges E_L are explicitly available and one needs to chose the coupling edges E_C to reflect the structure of the data.

In the case of a multilayer network with a single layer (i.e., $l = 1$), the definitions of the adjacency tensor (Eq. (2.6)) and adjacency matrix (Eq. (2.1)) are equivalent (except for implicitly taking the transpose of the adjacency matrix) with $\boldsymbol{A}_i^j = \boldsymbol{A}_{ij}$. Based on the analogy between the adjacency matrix and adjacency tensor, one can generalise many quantities defined for single-layer networks to multilayer networks [20, 43, 49].

Note that $G_M(V_M, E_M)$ defines an ordinary network between the state nodes of the multilayer network, the *flattened network* associated with the multilayer network M. The adjacency matrix of the flattened network is the *supra-adjacency matrix* [49, 50, 76, 103] of the multilayer network. One obtains the supra-adjacency matrix by flattening the adjacency tensor (Eq. (2.6)) of the multilayer network. One then usually eliminates any missing state nodes of the network (i.e., state nodes that have no connections), which correspond to row-column pairs of the supra-adjacency matrix with only zero entries. In the supra-adjacency matrix, layers correspond to diagonal blocks and interlayer connections to off-diagonal blocks. Supra-adjacency matrices are often a convenient way to represent a multilayer network in a compact way, in particular if we are interested in processes that take place at the level of the state nodes, e.g., random walks on multilayer networks as discussed in Section 2.3. However, unlike with the tensor representation, we need to keep track of the layer structure separately.

2.3 Random Walks on Networks

Random walks are often used as a prototypical model for diffusion processes on a network. The discrete-time *unbiased random walk* on a network with adjacency matrix

$\boldsymbol{A}$ is a Markov chain [101] with *transition matrix* $\boldsymbol{P}$, where the probability of reaching node i when starting at node j is given by $\boldsymbol{P}_{ij} = \frac{\boldsymbol{A}_{ij}}{k_j^{\text{out}}}$. One usually motivates this process by considering a random walker that jumps between the nodes of the network, where at each jump it picks one of the neighbours of the current node uniformly at random [134]. Hence, if $\boldsymbol{p}(t)$ is the probability distribution over the nodes of finding the random walker at a particular node at time t, then

$$\boldsymbol{p}_i(t+1) = \sum_{j=1}^{n} \boldsymbol{P}_{ij}\boldsymbol{p}_j(t), \qquad \boldsymbol{p}(t+1) = \boldsymbol{P}\boldsymbol{p}(t). \tag{2.7}$$

A *stationary distribution* of a Markov chain with transition matrix $\boldsymbol{P}$ is given by a probability distribution $\boldsymbol{p}^*$ over the nodes such that $\boldsymbol{p}^* = \boldsymbol{P}\boldsymbol{p}^*$, i.e., it is an eigenvector of the transition matrix with eigenvalue 1. The stationary distribution of the unbiased random walk exists and is unique by the Perron-Frobenius theorem (see, e.g., [64]) if the matrix $\boldsymbol{P}$ (or equivalently $\boldsymbol{A}$) is irreducible and hence if the network is strongly connected.

We say that a Markov chain is *ergodic* [101] if any initial distribution converges to the stationary distribution as $t \to \infty$. The unbiased random walk is ergodic provided the network is strongly connected and not bipartite. There is a direct connection between an arbitrary ergodic finite state and discrete time Markov chain and a random walk on a network (in general weighted and directed). If $\boldsymbol{P}$ is the transition matrix of an ergodic Markov chain with unique stationary distribution $\boldsymbol{p}^*$ (note that ergodicity also guarantees that $\forall i, \boldsymbol{p}_i^* > 0$), we can associate it with a network with adjacency matrix $\boldsymbol{A}$, where

$$\boldsymbol{A}_{ij} = \boldsymbol{P}_{ij}\boldsymbol{p}_j^*. \tag{2.8}$$

It then follows immediately from the column stochastic property (i.e., $\forall j, \sum_i \boldsymbol{P}_{ij} = 1$) of the transition matrix of a Markov chain, that the random walk on the network with the adjacency matrix $\boldsymbol{A}$ has transition matrix $\boldsymbol{P}$. Note that if the adjacency matrix $\boldsymbol{A}$ is symmetric, i.e., the network is undirected, the Markov chain defined by $\boldsymbol{P}$ is

reversible [101]. In particular, this connection means that we can apply any method to analyse directed networks to analyse the behaviour of arbitrary finite Markov chains.

To work around the problem of the lack of a unique stationary distribution for random walks on directed networks that are not strongly connected, one commonly introduces *teleportation* [72, 158]. That is, a random walker either follows an edge with probability $\alpha \in [0, 1]$ or teleports to a random node with probability $1 - \alpha$. The process is governed by the equation

$$\boldsymbol{p}(t+1) = \alpha \boldsymbol{P}\boldsymbol{p}(t) + (1-\alpha)\boldsymbol{s}, \tag{2.9}$$

where $\boldsymbol{s}$ is a probability distribution over the nodes, where $\boldsymbol{s}_i$ gives the likelihood of being teleported to node i. The stationary distribution $\mathrm{pr}(\alpha, \boldsymbol{s})$ of Eq. (2.9) is known as *PageRank* [158]. Equivalently, the PageRank $\mathrm{pr}(\alpha, \boldsymbol{s})$ is the solution to the linear system

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\,\mathrm{pr}(\alpha, \boldsymbol{s}) = (1-\alpha)\boldsymbol{s}, \tag{2.10}$$

which always has a unique solution provided $\alpha < 1$. The most common choice for $\boldsymbol{s}$ is the uniform distribution, but other choices can be used to indicate preferences for particular nodes (*personalised PageRank*) or to reduce teleportation bias [112]. In particular, Ref. [112] suggests the use of link teleportation, which corresponds to $\boldsymbol{s}_i = \frac{k_i^{\mathrm{in}}}{\sum_i k_i^{\mathrm{in}}}$, to reduce the teleportation bias.

One can extent the concept of a random walk to multilayer networks in different ways [48, 50, 149], all of which define Markov chains on the state nodes of the multilayer network. In particular, we can express a general random walk on a multilayer network in the form

$$\boldsymbol{p}_{i\boldsymbol{a}}(t+1) = \sum_{j\boldsymbol{b}} \boldsymbol{P}_{i\boldsymbol{a}}^{j\boldsymbol{b}} \boldsymbol{p}_{j\boldsymbol{b}}(t), \tag{2.11}$$

where $\boldsymbol{p}_{j\boldsymbol{b}}(t)$ is the probability for a random walker to be at node j in layer $\boldsymbol{b}$ at time step t and $\boldsymbol{P}_{i\boldsymbol{a}}^{j\boldsymbol{b}}$ is the probability for a random walker to transition from node j in layer $\boldsymbol{b}$ to node i in layer $\boldsymbol{a}$ in one time step. The most direct way to generalise a random walk to multilayer networks is the *classical random walk* [50, 149], which has

the transition tensor

$$\boldsymbol{P}_{i\boldsymbol{a}}^{j\boldsymbol{b}} = \frac{\boldsymbol{A}_{i\boldsymbol{a}}^{j\boldsymbol{b}}}{\sum_{i\boldsymbol{a}} \boldsymbol{A}_{i\boldsymbol{a}}^{j\boldsymbol{b}}}. \tag{2.12}$$

The classical random walk treats interlayer edges as equivalent to intralayer edges and is equivalent to a random walk on the flattened network. An alternative way to generalise the concept of a random walk is the *physical random walk* [48, 50] with transition tensor

$$\tilde{\boldsymbol{P}}_{i\boldsymbol{a}}^{j\boldsymbol{b}} = \frac{\boldsymbol{A}_{j\boldsymbol{a}}^{j\boldsymbol{b}}}{\sum_{\boldsymbol{a}} \boldsymbol{A}_{j\boldsymbol{a}}^{j\boldsymbol{b}}} \frac{\boldsymbol{A}_{i\boldsymbol{a}}^{j\boldsymbol{a}}}{\sum_{i} \boldsymbol{A}_{i\boldsymbol{a}}^{j\boldsymbol{a}}}. \tag{2.13}$$

One time step of the physical random walk corresponds to the random walker first switching layers with probabilities proportional to the weights of the interlayer edges and then performing an ordinary random walk step in the new layer. Note that the physical random walk is equivalent to the classical random walk on a multilayer network with adjacency tensor

$$\tilde{\boldsymbol{A}}_{i\boldsymbol{a}}^{j\boldsymbol{b}} = \boldsymbol{A}_{j\boldsymbol{a}}^{j\boldsymbol{b}} \boldsymbol{A}_{i\boldsymbol{a}}^{j\boldsymbol{a}}. \tag{2.14}$$

An interesting property of Eq. (2.14) is that the multilayer network defined by $\tilde{\boldsymbol{A}}$ is in general directed, even if the multilayer network defined by $\boldsymbol{A}$ is undirected. This means that the physical random walk on a multilayer network is in general not reversible, even if the multilayer network is undirected. We explore some of the implications of the different types of random walks on multilayer networks in Section 3.6.

We also consider continuous-time random walks on networks. In particular we consider a continuous-time random walk where the jumps are governed by independent Poisson processes at each node. The probability $\boldsymbol{p}_i(t)$ for a random walker to be at node i at time t is governed by the Kolmogorov equation

$$\frac{\mathrm{d}\boldsymbol{p}_i}{\mathrm{d}t}(t) = \sum_{j=1}^{n} \boldsymbol{P}_{ij}\tau_j \boldsymbol{p}_j(t) - \tau_i \boldsymbol{p}_i(t), \tag{2.15}$$

where τ_i is the expected *waiting time* of the Poisson process at a node i, i.e., the

expected time between the arrival of a random walker at node i and its next jump. Provided the network is undirected and connected, this process has the stationary distribution $\boldsymbol{p}_i^* \sim \frac{k_i}{\tau_i}$. Let $\boldsymbol{T}$ be the waiting time matrix with entries $\boldsymbol{T}_{ij} = \delta(i,j)\tau_i$, then we can write Eq. (2.15) in terms of the operator $\boldsymbol{R} = (\boldsymbol{P} - \boldsymbol{I})\boldsymbol{T}$, i.e.,

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t}(t) = \boldsymbol{R}\boldsymbol{p}. \tag{2.16}$$

One can define continuous-time random walks on multilayer networks in the same way, using either the classical or physical random walk for the walk steps.

2.4 Expansion Ratio and Expander Graphs

In this section, we provide a brief introduction to the concept of an *expander graph* (or, more simply, an *expander*) [181]. Essentially, expanders are graphs that are very well-connected. Because our empirical results in Chapter 3 indicate that many large social and information networks have expander-like properties—at least when viewed at large size-scales—it is useful to review basic properties about expander graphs. Most of the technical aspects of expander graphs are beyond the scope of this thesis, however [88] provides an excellent overview of this topic.

Let $G = (V, E)$ be a graph, which we assume for simplicity is undirected and unweighted. For the moment, we assume that all nodes have the same degree d (i.e., G is d-regular). For $S_1, S_2 \subset V$, the set of edges from S_1 to S_2 is then

$$E(S_1, S_2) = \{(i,j) : i \in S_1, j \in S_2, (i,j) \in E\}. \tag{2.17}$$

In this case, the number $|S|$ of nodes in S is a natural measure of the size of S. Additionally, the quantity $|E(S, \overline{S})|$, which indicates the number of edges that cross between S and $\overline{S}$, is a natural measure of the size of the boundary between S and $\overline{S}$.

We also define the *edge expansion of a set* of nodes $S \subset V$ as

$$h(S) = \frac{|E(S, \overline{S})|}{|S|}, \tag{2.18}$$

in which case the *edge expansion of a graph* G is the minimum edge expansion of any subset (of size no greater than $n/2$) of nodes:

$$h(G) = \min_{\{S \subset V : |S| \leq \frac{n}{2}\}} h(S)\,. \tag{2.19}$$

A sequence of d-regular graphs $\{G_i\}_{i \in \mathbb{N}}$ is a *family of expander graphs* if there exists an $\epsilon > 0$ such that $h(G_i) \geq \epsilon$ for all $i \in \mathbb{N}$. Informally, a given graph G is an expander if its edge expansion is large.

As reviewed in Ref. [88], one can view expanders from several complementary viewpoints. From a combinatorial perspective, expanders are graphs that are highly connected in the sense that one has to sever many edges to disconnect a large part of an expander graph. From a geometric perspective, this disconnection difficulty implies that every set of nodes has a very large boundary relative to its size. From a probabilistic perspective, expanders are graphs for which the natural random-walk process converges to its limiting distribution as rapidly as possible. Finally, from an algebraic perspective, expanders are graphs in which the first nontrivial eigenvalue of the Laplacian operator is bounded away from 0. (Because we are considering d-regular graphs here, note that this statement holds for both the combinatorial Laplacian and the normalised Laplacian.) In addition, constant-degree (i.e., d-regular, for some fixed value of d) expanders are the metric spaces that (in a very precise and strong sense [88]) embed least well in low-dimensional spaces (such as those discussed informally in Section 3.1). All of these interpretations imply that smaller values of expansion correspond more closely to the intuitive notion of better communities (whereas larger values of expansion correspond, by definition, to better expanders.)

The discussion of expander graphs in this section focuses on regular graphs, i.e., graphs where all nodes have the same degree. Real networks usually have highly heterogeneous degree distributions. For degree-heterogeneous graphs, edge expansion (Eqs. (2.18) and (2.19)) is not the most natural notion for expressing their isoperimetric properties. Given the deep connections between expansion and rapidly mixing random walks for regular graphs, a more natural notion for expressing isoperimetric

properties of degree-heterogeneous graphs is the conductance [93, 147]. We discuss conductance and its properties in Chapter 3. The difference between expansion and conductance simply amounts to a different notion of the size (or volume) of sets of nodes and the size of the boundary (or surface area) between a set of nodes and its complement. This difference is inconsequential for d-regular graphs. However, because of the deep connections between expansion and rapidly-mixing random walks, conductance is much more natural for graphs with substantial degree heterogeneity. The interpretation of failing to embed well in low-dimensional spaces (like lines or planes) is not as extremal in the case of conductance and degree-heterogeneous graphs as it is in the case of expansion and degree-homogeneous graphs. However, the interpretations of being well-connected, failing to provide bottlenecks to random walks, etc. all hold for conductance and degree-heterogeneous graphs such as those that we consider in Chapter 3. Accordingly, it is insightful to interpret our empirical results on small-scale versus large-scale structures in networks in light of known facts about expanders and expander-like graphs.

2.5 Community Detection

Communities are a particular type of meso-scale feature of a network and we discuss community structure and its relationship to other meso-scale features in Section 3.1. Intuitively, a community in a network is a set of nodes that are better connected to other nodes in the community than to nodes that do not belong to the community. This intuition for what a good community should be is rather vague and different methods to detect communities quantify this intuition in different ways. One can loosely distinguish between methods that define communities based on connection density and those that define communities based on the behaviour of dynamical process. From the first perspective, communities are groups of nodes that are more densely connected internally than they are to the rest of the network, whereas from the second perspective, communities are groups of nodes that trap a dynamical process (e.g., a random walk) for a long time. However, these two perspectives are often

intimately related. Examples include modularity, which is a popular quality function for quantifying community structure that was originally introduced using a connection density perspective [154], but that one can also think of as a linear approximation to a dynamic notion of community structure [110, 111, 149]. Another example is conductance, which has both a density based and dynamics based interpretation when used to quantify community structure. We discuss modularity in more detail in Section 2.5.1 and conductance in Chapter 3. One can further distinguish between local methods, which extract communities centred around a set of seed nodes and global methods, which find a community structure for the entire network.

Many different methods have been developed to algorithmically detect communities [61, 167] and these efforts have led to insights in wide variety of different applications including committee and voting networks in political science [138, 149, 166], friendship networks at universities and other schools [78, 199, 200], protein-protein interaction networks [130], granular materials [16], amorphous materials [175], brain and behavioural networks in neuroscience [18, 19, 209], collaboration patterns [59], human communication networks [60, 156], human mobility patterns [80], and many more. Here, we do not attempt to provide a comprehensive review of community detection methods. Rather, we survey key concepts and methods that are particularly relevant for the discussion in Chapters 3–5 and in particular more recent methods for identifying communities in multilayer networks.

In Chapter 3 we use local and "locally biased" methods to analyse community structure at different size scales. We discuss the three methods we use in Section 3.3. Examples of local methods are k-clique percolation [159], optimisation of local modularity [39], local spectral methods [4, 142], and random-walk-based methods [5]. Local methods are particularly suited to extracting small communities near nodes of interest in large networks—e.g., if the underlying network has to be sampled (such as by a web crawler). Furthermore, they naturally allow overlapping communities, as the communities constructed from different seed sets might share common nodes. However, to obtain a global view of community structure in a network using a local method, one would have to sample many different seed sets. An interesting procedure for making

this approach practical is to use an extremely cheap procedure to identify good seed nodes to use as an input for a local community detection algorithm [74, 208].

One of the oldest methods of community detection is hierarchical clustering, which was developed in the context of data clustering [41] and can be applied to networks by defining a distance measure between nodes [61, 69]. A standard way to implement hierarchical clustering is to start with the disconnected set of all nodes and iteratively link nodes in order of increasing distance. At each iteration, the components of the resulting graph provide a partition of the network into communities, which can be represented as a tree (also known as dendrogram) of clusters, with the single nodes at the bottom and the set of all nodes at the top. One obtains different hierarchical clustering algorithms depending on the update rule (the "linkage" function) for the distance between two clusters. The most efficient implementations [51, 188] scale as $\mathcal{O}(n^2)$ given the distance matrix, which itself can be costly to compute, and thus are not suitable for large networks. Furthermore, the communities found in this way are often not satisfactory [61, 69]. While hierarchical clustering is not the most effective algorithm for detecting communities, it illustrates a key link between the traditional data clustering problem [92] and community detection. In a similar way that one can apply algorithms for data clustering to identify communities in networks, one can also use community detection algorithms to identify clusters in data [47, 82]. We explore a similar approach using multilayer networks to represent the data in Chapter 5.

An alternative way to construct dendrograms are divisive methods, which iteratively remove links to obtain clusters. Girvan and Newman [69] proposed a divisive algorithm, which finds communities in a network by iteratively deleting edges with high betweenness, where the *betweenness* of an edge is given by the fraction of shortest paths between all pairs of nodes that contain the edge. At each step, the edge with highest betweenness is removed from the network and the betweenness values for the remaining edges are recalculated.

A different way to find communities is to optimise a quality function that identifies good partitions of the network. One of the most popular quality functions is

modularity, which was introduced by Newman and Girvan [154] to identify a good partition from the betweenness-based algorithm we discuss above. We discuss modularity and some of its variants in detail in Section 2.5.1. Modularity maximisation is NP-hard [26]. Hence finding an optimal partition is impossible for all but the smallest networks. Instead, the best one can achieve in practise are effective heuristics that find locally optimal partitions. A large number of heuristics have been developed that vary in speed and performance, including simulated annealing [84], extremal optimisation [55], spectral partitioning [152], and the Louvain locally greedy method [23]. We discuss a variant of the latter that allows one to easily experiment with different null models (see Section 2.5.1) in Appendix B.

A variety of other global community detection methods that do not rely on modularity maximisation have also been developed—including some that allow overlapping communities [3, 12, 168], identify statistically significant communities [119, OSLOM], or identify communities by minimising the description length of a random walk on a network [176, InfoMap].

2.5.1 Modularity

Having found a possible partition of a network into communities, one would often like to quantify the strength of the community structure—for example to compare different partitions of the same network or to distinguish networks with strong community structure from those with weak or non-existent communities. One can also directly optimise a quality function to obtain good partitions of a network.

A popular quality function for evaluating global community structure is "modularity" [154], which measures assortative mixing [151] with respect to communities. A key concept for modularity and other methods that deal with non-overlapping communities is the notion of a partition of a network. A *partition* $\mathcal{S} = \{S_1, \ldots, S_c\}$ of a network $G = (V, E)$ is a collection of c disjoint sets such that $V = \bigcup_{s=1}^{c} S_s$. We associate a partition $\mathcal{S}$ with its *partition vector* $\boldsymbol{s} \in \{1, \ldots, c\}^n$, where $\boldsymbol{s}_i = s$ if $i \in S_s$.

The original formulation [153, 154] of the *modularity* Q of a partition $\mathcal{S}$ of an

undirected network with adjacency matrix $\boldsymbol{A}$ into c disjoint communities is

$$Q_{\mathrm{NG}}(\mathcal{S};\boldsymbol{A}) = \frac{1}{2m}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\boldsymbol{A}_{ij} - \frac{k_i k_j}{2m}\right)\delta(\boldsymbol{s}_i,\boldsymbol{s}_j). \tag{2.20}$$

where $k_i = \sum_{j=1}^{n}\boldsymbol{A}_{ij}$ is the strength of node i, and $m = \frac{1}{2}\sum_{i,j=1}^{n}\boldsymbol{A}_{ij}$ is the total edge weight in the network. Thus, modularity compares the actual edge weight $\boldsymbol{A}_{ij}$ to that expected under a null model $\boldsymbol{N}_{ij} = \frac{k_i k_j}{2m}$, which preserves the expected strength of each node. We refer to this particular null model as the *Newman-Girvan null model*. By modifying the null model, one can take additional information about the network structure into account—including bipartite structure [15], signed edges [77], directed edges [102, 124], and spatial information [60, 183].

Resolution Parameters

Many real networks exhibit community structure at different scales, and one of the main issues with using modularity maximisation for community detection is that modularity in its original formulation [Eq. (2.20)] exhibits a resolution limit [62, 79], which imposes a lower limit on the size of communities in the optimal partition that depends on the size of the network. Additionally, modularity maximisation also has a tendency to split large communities [116]. Both of these problems are thought to also exist for other methods based on the global optimisation of a quality function [116] and in particular affect InfoMap [100, 184, 185].

In order to mitigate the issue of the resolution limits, resolution parameters have been introduced in the formulation of modularity [Eq. (2.20)], either by scaling the null model [169] or by introducing self-loops at each node [8]. The latter is sometimes known as "resistance", where the resistance r is the strength of the self-loop at each node. The resistance formulation of modularity is

$$Q_R(\mathcal{S};r,\boldsymbol{A}) = Q(\mathcal{S};\boldsymbol{A}+r\boldsymbol{I}), \tag{2.21}$$

where $\boldsymbol{I}$ is the identity matrix. Note that introducing resolution parameters does

not eliminate the resolution limit. For a fixed value of the resolution parameter the arguments for the resolution limits still apply and for most networks where the resolution limits are an issue, there will be no single value of the resolution parameter that allows one to detect both small- and large-scale structure simultaneously. However, changing the value of the resolution parameter allows one to shift the range of detectable community sizes and thus allows one to scan for structure at different size scales [8].

A particularly instructive way to derive an expression of modularity with a resolution parameter is as the linearised stability of random walk on a network [110]. The *stability* at time t of a partition $\mathcal{S}$ under a continuous time, ergodic random walk with operator $\boldsymbol{R}$ (see Eq. (2.15)) is defined as

$$R(\mathcal{S};\boldsymbol{R}) = \sum_{S\in\mathcal{S}} \mathbb{P}[S,t] - \mathbb{P}[S,\infty], \tag{2.22}$$

where $\mathbb{P}[S,t]$ is the probability that a random walker is in community S at time 0 and time t, and $\mathbb{P}[S,\infty]$ is the probability that a random walker is in community S at time 0 and in the limit as $t\to\infty$, which for an ergodic random walk is equivalent to the probability for two independent random walkers to be in S. At $t=0$ random walkers are distributed according to the stationary distribution of the random walk. A high value of the stability for a partition indicates that a random walker starting in a community of the partition at time 0 is more likely to be in the same communtiy at time t than in the limit as $t\to\infty$. Writing the probabilities $\mathbb{P}[S,t]$ and $\mathbb{P}[S,\infty]$ explicitly in terms of the operator $\boldsymbol{R}$ of the random walk, Eq. (2.22) becomes

$$R(\mathcal{S};t,\boldsymbol{R}) = \sum_{i,j=1}^{n} \left[\left(e^{t\boldsymbol{R}}\right)_{ij} \boldsymbol{p}_j^* - \boldsymbol{p}_i^*\boldsymbol{p}_j^*\right] \delta(\boldsymbol{s}_i,\boldsymbol{s}_j), \tag{2.23}$$

where $\boldsymbol{p}^*$ is the stationary distribution of the random walk.

One can interpret several different versions of modularity as linear approximations of stability based on different types of random walk. At linear order, the small-t

expansion of stability is

$$R(\mathcal{S};t,\boldsymbol{R}) = \sum_{i,j=1}^{n} \left[(\boldsymbol{I}+t\boldsymbol{R})_{ij}\boldsymbol{p}_j^* - \boldsymbol{p}_i^*\boldsymbol{p}_j^*\right]\delta(\boldsymbol{s}_i,\boldsymbol{s}_j). \tag{2.24}$$

Note that linearising stability is equivalent to considering only walks of length one.

In particular, if we consider an undirected network and choose uniform waiting times $\tau = 1$ in Eq. (2.15), such that $\boldsymbol{R} = \boldsymbol{P} - \boldsymbol{I}$, we recover *Newman-Girvan (NG) modularity* with a resolution parameter. Note that $\boldsymbol{R}$ is related by the similarity transformation $\boldsymbol{R} = -\boldsymbol{D}^{\frac{1}{2}}\mathcal{L}\boldsymbol{D}^{-\frac{1}{2}}$ to the normalised Laplacian. For a connected, undirected network this random walk has stationary distribution is $\boldsymbol{p}_i^* = \frac{k_i}{2m}$ and hence Eq. (2.24) simplifies to

$$R(\mathcal{S};t,\boldsymbol{R}) \approx (1-t) + \frac{1}{2m}\sum_{i,j=1}^{n}\left(t\boldsymbol{A}_{ij} - \frac{k_ik_j}{2m}\right)\delta(\boldsymbol{s}_i,\boldsymbol{s}_j), \tag{2.25}$$

which is equivalent by a linear transformation to the Reichardt and Bornholdt [169] formulation of modularity with a resolution parameter:

$$Q_{\mathrm{NG}}(\mathcal{S};\gamma,\boldsymbol{A}) = \frac{1}{2m}\sum_{i,j=1}^{n}\left(\boldsymbol{A}_{ij} - \gamma\frac{k_ik_j}{2m}\right)\delta(\boldsymbol{s}_i,\boldsymbol{s}_j), \tag{2.26}$$

where $\gamma = 1/t$. For directed networks, the same calculation—using the stationary distribution of the random walk where appropriate teleportation (see [112]) has to be introduced for networks that are not strongly connected—results in a version of modularity for directed networks [102] with a resolution parameter.

If one uses waiting times proportional to node strength, i.e., $\tau_i = k_i$, one can derive the *uniform* (Erdős-Rényi) version of modularity

$$Q_{\mathrm{U}}(\mathcal{S};\gamma,\boldsymbol{A}) = \frac{1}{2m}\sum_{i,j=1}^{n}\left(\boldsymbol{A}_{ij} - \gamma\frac{\langle k\rangle^2}{2m}\right)\delta(\boldsymbol{s}_i,\boldsymbol{s}_j) \tag{2.27}$$

in a similar way. Note that in this case $\boldsymbol{R} = \boldsymbol{P}\boldsymbol{D} - \boldsymbol{D} = -\boldsymbol{L}$ and hence the behaviour of this random walk is governed by the spectral properties of the combinatorial Lapla-

cian.

In some situations, one might have additional a priori information about the structure of a network, that one would like to take into account when evaluating community structure (e.g. bipartite or multilayer networks). In [149] Mucha et al. generalise the formulation of Eq. (2.22) to take additional structure into account. Instead of using $\mathbb{P}[S,\infty] = \sum_{i,j\in S} \boldsymbol{p}_i^* \boldsymbol{p}_j^*$, where $\boldsymbol{p}^*$ is the stationary distribution of the random walk (which gives the probability of two independent random walkers to be in community S) one introduces conditionally independent contributions, such that $\mathbb{P}[S,\infty] = \sum_{i,j\in S} \pi_{i|j} \boldsymbol{p}_j^*$, where $\pi_{i|j}$ is the conditional probability of jumping from node j to i in one step at stationarity given the a priori structure of a network. Note that this generalisation is consistent with linearised stability, which only accounts for one-step transitions, and does not make sense for stability itself. For a bipartite network, for example, one obtains $\pi_{i|j} = b_{ij}k_i/m$, where $b_{ij} = 0$ if node i and j belong to the same class and $b_{ij} = 1$ otherwise. Using the same type of calculation as before, one obtains a *bipartite* version of modularity by linearising stability:

$$Q_{\mathrm{B}}(\mathcal{S};\gamma,\boldsymbol{A}) = \frac{1}{2m}\sum_{i,j=1}^{n}\left(\boldsymbol{A}_{ij} - \gamma b_{ij}\frac{k_i k_j}{m}\right)\delta(\boldsymbol{s}_i,\boldsymbol{s}_j), \tag{2.28}$$

which is a generalisation of the Barber version of modularity for bipartite networks [15] that also includes a resolution parameter.

Multilayer Networks

Using the same technique, Mucha et al. [149] also derive a version of modularity for multilayer networks (see Section 2.1.1). The situation we consider here is one where only the intralayer edges are explicitly known, i.e., we have the intralayer adjacency tensor $\boldsymbol{A}_L$ but not the coupling adjacency tensor $\boldsymbol{A}_C$. We define the coupling tensor by introducing identity edges between state nodes that represent the same physical node in different layers in a way that reflects our a priori knowledge about the structure of the multilayer network. The two prototypical examples are *ordinal coupling* and *categorical coupling*. Ordinal coupling is appropriate if the layers of the multilayer

network have a natural order (e.g., different time snapshots of a temporal network). The coupling adjacency tensor for ordinal coupling only includes identity links for a state node in a layer to other state nodes that represent the same physical node in the previous and next layer, i.e.,

$$\boldsymbol{A}_{C\ ia}^{\mathrm{ord}\,jb} = w((jb, ia))\delta(i, j)\left(\delta(a, b-1) + \delta(a, b+1)\right). \tag{2.29}$$

Categorical coupling is appropriate if layers are unordered and structure in a particular layer can in principle depend on structure in any other layer (e.g., different types of connections in a multiplex network). The coupling adjacency tensor for ordinal coupling includes identity links between a state node and all other state nodes that represent the same physical node, i.e.,

$$\boldsymbol{A}_{C\ ia}^{\mathrm{cat}\,jb} = w((jb, ia))\delta(i, j)(1 - \delta(a, b)). \tag{2.30}$$

While this discussion only considers multilayer networks with a single aspect, it easily generalises to networks with more than one aspect (see Section 2.2), where we can have ordinal coupling for some aspect and categorical coupling for other aspects (e.g, a time-dependent multiplex network).

Given a choice for the coupling tensor, following [149], we derive a generalisation of NG-modularity for an undirected multilayer network by considering a continuous time random walk with classical walk steps (see Section 2.3), i.e.,

$$\frac{\mathrm{d}\boldsymbol{p}_{ia}}{\mathrm{d}t}(t) = \sum_{j=1}^{n}\sum_{b=1}^{l}\frac{\boldsymbol{A}_{ia}^{jb}}{k_{jb}}\tau_b\boldsymbol{p}_{jb}(t) - \tau_a\boldsymbol{p}_{ia}(t), \tag{2.31}$$

where we allow the expected waiting times τ_a for the Poisson processes at the state nodes to be different for state nodes in different layers but constrain them to be uniform within layers.

Conditional on the known structure of the multilayer network, the probability $\pi_{ia|jb}$ for a random walker to transition from state node jb to state node ia in one

step at stationarity is

$$\pi_{ia|jb} = \frac{k_{Lia}}{2m_a^a}\frac{k_{Ljb}}{k_{jb}}\delta(a,b) + \frac{\boldsymbol{A}_{C_{ia}}^{jb}}{k_{Cjb}}\frac{k_{Cjb}}{k_{jb}}\delta(i,j), \tag{2.32}$$

where $k_{Lia} = \sum_{j,b} \boldsymbol{A}_{L_{ia}}^{jb}$ is the intralayer strength, $k_{Cia} = \sum_{j,b} \boldsymbol{A}_{C_{ia}}^{jb}$ is the coupling strength, $k_{ia} = \sum_{j,b} \boldsymbol{A}_{ia}^{jb}$ is the strength of state node ia, and $m_a^b = \frac{1}{2}\sum_{i,j} \boldsymbol{A}_{ia}^{jb}$ is the total weight of edges linking state nodes in layer b to state nodes in layer a. The stationary distribution of Eq. (2.31) is

$$\boldsymbol{p}_{ia}^* = \frac{c}{\tau_a}\frac{k_{ia}}{2m}, \tag{2.33}$$

where $m = \frac{1}{2}\sum_{a,b} m_a^b$ is the total edge weight of the multilayer network and c is a normalisation constant that ensures that $\sum_{i,a} \boldsymbol{p}_{ia}^* = 1$. Substituting Eqs. (2.32) and (2.33) in the equation (Eq. (2.24)) yields (after applying a linear transformation)

$$Q(\mathcal{S}; t, \boldsymbol{A}) = \frac{1}{2m}\sum_{i,j,a,b}\Bigg[\left(\boldsymbol{A}_{L_{ia}}^{jb} - \frac{1}{\tau_b t}\frac{k_{Lia}k_{Ljb}}{2m_a^a}\right)\delta(a,b) + \left(\left(1 - \frac{1}{\tau_b t}\right)\boldsymbol{A}_{C_{ia}}^{jb}\right)\delta(i,j)\Bigg]\delta(\boldsymbol{s}_{ia}, \boldsymbol{s}_{ja}). \tag{2.34}$$

Note that the first term of the right hand side of Eq. (2.34) only depends on the intralayer edges, whereas the second term depends only on the coupling edges. Mucha et. al. [149] use this to motivate the following definition for *multilayer Newman-Girvan (MNG) modularity*,

$$Q_{\mathrm{MNG}}(\mathcal{S}; \boldsymbol{\gamma}, \omega, \boldsymbol{A}) = \frac{1}{2\mu}\sum_a\sum_{i,j}\left(\boldsymbol{A}_{L_{ia}}^{ja} - \boldsymbol{\gamma}_a\frac{k_{Lia}k_{Ljb}}{2m_a^a}\right)\delta(\boldsymbol{s}_{ia}, \boldsymbol{s}_{ja}) + \omega\frac{1}{2\mu}\sum_i\sum_{a,b}\boldsymbol{A}_{C_{ia}}^{ib}\delta(\boldsymbol{s}_{ia}, \boldsymbol{s}_{ib}), \tag{2.35}$$

which defines an equivalent optimisation problem to Eq. (2.34), with $\gamma_a = \frac{1}{\tau_a t}$, where one re-weights the coupling edges appropriately to introduce the ω parameter. Note that this re-weighting is not consistent with the dynamics based interpretation of

Eq. (2.34) as the re-weighting depends on t and the re-weighting diverges in the limit as $\gamma_a \to 1$. However, as an objective function for detecting communities Q_{MNG} has intuitive and desirable features which we motivate using a constraint optimisation perspective below. Here, $\mu = m_L + \omega m_C$, where $m_L = \frac{1}{2}\sum_{i,j,a,b} \boldsymbol{A}_{L}{}_{ia}^{jb}$ is the total intralayer edge weight and $m_C = \frac{1}{2}\sum_{i,j,a,b} \boldsymbol{A}_{C}{}_{ia}^{jb}$ is the total weight of the redefined coupling edges.

The first term of the right hand side of Eq. (2.35) is a weighted average of the NG-modularities for the individual layers, where the contribution of each layer is weighted by the total weight of the intralayer edges for the layer. The second term is a persistence contribution, where we define the (normalised) *persistence* of a multilayer partition as

$$\text{Per}(\mathcal{S}) = \frac{1}{2m_C} \sum_{i,ja,b} \boldsymbol{A}_{C}{}_{ia}^{jb} \delta(\boldsymbol{s}_{ia}, \boldsymbol{s}_{jb}), \tag{2.36}$$

generalising the persistence defined in [21] for ordered layers to general multilayer networks. If we assume that the coupling edges are identity links (i.e. linking state nodes that represent the same physical node), persistence is a measure of how uniform a partition is across layers.

Note that one can view Eq. (2.35) as the Lagrangian form of a constraint optimisation problem, where we want to maximise the intralayer modularity contributions subject to the constraint that the multilayer partition is sufficiently uniform across layers as measured by its persistence. If we ignore the $1/2\mu$ normalising factor which does not affect the optimal partitions for a given value of ω, Eq. (2.35) is a linear function of ω for a given partition. Many intuitive properties of multilayer modularity are a direct consequence of this observation and Ref. [21] shows many of them explicitly in the case of ordinal coupling, but most of them still hold for the more general version of persistence in Eq. (2.36). We explore the trade-off between intralayer modularity and persistence on artificial multilayer benchmark networks in Chapter 4 and on multilayer networks constructed from survey data in Chapter 5.

2.5.2 Consensus Clustering

A problem of modularity, and similar quality functions, is its near-degeneracy [79] in the space of partitions. In most networks, there are many near-optimal local maxima that can be very different from the global maximum. A large part of the near-degeneracy is a result of the at least exponentially growing number of ways that one can combine small modules into larger, near-optimal modules [79]. Although this poses severe problems for optimising modularity, it suggests a potential approach to mitigate the effects of the resolution limit other than introducing a resolution parameter. Small modules below the resolution limit can potentially be assigned to different, larger modules in different, near-optimal partitions. Thus, by extracting common features of many good partitions, one should be able to obtain a more accurate, nuanced picture of community structure in a network than by simply keeping the partition with the highest modularity that one was able to find. This approach has been explored in [179] to explore hierarchical structure of communities and in [117] to obtain communities that are both more consistent with planted ground-truth communities and less variable under recalculation than considering only the best partition.

To combine the results from multiple runs of a stochastic heuristic, [179] proposes an affinity measure based on the basin of attraction of local optima. In an abstract sense, the sampling procedure under consideration is a modularity-increasing stochastic process on the space of partitions. A modularity-increasing stochastic process $\mathcal{S}(t)$ satisfies

$$\mathbb{P}\left[\mathcal{S}(t+1) = \mathcal{S} | \mathcal{S}(t) = \mathcal{S}'\right] > 0, \ \mathcal{S} \neq \mathcal{S}' \quad \Rightarrow \quad Q(\mathcal{S}) > Q(\mathcal{S}'), \tag{2.37}$$

where the set of local maxima is $\mathcal{S}_{\max} = \{\mathcal{S} : \mathbb{P}[\mathcal{S}(t+1) = \mathcal{S} | \mathcal{S}(t) = \mathcal{S}] = 1\}$. Note that since the space of all partitions of a fixed number of nodes n is finite, any modularity-increasing process converges to a local maximum after a finite number of steps for any initial partition. The size of the basin of attraction of a local maximum

$\mathcal{S} \in \mathcal{S}_{\max}$ under a set of initial conditions $\mathcal{S}_0$ is

$$b(\mathcal{S}) = \frac{1}{|\mathcal{S}_0|} \sum_{\tilde{\mathcal{S}} \in \mathcal{S}_0} \mathbb{P}[\mathcal{S}(\infty) = \mathcal{S} | \mathcal{S}(0) = \tilde{\mathcal{S}}]. \tag{2.38}$$

The *association matrix* $\tilde{\boldsymbol{A}}$ of the sampling procedure (i.e., a set of initial partitions and a modularity-increasing stochastic process) is

$$\tilde{\boldsymbol{A}}_{ij} = \sum_{\mathcal{S} \in \mathcal{S}_{\max}} \delta(\boldsymbol{s}_i, \boldsymbol{s}_j) b(\mathcal{S}), \tag{2.39}$$

where $\boldsymbol{s}$ is the group vector of $\mathcal{S}$. In practise, one estimates the association matrix by averaging the co-classification index over the partitions obtained from many runs of a stochastic heuristic algorithm. Different heuristics for optimising modularity correspond to different choices for the modularity-increasing stochastic process and hence have different definitions of local maxima. Even if a partition is a local maximum for two different heuristics, its basin of attraction can be very different in the two cases.

To obtain a sample of the modularity landscape, Sales-Pardo et al. [179] use simulated annealing [84] with starting temperature $T = 0$. For this sampling procedure, the local maxima are partitions whose modularity score cannot be improved by changing the community assignment of a single node, by merging two communities, or by splitting a community in two. Although this sampling produces an accurate picture of the modularity landscape, it tends to be computationally expensive. We use GenLouvainRand (see Appendix B) in our numerical experiments, which selects moves with a bias towards larger increases in modularity and does not consider splitting communities.

Having obtained a suitably accurate sample of the modularity landscape, one still needs a method to extract information about community structure contained within an association matrix. Sales-Pardo et al. [179] fitted a hierarchical block model to the association matrix in order to extract community structure, which is computationally expensive even for moderately large networks with 1000's of nodes. Lancichinetti and Fortunato [117] suggest reapplying the same method used to obtain the association

matrix on the association matrix itself to obtain a consensus partition. The method we adopt here is a variation of the latter. Using modularity maximisation on the association matrix with the Newman-Girvan null model (Eq. (2.20)) is problematic because of the resolution limit. In particular, the strength of a node in the network defined by the association matrix is strongly related to the size of the community it belongs to. In the extreme case where all the sampled partitions are identical, a node's strength is exactly equal to the size of its community and this is still approximately true when partitions are noisy. As a result, the Newman-Girvan modularity has a tendency to split large communities and merge small communities.

This problem can be avoided by using a more appropriate null model in the formulation of modularity. Instead of fixing the expected strength of a node, we keep the sizes and numbers of communities of the original sample fixed while assigning nodes uniformly at random to each community. Under these assumptions, one can obtain the expected values of the entries of the null association matrix analytically. Let $\mathcal{C}$ be the set of all communities in a sample of q partitions. The expected values of the null association matrix are then given by

$$\left\langle \tilde{\boldsymbol{A}}_{ij} \right\rangle = \frac{1}{q} \sum_{C \in \mathcal{C}} \mathbb{P}[i \in C, j \in C]. \tag{2.40}$$

For fixed community sizes, the joint probabilities $\mathbb{P}[i \in C, j \in C]$ in Eq. (2.40) follow a hypergeometric distribution. Hence,

$$\mathbb{P}[i \in C, j \in C] = \begin{cases} \frac{|C|(|C|-1)}{n(n-1)}, & i \neq j, \\ \frac{|C|}{n}, & i = j. \end{cases}$$

With this choice of null model, we can identify communities independently of their size, depending only on the average co-classification of nodes in the community [198].

The concept of co-classification based consensus clustering extends to multilayer partitions in a straight-forward manner. The definitions of a modularity-increasing stochastic process (see Eq. (2.37)) and the basin of attraction (Eq. (2.38)) remain

unchanged, only the partitions we consider are now partitions of the state nodes of the multilayer network. The association matrix is replaced by an *association tensor*,

$$\tilde{\boldsymbol{A}}_{i\boldsymbol{a}}^{j\boldsymbol{b}} = \sum_{\mathcal{S}\in\mathcal{S}_{\max}} \delta(\boldsymbol{s}_{i\boldsymbol{a}}, \boldsymbol{s}_{j\boldsymbol{b}}) b(\mathcal{S}), \tag{2.41}$$

where $\boldsymbol{s}$ is the group tensor for the multilayer partition $\mathcal{S}$. We define a null model for the association tensor analogously to Eq. (2.40), i.e,

$$\left\langle \tilde{\boldsymbol{A}}_{i\boldsymbol{a}}^{j\boldsymbol{b}} \right\rangle = \frac{1}{q} \sum_{C\in\mathcal{C}} \mathbb{P}(i\boldsymbol{a} \in C, j\boldsymbol{b} \in C), \tag{2.42}$$

where $\mathcal{C}$ is the set of all individual multilayer communities in a sample of q multilayer partitions. However, when calculating the joint probabilities for community assignments in Eq. (2.42), we preserve the layer structure of the multilayer network. We assume that state nodes are assigned to communities at random in such a way that the size of each community within each layer is preserved. Let $C|_{\boldsymbol{a}} = \{i\boldsymbol{a} : i\boldsymbol{a} \in C\}$ be the restriction of community C to layer $\boldsymbol{a}$, then

$$\mathbb{P}(i\boldsymbol{a} \in C, j\boldsymbol{b} \in C) = \begin{cases} \frac{|C|_{\boldsymbol{a}}||C|_{\boldsymbol{b}}|}{n^2}, & i \neq j, \boldsymbol{a} \neq \boldsymbol{b}, \\ \frac{|C|_{\boldsymbol{a}}(|C|_{\boldsymbol{a}}|-1)}{n(n-1)}, & i \neq j, \boldsymbol{a} = \boldsymbol{b}, \\ \frac{|C|_{\boldsymbol{a}}|}{n}, & i = j, \boldsymbol{a} = \boldsymbol{b}. \end{cases} \tag{2.43}$$

Based on this null model we can identify multilayer consensus partitions by maximising

$$Q_{\mathrm{MC}}(\mathcal{S}; \tilde{\boldsymbol{A}}) = \sum_{i\boldsymbol{a}, j\boldsymbol{b}} \left(\tilde{\boldsymbol{A}}_{i\boldsymbol{a}}^{j\boldsymbol{b}} - \left\langle \tilde{\boldsymbol{A}}_{i\boldsymbol{a}}^{j\boldsymbol{b}} \right\rangle \right) \delta(\boldsymbol{s}_{i\boldsymbol{a}}, \boldsymbol{s}_{j\boldsymbol{b}}). \tag{2.44}$$

We use this form of modularity to identify consensus communities in Section 5.3.3.

Consensus clustering using association matrices is limited by the $O(n^2)$ memory and computational complexity of computing association matrices. Both can potentially be significantly reduced in the case of networks with clear community structure, as from the perspective of co-classification any two nodes that have identical commu-

nity assignments in all partitions are indistinguishable. Other methods of consensus clustering that avoid the need to calculate an association matrix entirely do exist [196] and could be useful for large networks.

2.5.3 Measures of Similarity of Partitions

The need to compare different partitions of the same network arises often and in many different contexts. For example, one might want to compare algorithmically-obtained community structure to some knowledge of ground-truth groupings, compare the output of different algorithms, or compare different runs of the same stochastic algorithm. Many different measures have been developed, which can be classified [145, 199] into information-theoretic, pair-counting, and cluster-matching methods. Key quantities for the definition of information-theoretic measures [42] include the *entropy*

$$H(X) = -\sum_{x \in X} p(x) \log(p(x)) \tag{2.45}$$

of a discrete random variable X, the *joint entropy*

$$H(X,Y) = -\sum_{x \in X, y \in Y} p(x,y) \log(p(x,y)), \tag{2.46}$$

of two discrete random variables X and Y, and the *conditional entropy*

$$H(X|Y) = -\sum_{x \in X, y \in Y} p(x,y) \log\left(\frac{p(x,y)}{p(y)}\right). \tag{2.47}$$

of X given Y.

The *mutual information* of two random variables X and Y measures the mean decrease in entropy of X given by knowledge of Y. That is,

$$I(X,Y) = H(X) - H(X|Y) = \sum_{x \in X, y \in Y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right). \tag{2.48}$$

For two partitions $\mathcal{S}$ and $\mathcal{T}$ of n nodes with partition vectors $\boldsymbol{s}$ and $\boldsymbol{t}$, one

estimates the joint and marginal probabilities (where we treat group assignments as random variables) in Eqs. (2.45)–(2.48) using the formulas

$$p(x,y) = \frac{\sum_{i=1}^{n} \delta(\boldsymbol{s}_i, x)\delta(\boldsymbol{t}_i, y)}{n}, \quad p(x) = \frac{\sum_{i=1}^{n} \delta(\boldsymbol{s}_i, x)}{n}, \quad p(y) = \frac{\sum_{i=1}^{n} \delta(\boldsymbol{t}_i, y)}{n}.$$

The conditional entropy can be used to identify hierarchical organisation, as $H(\mathcal{S}|\mathcal{T})$ vanishes if and only if partition $\mathcal{T}$ is nested within partition $\mathcal{S}$ [110]. As a general measure of similarity, the conditional entropy alone is unsuitable, as it is not symmetric and does not discriminate between different nested partitions that might be rather dissimilar. A quantity based on the conditional entropy that is often used to measure distance between partitions is the *variation of information* (VI) [145]

$$VI(\mathcal{S},\mathcal{T}) = H(\mathcal{T}|\mathcal{S}) + H(\mathcal{S}|\mathcal{T}) = H(\mathcal{S},\mathcal{T}) - I(\mathcal{S},\mathcal{T}), \tag{2.49}$$

which is the sum of the information needed to describe $\mathcal{T}$ given $\mathcal{S}$ and $\mathcal{S}$ given $\mathcal{T}$. The variation of information defines a genuine metric on the space of partitions, as it vanishes if and only if $\mathcal{S}$ and $\mathcal{T}$ are identical, is symmetric in $\mathcal{S}$ and $\mathcal{T}$, and satisfies the triangle inequality. A different metric on the space of partitions is the *normalised variation of information* (NVI) [105, 131]

$$NVI(\mathcal{S},\mathcal{T}) = \frac{VI(\mathcal{S},\mathcal{T})}{H(\mathcal{S},\mathcal{T})} = 1 - \frac{I(\mathcal{S},\mathcal{T})}{H(\mathcal{S},\mathcal{T})}. \tag{2.50}$$

The *normalised mutual information* (NMI),

$$NMI(\mathcal{S},\mathcal{T}) = \frac{I(\mathcal{S},\mathcal{T})}{H(\mathcal{S},\mathcal{T})}, \tag{2.51}$$

is also frequently used to compare partitions. Both the NVI and the NMI take values in [0,1] and are related by $NVI = 1 - NMI$. The choice of normalisation in Eq. (2.51) corresponds to an information theoretic analog of the Jaccard coefficient. One can normalise the mutual information in different ways, all of which are commonly referred to as "normalised mutual information". However when we use

NMI or normalised mutual information in this thesis, we always refer to the particular variant defined by Eq. (2.51). As a generic tool for quantifying the similarity of partitions, the information theoretic measures are well behaved, in particular when comparing relatively similar partitions. We use NMI as our main tool to compare partitions in Chapters 4 and 5.

However, in some situation other similarity measures can have key advantages. When comparing algorithmically obtained communities for real networks with external information about the nodes (which may or may not be related to community structure), one usually does not expect the partitions to be similar. Instead, one seeks to distinguish partitions that are completely unrelated from those which have some correlation. In this situation, one usually wants to compare the observed similarity between partitions too that expected under a null model for unrelated partitions. Estimating the behaviour of mutual information under a null-model can be cumbersome, though for sufficiently large sample sizes, the mutual information between two independent random variables approximately follows a gamma distribution [75]. Alternatively, one can use standardised pair-counting measures to identify correlated partitions [199] which are more tractable analytically. Pair-counting measures are based on counting the number of pairs of nodes that are co-classified in both partitions, one partition but not the other, or neither partition. Note the similarity between pair-counting and the co-classification-based association matrices we discuss in Section 2.5.2. In particular the null-model used in [199] to standardise different pair-counting measures is the same that we use in Section 2.5.2 to estimate the expected value of the association matrix.

Cluster-matching measures define the similarity between two partitions based on matching communities (i.e., clusters) in one partition with communities in the other partition. This reliance on matching communities can be desirable in some situations and makes the resulting values easy to interpret. However, these measures, by design, ignore any similarity between unmatched parts of communities, which can have counter-intuitive consequences [145]. Examples of cluster-matching measures are used in [54, 120, 146]. Here we are particularly interested in the "classification

error distance" [144–146] as it is related to the persistence contribution of multilayer modularity. Given two partitions $\mathcal{S}$ and $\mathcal{T}$ of n nodes with $|\mathcal{S}| \leq |\mathcal{T}|$ (i.e., partition $\mathcal{S}$ has at most as many communities as partition $\mathcal{T}$), we identify the communities of $\mathcal{S}$ with the communities of $\mathcal{T}$ by an injective map $\sigma : \mathcal{S} \to \mathcal{T}$. The *classification error* of a matching σ is defined as

$$CE(\sigma; \mathcal{S}, \mathcal{T}) = 1 - \frac{1}{n} \sum_{S \in \mathcal{S}} |S \cap \sigma(S)| \tag{2.52}$$

and the *classification error distance* between $\mathcal{S}$ and $\mathcal{T}$ is

$$d_{CE}(\mathcal{S}, \mathcal{T}) = \min_{\sigma} CE(\sigma; \mathcal{S}, \mathcal{T}) \tag{2.53}$$

Note the similarity between the sum in Eq. (2.36) which defines persistence and Eq. (2.52) which defines the classification error distance. This leads us to define the *persistence between a pair of partitions* [21] as

$$\mathrm{Per}(\mathcal{S}, \mathcal{T}) = \max_{\sigma} \frac{1}{n} \sum_{S \in \mathcal{S}} |S \cap \sigma(S)|. \tag{2.54}$$

We can think of partitions of the state nodes of a multilayer network as a collection of partitions of the physical nodes (one for each layer) with naturally defined mappings between them. More precisely, we define the *restriction of a multilayer community* S to a layer $\boldsymbol{a}$ as $S|_{\boldsymbol{a}} = \{i\boldsymbol{b} \in S : \boldsymbol{b} = \boldsymbol{a}\}$ and the *restriction of a multilayer partition* $\mathcal{S}$ to a layer $\boldsymbol{a}$ as $\mathcal{S}|_{\boldsymbol{a}} = \{S|_{\boldsymbol{a}}\}_{S \in \mathcal{S}}$. The multilayer partition $\mathcal{S}$ then induces a mapping $\sigma_{\boldsymbol{ab}} : \mathcal{S}|_{\boldsymbol{a}} \to \mathcal{S}|_{b}$ between the restricted partitions, where $\sigma_{\boldsymbol{ab}}(S|_{\boldsymbol{a}}) = S|_{\boldsymbol{b}}$. If we consider the special case of a multilayer network with only two layers, the mapping induced on the restricted partitions by a multilayer partition that optimises multilayer modularity is exactly the mapping that minimises the classification error between the restricted partitions. Formally, let $\boldsymbol{A}$ be the adjacency matrix of a multilayer network with two layers and $\mathcal{O} = \mathrm{argmax}_{\mathcal{S}}\, Q_{\mathrm{MNG}}(\mathcal{S}; \gamma, \omega, \boldsymbol{A}), \omega > 0$ be an optimal partition, then $\sigma_{12} = \mathrm{argmin}_{\sigma}\, CE(\sigma; \mathcal{O}|_1, \mathcal{O}|_2)$. For a general multilayer network, one can still

interpret the persistence contribution as a cluster-matching similarity measure in a similar way.

This means that multilayer modularity maximisation identifies communities by maximising an intralayer modularity contribution subject to a constraint that the restricted partitions of the multilayer partition are sufficiently similar based on a cluster-matching similarity measure. As a result, multilayer modularity maximisation inherits the key limitations of cluster-matching we just discussed, which potentially restricts the type of partitions we can identify. Conceivably, one could have defined multilayer modularity by using a different similarity measure in a similar way. However, using persistence (or another similar cluster-matching measure) has some key advantages. First, multilayer modularity as defined in Eq. (2.35) is consistent with other forms of modularity in that one can write it in terms of a modularity matrix, which means that one can use many existing algorithms for optimising modularity to optimise multilayer modularity (see Appendix B). Second, using a cluster-matching similarity measure means that we can identify multilayer partitions and not just a collection of correlated single-layer partitions.

2.5.4 Synthetic Benchmark Networks

As a result of the ill-defined nature of the community detection problem, synthetic benchmark networks with a known planted community structure are a crucial tool for validating and gaining a better understanding of the behaviour of community-detection algorithms. For such an approach to be optimally useful, it is desirable for the synthetic benchmarks to reproduce relevant features of real networks with community structure and it is challenging to develop good benchmarks that reproduce community structure and other structural properties of medium-sized and larger realistic networks.

For single-layer networks, many different types of benchmark networks have been developed to capture different aspects of community structure, starting with the simple benchmarks based on the "planted-l-partition model" [40] used in Ref. [69].

Perhaps the most popular family of benchmark networks are the so-called *LFR (Lancichinetti-Fortunato-Radicchi) networks* (later generalised in [114] to generate weighted networks, directed networks and networks with overlapping communities), which have power-law[2] degree and community size distributions to capture the heterogeneity inherent in most real networks.

To parametrise the family of LFR networks, one specifies its power-law degree distribution using its exponent τ_1, mean degree $\langle k \rangle$, and maximum degree $k_{\max}$. Similarly, one specifies its power-law community size distribution using its exponent τ_2, minimum community size $c_{\min}$, and maximum community size $c_{\max}$. Furthermore, one specifies the strength of community memberships using a mixing parameter μ, where each node shares a fraction $1 - \mu$ of its edges with nodes in its own community.

To construct a network with n nodes and these parameters, one samples n degrees from the degree distribution and samples community sizes from the community size distribution, subject to the constraint that the sum of community sizes should equal n. One then assigns nodes to communities uniformly at random, with the constraint that a node cannot be assigned to a community that is too small for the node to have the correct mixing-parameter value. Finally, one then constructs inter-community and intra-community edges separately by connecting the corresponding stubs (i.e., ends of edges) uniformly at random.

We analyse the community structure of LFR benchmark networks in Section 3.5 using local community detection methods which reveals some limitations of the LFR benchmark as a tool for generating realistic community structure. One should keep these limitations in mind when using the benchmark to compare the performance of community detection algorithms, as an algorithm that performs well on the benchmark may not handle the more complicated structure of real networks gracefully. We

[2]Note that a *power-law distribution* is a probability distribution with probability density function $p(x) = Cx^{\tau}$, where C is a normalisation constant which ensures that

$$\int_{x_{\min}}^{x_{\max}} p(x)\,\mathrm{d}x = 1,$$

τ is the *exponent* of the power law, and $x_{\min}$ and $x_{\max}$ are the *cutoffs* of the power law.

use the implementation by Lancichinetti[3] to generate LFR networks.

Other benchmarks have also been developed to capture particular aspects of some networks, e.g., spatially embedded networks [60, 183]. We are particularly interested in analysing community structure in multilayer networks and so far no benchmark networks for general multilayer networks exist. We introduce a framework for generating multilayer benchmark networks in Chapter 4 and use it to generalise LFR-like networks to a multilayer setting.

[3]available from `http://sites.google.com/site/andrealancichinetti/software`

Chapter 3

Meso-Scale Structure and Communities: Beyond a Block-Diagonal View of the World

This chapter presents the main results from Jeub et al. [94]. We start with the following question: What are possible ways that a network can "look like," very roughly if one "squints" at it? This question is admittedly vague, but the answer to it governs how small-scale network structure "interacts" with large-scale network structure, and it informs researchers' intuitions and the design decisions that they make when analysing networks (and when developing methods to analyse networks). As an example of this idea, it should be intuitively clear that if one "squints" at the nearest-neighbour $\mathbb{Z}^2$ network (i.e., the uniform lattice of pairs of integers on the Euclidean plane), then they "look like" the Euclidean plane $\mathbb{R}^2$. Distances are approximately preserved, and up to boundary conditions and discretisation effects, dynamical processes on one approximate the analogous dynamical processes on the other. In the fields of geometric group theory and coarse geometry, this intuitive connection between $\mathbb{Z}^2$ and $\mathbb{R}^2$ has been made precise using the notions of coarse embeddings and quasi-isometries [29].

3.1 Meso-Scale Structure and Simplified Block-Models

Establishing quasi-isometric relationships on networks that are expander graphs (see Section 2.4) is technically brittle [34]. Thus, for the present informal discussion, we rely on a simpler notion. Suppose that we are interested in the "best fit" of the adjacency matrix $\boldsymbol{A}$ to a 2×2 block matrix:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12}^T \\ \boldsymbol{A}_{12} & \boldsymbol{A}_{22} \end{pmatrix} ,$$

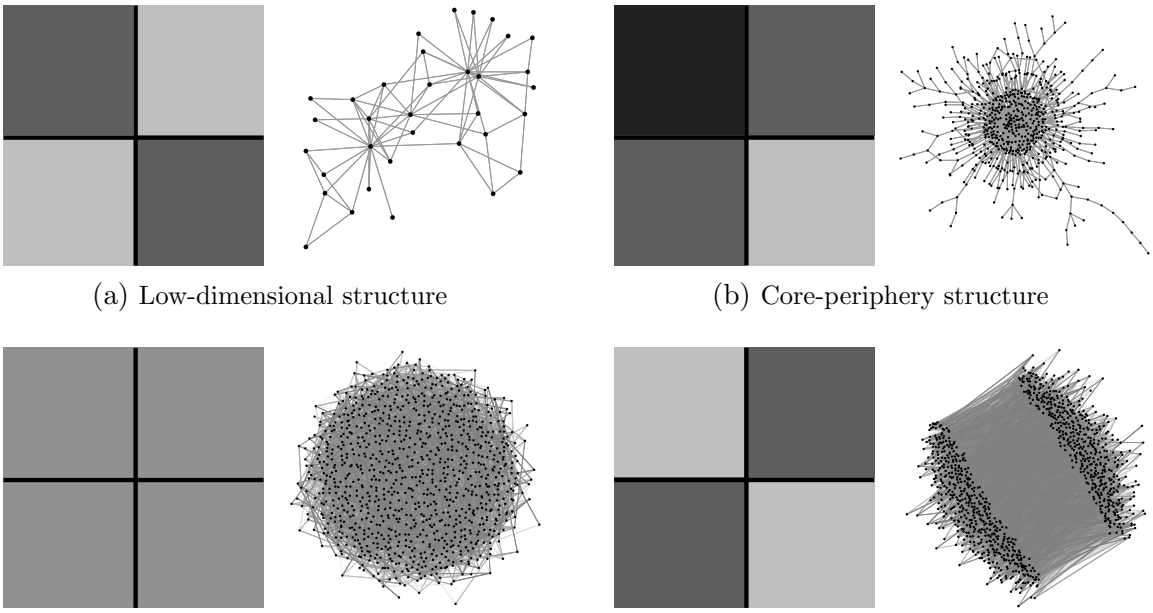
where $\boldsymbol{A}_{ij} = \alpha_{ij} \mathbf{1}\mathbf{1}^T$, where the "1-vector" $\mathbf{1}$ is a column vector of the appropriate dimension that contains a 1 in every entry and $\alpha_{ij} \in \mathbb{R}^+$. Thus, each block in $\boldsymbol{A}$ has uniform values for all its elements, and larger values of α_{ij} correspond to stronger interactions between nodes. The structure of $\boldsymbol{A}$ is then determined based on the relative sizes of α_{11}, α_{12}, and α_{22}. The various relative sizes of these three scalars have a strong bearing on the structure of the network associated with $\boldsymbol{A}$.

We illustrate several examples in Fig. 3.1. For the block models that we use for Panels (b)–(d) of Fig. 3.1, one block has n_1 nodes and the second block has n_2 nodes, and a node in block i is connected to a node in block j with probability α_{ij}[1].

- **Low-Dimensional Structure.** In Fig. 3.1a, we illustrate the case in which $\alpha_{11} \approx \alpha_{22} \gg \alpha_{12}$. In this case, each half of the network interacts with itself more densely than it interacts with the other half of the network. This "hot dog" or "pancake" structure corresponds to the situation in which there are two (or any

[1]The parameter choices for the block models have subtle effects that are not explained by the relative sizes of the probabilities α_{11}, α_{12}, and α_{22}. For a network with pronounced core-periphery structure (such as the one we show in Fig. 3.1b), where $\alpha_{11} \gg \alpha_{12} \gg \alpha_{22}$, one additionally needs $n_1\alpha_{11} \gg 1$ (for the core to be sufficiently connected), $n_1\alpha_{12} \ll 1$ (otherwise most peripheral nodes are directly connected to the core), and $n_2\alpha_{22} \approx 1$ (such that the periphery is reasonably connected but not too dense). Other parameter choices that respect the basic constraint on the probabilities still produce networks with well-defined density-based core-periphery structure [25, 44, 174], but they do not "look like" Fig. 3.1b. They tend to resemble Fig. 3.1c, provided they are reasonably well-connected).

For an expander-like network (such as the one we show in Fig. 3.1c), where $\alpha_{11} \approx \alpha_{12} \approx \alpha_{22} \approx \alpha$, one needs $(n_1 + n_2)\alpha \gg 1$ for the network to be sufficiently well-connected.



(a) Low-dimensional structure

(b) Core-periphery structure

(c) Expander or complete graph

(d) Bipartite structure

Figure 3.1: Idealised block models of network adjacency matrices; darker blocks correspond to denser connections among its component nodes. Panel (a) illustrates a low-dimensional "hot dog" or "pancake" structure; Panel (b) illustrates a "core-periphery" structure; Panel (c) illustrates an unstructured expander or complete graph; and Panel (d) illustrates a bipartite graph. Our example networks are the Zachary Karate Club [212] in Panel (a) and a realisation of a random-graph block model in Panels (b)–(d). For Panel (b) we only show the largest connected component (LCC), whereas the networks in Panels (c) and (d) are connected. The parameters for the block-model graphs are as follows: (b) $\alpha_{11} = 0.3$, $\alpha_{22} = 0.001$, $\alpha_{12} = 0.005$, $n_1 = 50$ nodes, and $n_2 = 950$ nodes (the LCC has 615 nodes); (c) $\alpha_{11} = \alpha_{22} = \alpha_{12} = 0.01$, and $n_1 + n_2 = 1000$ nodes; and (d) $\alpha_{11} = \alpha_{22} = 0$, $\alpha_{12} = 0.02$, and $n_1 = n_2 = 500$ nodes.

number, in more general cases) dense communities of nodes that are reasonably well-balanced in the sense that each community has roughly the same number of nodes. In this case, the network embeds relatively well in a one-dimensional, two-dimensional, or other low-dimensional space. Spectral clustering or other clustering methods often find meaningful communities in such networks, and one can often readily construct meaningful and interpretable visualisations of network structure.

- **Core-Periphery Structure.** In Fig. 3.1b, we illustrate the case in which $\alpha_{11} \gg \alpha_{12} \gg \alpha_{22}$. This is an example of a network with a density-based "core-periphery" structure [25, 44, 127, 128, 174]. In these cases, there is a core set

of nodes that are relatively well-connected amongst themselves as well as to a peripheral set of nodes that interact very little amongst themselves.

- **Expander or Complete Graph.** In Fig. 3.1c, we illustrate the case in which $\alpha_{11} \approx \alpha_{12} \approx \alpha_{22}$. This corresponds to a network with little or no discernible structure. For example, if $\alpha_{11} = \alpha_{12} = \alpha_{22} = 1$, then the graph is a clique (i.e., the complete graph). Alternatively, if the graph is a constant-degree expander, then $\alpha_{11} \approx \alpha_{12} \approx \alpha_{22} \ll 1$. As discussed in Section 2.4, constant-degree expanders are the metric spaces that embed least well in low-dimensional Euclidean spaces. In terms of the idealised block models in Fig. 3.1, they "look like" complete graphs, and partitioning them would not yield network structure that one should expect to construe as meaningful. Informally, they are largely unstructured when viewed at large size scales.

- **Bipartite Structure.** In Fig. 3.1d, we illustrate the case in which $\alpha_{12} \gg \alpha_{11} \approx \alpha_{22}$. This corresponds to a bipartite or nearly-bipartite graph. Such networks arise, e.g., when there are two different types of nodes, such that one type of node connects only to (or predominantly to) nodes of the other type [152].

Most methods for algorithmic detection of communities have been developed and validated using the intuition that networks have some sort of low-dimensional structure [128, 167, 211]. As an example, consider the infamous Zachary Karate Club network [212], which we show in Fig. 3.1a. This well-known benchmark graph, which seems to be an almost obligatory example to discuss in papers about community structure [173, 186], clearly "looks like" it has a nice low-dimensional structure. For example, there is a clearly identifiable left half and right half, and two-dimensional visualisations of the network (such as that in Fig. 3.1a) highlight that bipartition. Indeed, the Zachary Karate Club network possesses well-balanced and (quoting Herbert Simon [189]) "nearly decomposable" communities; and the nodes in each community are more densely connected to nodes in the same community than they are to nodes in the other community. Relatedly, appropriately reordering the nodes of the Zachary

Karate Club yields an adjacency-matrix representation with an almost block-diagonal structure with two blocks (as typified by the cartoon in Fig. 3.1a). Moreover, any reasonable community-detection algorithm should be able to find (exactly or approximately) the two communities.

Another well-known network that (slightly less obviously) "looks like" it has a low-dimensional structure is a so-called caveman network, which we illustrate later (in Fig. 3.2c). Arguably, a caveman network has many more communities than the Zachary Karate Club, so details such as whether an algorithm "should" split it into two or a somewhat larger number of reasonably well-balanced communities might be different than in the Zachary Karate Club network. However, a caveman network also has a natural well-balanced partition that respects intuitive community structure. Reasonable two-dimensional visualisations of this network (such as the one that we present in Fig. 3.2c) shed light on that structure and any reasonable community-detection algorithm can be adjusted to find (exactly or approximately) the expected communities. In Section 3.4, we will demonstrate that most realistic networks do *not* "look like" these small examples. Instead, realistic networks are often poorly-approximated by low-dimensional structures—e.g., with a small number of relatively well-balanced communities, each of which is more densely connected internally than it is with the rest of the network. Realistic networks often include substructures that more closely resemble core-periphery graphs or expander graphs (see Figs. 3.1bc) and networks that can be partitioned into nice nearly-decomposable communities tend to be the exception rather than typical [127, 128, 211].

In the remainder of this chapter, we investigate meso-scale structure in networks by analysing the local behaviour of random walks and other related dynamical processes. In particular, one can think of local community structure in terms of bottlenecks to dynamical processes. By investigating bottlenecks in a size-resolved manner, we can then identify size scales in a network where one can identify communities in the traditional sense that we discuss in Section 2.5. Furthermore, comparing bottlenecks at different size scales reveals information about how local communities in a network combine to form large scale features that often do not resemble global

community structure as exemplified by the block-diagonal model in Fig. 3.1a. We discuss the way we quantify bottlenecks in Section 3.2 and the methods we use to identify local communities in Section 3.3.

One can also investigate meso-scale structure in networks by directly fitting the parameters of a generative network model to an observed network. This approach also has the potential to identify structure that is not block-diagonal if one uses a generative model that is sufficiently general. Stochastic block models [86, 98, 162–164], which generalise the simplified block model we discuss in this section, are popular examples of generative models that have this feature. Stochastic block models rely on the assumption that the structure of the network of interest can be reasonably well approximated by using a relatively small number of discrete node types (the probability of observing an edge between a pair of nodes only depends on the nodes types and potentially a degree correction term). Another interesting model, which does not assume a block-diagonal structure, is the latent space model introduced in [155]. This model uses a continuous variable to represent node type, where the probability of observing an edge between a pair of nodes depends smoothly on the nodes types.

The approach we advocate in this chapter and the model fitting approach are complementary. Most generative models have inherent limitations in terms of the structure of networks they can produce. As we see in Section 3.5 on the example of networks generated from the LFR benchmark [114, 118] (which is a particular type of generative model with block-diagonal structure), computing a summary of network structure (such as the network community profile we discuss in the next section) for samples from a generative model can reveal how well they approximate qualitative features of real networks. This may help inform decisions about which model to use in a particular application.

3.2 Network Community Profiles (NCPs) and Their Interpretation

The main tool we use to quantify meso-scale structure in networks is the "network community profile" or "NCP". An NCP measures the quality of the best possible community of a given size as a function of the size of the purported community [127–129]. In this section, we provide a brief description of NCPs and how we use them.

3.2.1 The Basic NCP: Measuring Size-Resolved Community Quality

We start with the definition of conductance and the original conductance-based definition of an NCP from Ref. [128], and we then discuss our extensions of such ideas. For more details on conductance and NCPs, see Refs. [37, 38, 128, 142]. If $G = (V, E, w)$ is a graph with weighted adjacency matrix $\boldsymbol{A}$, then the *volume* between two sets S_1 and S_2 of nodes (i.e., $S_i \subset V$) equals the total weight of edges with one end in S_1 and one end in S_2. That is,

$$\mathrm{vol}(S_1, S_2) = \sum_{i \in S_1} \sum_{j \in S_2} \boldsymbol{A}_{ij} \,. \tag{3.1}$$

In this case, the *volume* of a set $S \subset V$ of nodes is

$$\mathrm{vol}(S) = \mathrm{vol}(S, V) = \sum_{i \in S} \sum_{j \in V} \boldsymbol{A}_{ij} \,. \tag{3.2}$$

In other words, the set volume equals the total weight of edges that are attached to nodes in the set. The volume $\mathrm{vol}(S, \overline{S})$ between a set S and its complement $\overline{S}$ has a natural interpretation as the "surface area" of the "boundary" between S and $\overline{S}$. In this study, a set S is a hypothesised community. Informally, the conductance of a set S of nodes is the "surface area" of that hypothesised community divided by "volume" (i.e., size) of that community. From this perspective, studying community structure amounts to an exploration of the isoperimetric structure of G.

Somewhat more formally, the *conductance* of a set of nodes $S \subset V$ is

$$\phi(S) = \frac{\mathrm{vol}(S, \overline{S})}{\min(\mathrm{vol}(S), \mathrm{vol}(\overline{S}))} \,. \tag{3.3}$$

Thus, smaller values of conductance correspond to better communities. The *conductance* of a graph G is the minimum conductance of any subset of nodes:

$$\phi(G) = \min_{S \subset V} \phi(S) \,. \tag{3.4}$$

Computing the conductance $\phi(G)$ of an arbitrary graph is an intractable problem (in the sense that the associated decision problem is NP-hard [9]), but $\phi(G)$ can be approximated by the second smallest eigenvalue λ_2 of the normalised Laplacian [37, 38].

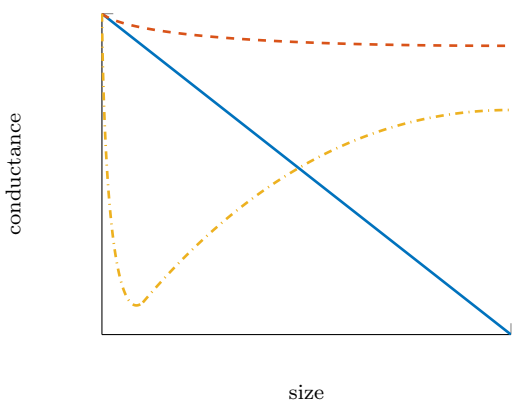
If the "surface area to volume" (i.e., isoperimetric) interpretation captures the notion of a good community as a set of nodes that is connected more densely internally than with the remainder of a network, then computing the solution to Eq. (3.4) leads to the "best" (in this sense) community of any size in the network.

Instead of defining a community quality score in terms of the best community of any size, it is useful to define a community quality score in terms of the best community of a given size k as a function of the size k. To do this, Ref. [128] introduced the idea of a *network community profile (NCP)* as the lower envelope of the conductance values of communities of a given size:

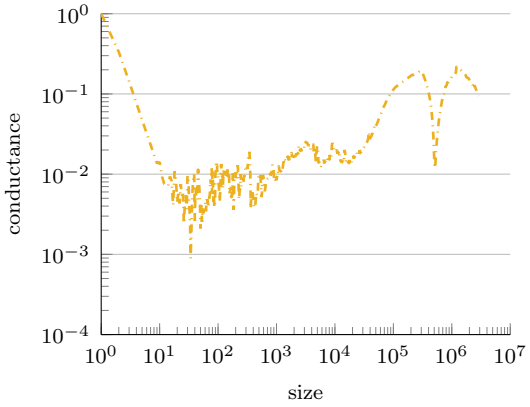
$$\phi_k(G) = \min_{S \subset V, |S|=k} \phi(S) \,. \tag{3.5}$$

An NCP plots a community quality score (which, as in Ref. [128], we take to be the set conductance of communities) of the best possible community of size k as a function of k. Clearly, it is also intractable to compute the quantity $\phi_k(G)$ in Eq. (3.5) exactly. Previous work has used spectral-based and flow-based approximation algorithms to approximate it [127–129].

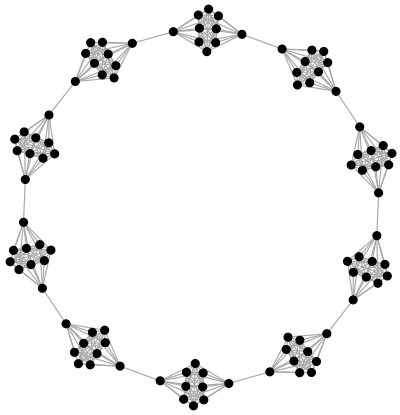
To gain insight into how to understand an NCP and what it reveals about network



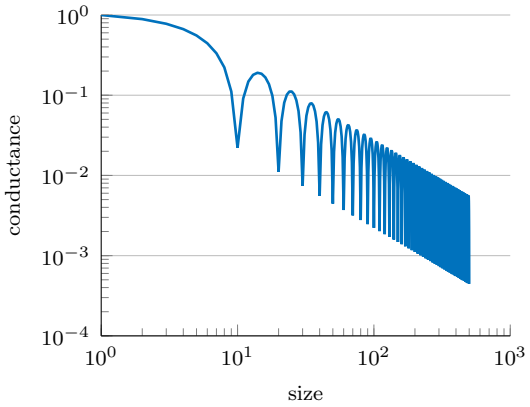
(a) Three possible NCPs



(b) Realistic NCP from [128]



(c) A caveman network



(d) NCP of caveman network

Figure 3.2: Illustration of network community profiles (NCPs). (a) Stylised versions of possible shapes for an NCP: downward-sloping (blue), upward-sloping (yellow), and flat (red). (b) NCP of a LiveJournal network that illustrates the characteristic upward-sloping NCP that is typical for many large empirical social and information networks [128]. (c) A toy "caveman network" with 10 cliques of 10 nodes each, where one edge from each clique has been rewired to create a ring [205]. (d) NCP for a similar caveman network with 100 cliques of 10 nodes each (the NCP for the network in Panel (c) is identical for communities with fewer than 50 nodes), illustrating the characteristic downward-sloping NCP that is typical of networks that are embedded in a low-dimensional space.

structure, consider Fig. 3.2. In Fig. 3.2a, we illustrate three possible ways that an NCP can behave. In each case, we are using conductance as a measure of community quality.

- **Upward-Sloping NCP.** In this case, small communities are "better" than large communities.

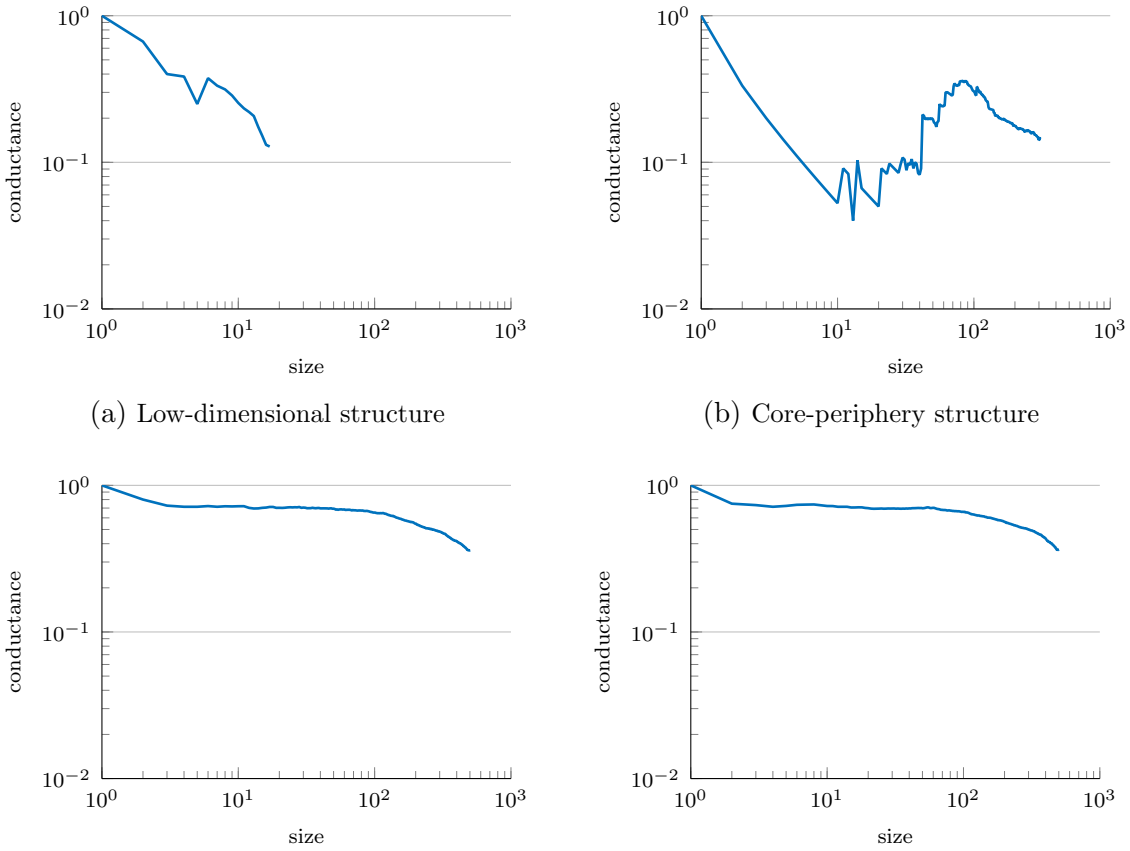
- **Flat NCP.** In this case, community quality is independent of size. (As illustrated in this figure, the quality tends to be comparably poor for all sizes.)

- **Downward-Sloping NCP.** In this case, large communities are "better" than small communities.

For ease of visualisation and computational considerations, we only show NCPs for communities up to half of the size of a network. A sampled NCP for very large communities, that we do not show in figures as a result of this choice, roughly[2] mirrors that for small communities, as the complement of a good small community is a good large community because of the inherent symmetry in conductance (see Eq. (3.3)).

In Fig. 3.2b, we show an NCP of a LiveJournal network from Ref. [128]. It demonstrates an empirical fact about a wide range of large social and information networks: there exist good small conductance-based communities, but there do not exist any good large conductance-based communities in many such networks. See Refs. [37, 38, 127–129, 142]) for more empirical evidence that large social and information networks tend not to have large communities with low conductances. In contrast, Fig. 3.2c illustrates a small toy network—a so-called "caveman network"—formed from several small cliques connected by rewiring one edge from each clique to create a ring [205]. As illustrated by the downward-sloping NCP in Fig. 3.2d, this network possesses good conductance-based communities, and large communities are better than small ones. One obtains a similar downward-sloping NCP for the Zachary Karate Club network [212] as well as for many other networks for which there exist meaningful visualisations [128]. The wide use of networks that have interpretable visualisations (such as the Zachary Karate Club and planted-partition models [40] with balanced communities) to help develop and evaluate methods for community detection and other procedures can lead to a strong selection bias when evaluating

[2]The exact NCP would necessarily be symmetric, but practical algorithms for approximately optimising conductance have implicit biases with respect to the internal structure of the communities they identify [128, 140]. Even just requiring that communities are connected subgraphs is enough for the NCP to not be exactly symmetric in general.



(a) Low-dimensional structure

(b) Core-periphery structure

(c) Expander or complete graph

(d) Bipartite structure

Figure 3.3: Network community profiles (NCPs) of the idealised example networks from Fig. 3.1. (a) NCP for the Zachary Karate Club network. (b) NCP for an example network generated from a block model with core-periphery structure. (c) NCP for an Erdős-Rényi graph. (d) NCP for an example network generated from a bipartite block model.

the quality of those methods.

We now consider the relationship between the phenomena illustrated in Fig. 3.2 and the idealised block models of Fig. 3.1. As a concrete example, Fig. 3.3 shows the NCPs for the example networks in the right panels of Fig. 3.2.

First, note that the best partitions consist roughly of well-balanced communities in the low-dimensional case of Figs. 3.1a and 3.3a, and the "lowest" point on an NCP tends to be for large community sizes. Thus, an NCP tends to be downward-sloping.

Networks with pronounced core-periphery structure—i.e., networks that "look like" the example network in Fig. 3.1b—tend to have many good small communities but no equally good or better large communities. This situation arises in many

large, extremely sparse networks [127–129]. The good small communities in such networks are sets of connected nodes in the extremely sparse periphery, and they do not combine to form good, large communities, as they are only connected via a set of core nodes with denser connections than the periphery. Thus, an NCP of a network with core-periphery structure tends to be upward-sloping (once one considers communities above some small size), as illustrated in Figs. 3.1b and 3.3b. However, this observation does not apply to all networks with well-defined density-based core-periphery structure. If the periphery is sufficiently well-connected (though still much sparser than the core), then one no longer observes good, small communities. Such networks act like expanders from the perspective of the behaviour of random walkers, so they have a roughly flat NCP. One can generate examples of such networks by modifying the parameters of the block model that we used to generate the example network in Fig. 3.1b (see Footnote 1 on page 44).

For a complete graph or a degree-homogeneous expander (see Figs. 3.1c and 3.3c), all communities tend to have poor quality, so an NCP is roughly flat. (See Section 2.4 for a discussion of expander graphs.)

Finally, bipartite structure itself does not have any characteristic influence on an NCP. Instead, an NCP of a bipartite network reveals other structure present in a network. For the example network in Fig. 3.1d, the two types of nodes are connected uniformly at random, so its NCP (see Fig. 3.3d) has the characteristic flat shape of an expander.

3.2.2 Robustness and Information Content of NCPs

It is important to discuss the robustness properties of NCPs. These are not obvious a priori, as the NCP is an extremal diagnostic. Importantly, however, the qualitative property of being downward-sloping, upward-sloping, or roughly flat is very robust to the removal of nodes and edges, variations in data generation and preprocessing decisions, and similar sources of perturbation [127–129]. For example, upward-sloping NCPs typically have many small communities of good quality, so losing some com-

munities via noise or some other perturbations has little effect on a realistic NCP. Naturally, whether a particular set of nodes achieves a local minimum is not robust to such modifications. In addition, one can easily construct pathological networks whose NCPs are not robust.

It is also important to consider the robustness of a network's NCP with respect to the use of conductance versus other measures of community quality. (Recall that many other measures have been proposed to capture the criteria that a good community should be densely-connected internally but sparsely connected to the rest of a network [128, 167].) Indeed, it has been shown that measures that capture both criteria of community quality (internal density and external sparsity) behave in a roughly similar manner to conductance-based NCPs, whereas measures that capture only one of the two criteria exhibit qualitatively different behaviour, typically for rather trivial reasons [129].

Although the basic NCP that we have been discussing yields numerous insights about both small-scale and large-scale network structure, it also has important limitations. For example, an NCP gives no information on the number or density of communities with different community quality scores. (This contributes to the robustness properties of NCP with respect to perturbations of a network.) Accordingly, the communities that are revealed by an NCP need not be representative of the majority of communities in a network. However, the extremal features that are revealed by an NCP have important system-level implications for the behaviour of dynamical processes on a network: they are responsible for the most severe bottlenecks for associated dynamical processes on networks [147].

Another property that is not revealed by an NCP is the internal structure of communities. Recall from Eq. (3.3) that the conductance of a community measures how well (relative to its size) that it is separated from the rest of a network, but it does not consider the internal structure of a community (except for size and edge density). In an extreme case, a community with good conductance might even consist of several disjoint pieces. Recent work has addressed how spectral-based approximations to optimising conductance also approximately optimise measures of internal

connectivity [213].

We augment the information from basic NCPs with some additional computations. To obtain an indication of a community's internal structure, we compute the internal conductance of the communities that form an NCP. The *internal conductance* $\phi_{\text{in}}(S)$ of a community S is

$$\phi_{\text{in}}(S) = \phi(G|_S)\,, \tag{3.6}$$

where $G|_S$ is the subgraph of G induced by nodes in the community S. The internal conductance is equal to the conductance of the best partition into two communities of the network $G|_S$ viewed as a graph in isolation. Because a good community should be well-separated from the remainder of a network and also relatively well-connected internally, we expect good communities to have low conductance but high internal conductance. We thus compute the *conductance ratio*

$$\Phi(S) = \frac{\phi(S)}{\phi_{\text{in}}(S)} \tag{3.7}$$

to quantify this intuition. A good community should have a small conductance ratio. We also plot so-called *conductance ratio profiles (CRPs)* [128] to illustrate how conductance ratio depends on community size in networks.

We employ three different methods, which we introduce in detail in Section 3.3, for sampling an NCP: one based on local diffusion dynamics (the AclCut method), one based on a local spectral optimisation (the MovCut method), and one based on geodesic distance from a seed node (the EgoNet method). In each case, we find communities of different sizes, and we then plot the conductance of the best community for each size as a function of size.

An NCP provides a signature of community structure in a network, and we can thereby compare community structure across different networks. Using different algorithms to identify potential communities helps one to discern which properties are attributable predominantly to network structure and which are attributable predominantly to choice of algorithms for community detection. Our approach of comparing

community structures in networks using NCPs and CRPs is very general: one can of course follow a similar procedure with other community-quality diagnostics on the vertical axis, other procedures for community generation, and so on.

3.3 Community Quality, Dynamics on Graphs, and Bottlenecks to Dynamics

The idea of using dynamics on a network has been exploited successfully by many methods for finding "traditional" communities (of densely connected nodes) [7, 52, 58, 109, 110, 149, 176] as well as for finding sets of nodes that are related to each other in other ways [6, 7, 22, 113, 125]. These methods exploit the relationship between community structure and dynamics at a global level. An important aspect of most dynamical processes is their dependence on initial conditions. Although we are concerned predominately with ergodic Markov chains, which eventually converge to a unique stationary distribution irrespective of the initial condition, the way in which and how quickly they converge is still depends strongly on initial conditions. We build on the idea that random walks and related diffusion-based dynamics, as well as other types of local dynamics (e.g., ones, like geodesic hops, that depend on ideas based on egocentric networks), should get "trapped" in good communities. As we see in Section 3.4, the location of such bottlenecks to dynamical processes can reveal aspects of meso-scale structure in networks that go beyond simple community structure.

In this section, we describe in more detail how we algorithmically identify possible communities that are centred on a seed node (or a seed set of nodes). We take an operational approach and view communities as the output of various dynamical processes (e.g., diffusions or geodesic hops), and we discuss the relationship between the output of those procedures to well-defined optimisation problems. In particular, we consider the following three dynamical methods for community identification.

3.3.1 Dynamics Type 1: Local Diffusions (the "AclCut" method)

In this procedure, we consider a random walk that starts at a given seed node s (sampled from a probability distribution given by the seed vector $\boldsymbol{s}$) and runs for some small number of steps. We take advantage of the idea that if a random walk starts inside a good community and takes only a small number of steps, then it should become trapped inside that community. To do this, we use the locally-biased personalised PageRank (PPR) procedure of Refs. [4, 5]. Recall from Eq. (2.9) that a PPR vector is implicitly defined as the solution of the equation

$$\mathrm{pr}(\alpha, \boldsymbol{s}) = \alpha \boldsymbol{A}\boldsymbol{D}^{-1}\mathrm{pr}(\alpha, \boldsymbol{s}) + (1-\alpha)\boldsymbol{s}\,, \tag{3.8}$$

where $1-\alpha$ is a "teleportation" probability and $\boldsymbol{s}$ is a seed vector. From the perspective of random walks, evolution occurs either by the walker moving to a neighbour of the current node or by the walker teleporting to a random node (e.g., determined uniformly at random as in the usual PageRank procedure or to a random node that is biased towards $\boldsymbol{s}$ in the PPR procedure). In general, teleportation results in a bias to the random walk, and one usually tries to minimise such a bias when detecting communities. (See Ref. [112] for clever ways to choose $\boldsymbol{s}$ with this goal in mind.)

The algorithm of Refs. [4, 5] deliberately exploits the bias from teleportation to achieve localised results. It computes an approximation to the solution of Eq. (3.8) (i.e., it computes an *approximate PPR vector*) by strategically "pushing" mass between the iteratively-updated approximate solution vector $\boldsymbol{p}$ and a residual vector $\boldsymbol{r}$ in such a way that most of the nodes in the original network are *not* reached. Consequently, this algorithm is typically *much* faster for moderately-large to very large graphs than is the naïve algorithm to compute a solution to Eq. (3.8) via, e.g., the power method. The seed vector for this method is the normalised indicator vector of a seed set $S_0 \subset V$ of nodes (i.e., $\boldsymbol{s}_i = 1/|S_0|$ if $i \in S_0$ and $\boldsymbol{s}_i = 0$ otherwise). For the calculations in Sections 3.4 and 3.5 we consider only seed sets that consist of a single

Algorithm 1 Compute an ϵ-approximate personalised PageRank vector with teleportation parameter α for a seed vector $\boldsymbol{s}$. Here, $\boldsymbol{P}$ is a Markov-chain transition matrix and $\boldsymbol{v}$ is a vector of node volumes (usually proportional to the stationary distribution of $\boldsymbol{P}$).

function AclCut($\boldsymbol{P}$, $\boldsymbol{v}$, $\boldsymbol{s}$, α, ϵ)
  $\tilde{\alpha} = (1-\alpha)/(1+\alpha)$ ▷ convert to equivalent lazy random walk teleportation
  $\boldsymbol{p} = \mathbf{o}$, $\boldsymbol{r} = \boldsymbol{s}$ ▷ initialise PageRank vector and residual
  $S = \{i : \boldsymbol{r}_i \geq \epsilon \boldsymbol{v}_i\}$
  while $S \neq \emptyset$ **do**
    choose $i \in S$ ▷ one can choose the next node for "push" arbitrarily
    $\tilde{r} = \boldsymbol{r}_i$
    $\boldsymbol{p}_i = \boldsymbol{p}_i + \tilde{\alpha}\tilde{r}$ ▷ push probability mass to PageRank vector
    $\boldsymbol{r}_i = (1-\tilde{\alpha})\tilde{r}/2$
    $\boldsymbol{r} = \boldsymbol{r} + (1-\tilde{\alpha})\boldsymbol{P}_{\cdot i}\tilde{r}/2$
    $S = \{i : \boldsymbol{r}_i \geq \epsilon \boldsymbol{v}_i\}$ ▷ check i and its neighbours to update S
  end while
  return $\boldsymbol{p}/\boldsymbol{v}$ ▷ return normalised PageRank vector for use with SweepCut
end function

seed node but in Section 3.6 we also consider other seed sets.

The algorithm is parametrised in terms of a "truncation" parameter ϵ where larger values of ϵ correspond to more locally-biased solutions. In Algorithm 1, we show the algorithm for a general Markov chain with transition matrix $\boldsymbol{P}$. We use this general formulation in Section 3.6. For undirected networks, which is the situation considered in [4, 5], we have $\boldsymbol{P} = \boldsymbol{A}\boldsymbol{D}^{-1}$ and $\boldsymbol{v} = \boldsymbol{k}$. At each iteration, the algorithm picks a node i, where $\boldsymbol{r}_i/k_i \geq \epsilon$ and performs a "push" operation which moves some probability mass from $\boldsymbol{r}_i$ to $\boldsymbol{p}_i$ and updates the residual vector in a way that preserves the invariant $\boldsymbol{p} = \mathrm{pr}(\alpha, \boldsymbol{s} - \boldsymbol{r})$. Initially all probability mass is concentrated at the residual of the seed nodes, i.e., $\boldsymbol{p} = \mathbf{o}$ and $\boldsymbol{r} = \boldsymbol{s}$. The algorithm terminates when $\boldsymbol{r}_i/k_i < \epsilon$ for all i. The running time of the algorithm is $\mathcal{O}(1/(\epsilon(1-\alpha))$. By the same argument, one can also bound the support of the approximate PageRank vector, where $\mathrm{vol}(\{i : \boldsymbol{p}_i > 0\}) \leq 3/\epsilon$. We refer to this procedure as the AclCut method.

3.3.2 Dynamics Type 2: Local Spectral Partitioning (the "MovCut" method)

In this procedure, we formalise the idea of a locally-biased version of the leading nontrivial eigenvector of the normalised Laplacian $\mathcal{L}$ that can be used in a locally-biased version of traditional spectral graph partitioning.

Following Ref. [142], consider the following optimisation problem:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimise}} \quad & \boldsymbol{x}^T \mathcal{L} \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{x}^T \boldsymbol{x} = 1\,, \\ & \boldsymbol{x}^T \boldsymbol{D}^{1/2} \mathbf{1} = 0 \\ & (\boldsymbol{x}^T \boldsymbol{D}^{1/2} \boldsymbol{s})^2 \geq \kappa\,, \end{aligned} \tag{3.9}$$

where κ is a locality parameter and $\boldsymbol{s}$ is a vector, which satisfies the constraints $\boldsymbol{s}^T D \boldsymbol{s} = 1$ and $\boldsymbol{s}^T D \mathbf{1} = 0$, and which represents a seed set of nodes. That is, in the norm defined by the diagonal D matrix, the seed vector $\boldsymbol{s}$ is unit length and is exactly orthogonal to the all-ones vector. This *locally-biased* version of standard spectral graph partitioning (which becomes the usual global spectral-partitioning problem if the locality constraint $(\boldsymbol{x}^T D^{1/2} \boldsymbol{s})^2 \geq \kappa$ is removed) was introduced in [142], where it was shown that the solution vector $\boldsymbol{x}^*$ inherits many of the nice properties of the solution to the usual global spectral-partitioning problem. The solution $\boldsymbol{x}^*$ is of the form

$$\boldsymbol{x}^* = c\,(\boldsymbol{L} - \gamma \boldsymbol{D})^+ \boldsymbol{D} \boldsymbol{s}\,, \tag{3.10}$$

where the parameter $\gamma \in (-\infty, \lambda_2(G))$ and $c \in [0, \infty)$ is a normalisation constant. Substituting $\gamma = (\alpha - 1)/\alpha$ and $c = (1 - \alpha)/\alpha$ in Eq. (3.10), one obtains

$$\boldsymbol{x}^* = \alpha \boldsymbol{A} \boldsymbol{D}^{-1} \boldsymbol{x}^* + (1 - \alpha) \boldsymbol{s}, \tag{3.11}$$

which relates the solution $\boldsymbol{x}^*$ of Eq. (3.9) and the PPR vector [142]. Consequently, it can be computed as the solution to a system of linear equations. Using iterative

solvers, one can find a sufficiently accurate solution for $\boldsymbol{x}^*$ in time approximately $\mathcal{O}(m)$ (see [142]). We use Bi-CGStab [202] to compute $\boldsymbol{x}^*$.

Note that the seed vector $\boldsymbol{s}$ in Eq. (3.11) is different from the one in Eq. (3.8), as it needs to satisfy the constraint $\boldsymbol{s}^T\boldsymbol{D}\mathbf{1} = 0$ and thus in general has negative entries. We thus call $\boldsymbol{x}^*$ a *generalised* PPR (GPPR) vector. To represent a seed set of nodes in this procedure, the indicator vector of the set is orthogonalised and normalised appropriately to satisfy the constraints.

If one performs a sweep cut (see Section 3.3.4) of this solution vector to obtain a locally-biased network partition, then one obtains Cheeger-like guarantees on approximation quality for the associated network community. Moreover, if the seed vector $\boldsymbol{s}$ corresponds to the indicator vector of a single node i, then this is a relaxation of the following *locally-biased graph partitioning problem*: Given as input a graph $G = (V, E, w)$, an input node i, and a positive integer k, find a set of nodes $S \subseteq V$ that achieves the minimum conductance among all sets of nodes that contain the input node i and have volume no greater than k [142]. We refer to this procedure (with a seed vector corresponding to a single seed node) as the MovCut method. Note that this method, while it finds local communities, is not a true local method as computing the solution to Eq. (3.10) uses global information about the network.

3.3.3 Dynamics Type 3: Local Geodesic Spreading (the "EgoNet" method)

In this procedure, we perform a geodesic-based (i.e., ego-network-based) dynamics that is analogous to the local random walks that we describe above. This method is similar to the technique for finding local communities that was introduced in [11] and that was generalised to weighted networks in [165]. Starting with a seed node s and a distance parameter k, this method considers all nodes j whose geodesic distance from s is at most k away—i.e., all nodes j such that $\Delta_{sj} \leq k$—to form a local community. Consequently, the local communities that we obtain using this method are simply the k-ego-nets of the seed node. For consistency with the other two methods, it is useful

to think of this method as inducing a ranking of the nodes,

$$\text{EgoRank}_i(s) = \frac{1}{1 + \Delta_{is}}\,, \tag{3.12}$$

where i is some node in a network. To compute the EgoRank vector, we need to find the geodesic distance from the seed node to all other nodes in the network which takes $\mathcal{O}(m + n \log n)$ using Dijkstra's algorithm [53].

Given the ranking interpretation in Eq. (3.12), we recover local geodesic-based communities from the EgoRank vector by using the sweep cut procedure that we describe in Section 3.3.4. The underlying dynamics for this method is analogous to the extreme case of a susceptible-infected (SI) spreading process [71, 99], in which an infected node infects all its neighbours with probability 1 at the time step following the one in which it is infected. One can then interpret the EgoRank of node i for a seed node s as the inverse of the time that it takes for node i to first become infected when only the seed node s is infected initially at $t = 1$. We refer to this procedure as the EgoNet method.

3.3.4 Sampling Procedures and Parameter Choices

To obtain an accurate picture of local community structure at different size scales throughout a network, we run each of the above community-identification procedures many times, starting at different seed nodes and running for different numbers of steps, and we then examine which nodes get visited as the dynamical processes unfold. For each seed node and value of the parameters, each of the AclCut, MovCut, and EgoNet methods returns a vector that can be used to "rank" the nodes of a network (in a locally-biased and size-resolved manner): AclCut and MovCut return a variant of the PPR vector, and EgoNet returns the EgoRank vector in Eq. (3.12). Given a ranking vector $\boldsymbol{p}$, the so-called "sweep sets" are given by $S_t = \{i \in V : \boldsymbol{p}_i \geq t\}$. Thus, there are at most $n + 1$ distinct sweep sets (where we recall that n is the number of nodes in the graph). A corresponding "sweep cut" is the network partition obtained from a sweep set that has minimal conductance among

Algorithm 2 Compute the conductance for all sweep sets of a ranking vector $\boldsymbol{p}$

function SweepCut($\boldsymbol{A}$, $\boldsymbol{p}$)
  $N = \text{RankOrder}(\boldsymbol{p})$ ▷ return nodes in descending order of $\boldsymbol{p}$
  $S = \emptyset$
  $c = 0$
  $v_{\text{in}} = 0$
  $v_{\text{out}} = \sum_{i,j=1}^{n} \boldsymbol{A}_{ij}$
  for $i \in N$ **do**
    $k = \sum_{j=1}^{n} \boldsymbol{A}_{ji}$
    $c = c - \sum_{j \in S} \boldsymbol{A}_{ij} + \sum_{j \notin S} \boldsymbol{A}_{ji}$
    $v_{\text{in}} = v_{\text{in}} + k$
    $v_{\text{out}} = v_{\text{out}} - k$
    $\boldsymbol{\phi}_i = \frac{c}{\min(v_{\text{in}}, v_{\text{out}})}$
    $S = S \cup i$
  end for
  return $\boldsymbol{\phi}$
end function

all all the $n + 1$ possible sweep sets.

By computing the conductance for each of the sweep sets, one obtains a locally-biased estimate for an NCP, centred around a seed node. One can then estimate a global NCP by taking the lower envelope over local NCPs for different seed nodes and parameter values[3]. One can compute the conductance for all sweep sets efficiently by incrementally updating the value of the conductance as one adds nodes to the sweep sets (see Algorithm 2). Let $S = \{i : \boldsymbol{p}_i \neq 0\}$ be the support of the ranking vector, then one can compute the conductance for all sweep sets in $\mathcal{O}(\text{vol}(S) + |S| \log |S|)$.

Recall that AclCut has two parameters (the teleportation parameter α and the truncation parameter ϵ), but that MovCut only has a single parameter (a teleportation parameter).

For AclCut, theoretical results [4] suggest that the method should find good communities of volume roughly ϵ^{-1}, where we have ignored constants and logarithmic

[3]Our Matlab code that implements these methods is available at `https://github.com/LJeub/LocalCommunities`.

factors. Furthermore, for a seed node i with strength k_i, AclCut returns empty communities for $\epsilon < k_i^{-1}$. This suggests that sampling using $\epsilon \in [k_{\max}^{-1}, \mathrm{vol}(G)^{-1}]$ gives good coverage of different size scales in practise. We use 20 logarithmically-spaced points in $[k_{\max}^{-1}, \mathrm{vol}(G)^{-1}]$ (including the endpoints) for computing NCPs using the AclCut method in this thesis. In addition, we use $\tilde{\alpha} = 0.001$, where $\tilde{\alpha}$ is the teleportation parameter of the "lazy random walk" defined in [4]. The (conventional) teleportation parameter that we use satisfies $\alpha = 1 - \frac{2\tilde{\alpha}}{1+\tilde{\alpha}}$, so that $\alpha \approx 0.998$ in Eq. (3.8). In our computations, we observe that increasing α leads to more accurate NCPs at the cost of longer computation times.

For MovCut, we use 20 equally-spaced values of α in the interval $[0.7, (1 - \lambda_2)^{-1} - 10^{-10}]$ (including the endpoints), where $(1-\lambda_2)^{-1}$ is the theoretical maximum for α (see [142]). The initial value of $\alpha = 0.7$ for the sampling interval is arbitrary to some extent. Note however that for small values of α the solution of Eq. (3.11) is dominated by the seed vector. For the networks we consider in this thesis, this choice of sampling interval yields a reasonable coverage of the interesting α-values. For other networks, one may want to verify that this holds by comparing the solution of Eq. (3.11) with the seed vector for a range of different values of α and different seed nodes.

To sample seed nodes, we modify the strategy described in Ref. [128] to be applicable to the MovCut method as well as the AclCut method. For each choice of parameter values, we sample nodes uniformly at random without replacement and stop the sampling process either when all nodes have been sampled or when the sampled local communities sufficiently cover the entire network. To determine sufficient coverage, we track how many times each node is included in the best local community that we obtain from the sweep sets and stop the procedure once each node has been included at least 10 times. This procedure ensures that good communities are sampled consistently. For the AclCut method, restricting the number of nodes sampled in this way allows one to sample an NCP in a time that scales almost linearly with the total number of edges in a network.

The EgoNet method does not have any size-scale parameters. For the network

sizes that we consider, it is feasible to use all nodes rather than sampling them, which is the procedure we use in Section 3.4.4.

3.4 Empirical Results on Real Networks

In this section, we present the results of our empirical evaluation of the small-scale, medium-scale, and large-scale community structure in several example networks.

3.4.1 Example Network Data Sets

We examine six empirical networks in depth. They fall into three classes: coauthorship networks, Facebook networks, and voting similarity networks. For each class, we consider two networks of two different sizes.

- **Collaboration Networks.** The two (unweighted) coauthorship networks were constructed from papers submitted to the `arXiv` preprint server in the areas of general relativity and quantum cosmology (CA-GrQc) and Astrophysics (CA-AstroPh). In each case, two authors are connected by an edge if they coauthored at least one paper, so a paper with k authors appears as a k-clique (i.e., a complete k-node subgraph) in the network. These network data are available as part of the Stanford Network Analysis Package (SNAP), and they were examined previously in Refs. [127–129].

- **Facebook Networks.** The two (unweighted) Facebook networks are anonymised data sets that consist of a snapshot of "friendship" ties on one particular day in September 2005 for two United States (U.S.) universities: Harvard (FB-Harvard1) and Johns Hopkins (FB-Johns55). They form a subset of the Facebook100 data set from Refs. [199, 200]. In addition to the friendship ties, we possess node labels for gender and class year as well as numerical identifiers for student or some other (e.g., faculty) status, major, and high school for some of the nodes.

- **Congressional Voting Networks.** The two (weighted) Congressional voting networks represent similarities in voting patterns among members of the U.S. House of Representatives (US-House) and U.S. Senate (US-Senate). Our construction follows prior work [148, 149]. In particular, we represent these two data sets as "multilayer" temporal networks [103, 149]. Each layer corresponds to a single two-year Congress, and edge weights within a layer represent the voting similarity between two legislators during the corresponding Congress. In layer s, this yields adjacency elements of $A_{ij}^{(s)} = \frac{1}{b_{ij}(s)} \sum_k \gamma_{ijk}$, where $\gamma_{ijk} = 1$ if both legislators voted the same way on the k^{th} bill, $\gamma_{ijk} = 0$ if they voted in different ways on that bill, $b_{ij}(s)$ is the number of bills on which both legislators voted during that Congress, and the sum is over bills. A tie between the same legislator in consecutive Congresses is represented by an interlayer edge with weight ω [149]. (We use $\omega = 1$; the effect of changing ω was investigated in [17, 148].) We represent each multilayer voting network using a single "supra-adjacency matrix" (see Refs. [49, 50, 76, 103]) in which the different Congresses correspond to diagonal blocks and interlayer edges correspond to off-block-diagonal terms in the matrix. Note that throughout this chapter we treat the Congressional voting graphs at the level of this supra-adjacency matrix, without any additional labelling or distinguished treatment of inter- and intra-layer edges (cf. [149]).

We chose these three sets of networks because as they have very different properties with respect to their large-scale versus small-scale community structures. We thus emphasise that, with respect to the topic of this chapter, these six networks are representative of several broad classes of previously-studied networks: CA-GrQc and CA-AstroPh are representative of the SNAP networks that were examined previously in Refs. [127–129]; both FB-Harvard1 and FB-Johns55 (aside from a few very small communities in FB-Harvard1) are representative of the Facebook100 networks that were examined previously in Refs. [199, 200]; and US-House and US-Senate give examples of networks (which are larger than the Zachary Karate

	Nodes	Edges	$\langle k \rangle$	λ_2	$\langle$C$\rangle$	Refs.	Description
CA-GrQc	4 158	13 422	6.5	0.0019	0.56	[127–129]	Coauthorship network: `arXiv` general relativity
CA-AstroPh	17 903	196 972	22.0	0.0063	0.63	[127–129]	Coauthorship network: `arXiv` astrophysics
FB-Johns55	5 157	186 572	72.4	0.1258	0.27	[199, 200]	Johns Hopkins Facebook network
FB-Harvard1	15 086	824 595	109.3	0.0094	0.21	[199, 200]	Harvard Facebook network
US-Senate	8 974	422 335	60.3	0.0013	0.50	[17, 148, 149]	Network of voting patterns in U.S. Senate
US-House	36 646	6 930 858	240.5	0.0002	0.58	[17, 148, 149]	Network of voting patterns in U.S. House

Table 3.1: Overview of the six example networks. For each network, we show the number of nodes and edges in the largest connected component (LCC), the mean degree/strength ($\langle k_i \rangle$), the second-smallest eigenvalue (λ_2) of the normalised Laplacian matrix, the mean clustering coefficient ($\langle C_i \rangle$), prior references that used these networks, and a brief description.

Club and caveman networks) on which conventional notions of and algorithms for community detection have been validated successfully [148, 149].

In Table 3.1, we provide summary statistics for each of the six networks. We give the numbers of nodes and edges in the largest connected component, the mean degree/strength ($\langle k_i \rangle$), the second-smallest eigenvalue (λ_2) of the normalised Laplacian matrix, and mean clustering coefficient ($\langle C_i \rangle$). We use the local clustering coefficient $C_i = \frac{1}{k_i(k_i-1)} \sum_{j,k} (\hat{w}_{ij}\hat{w}_{ik}\hat{w}_{jk})^{\frac{1}{3}}$, where $\hat{w}_{ij} = \frac{w_{ij}}{\max_{ij} w_{ij}}$, which reduces to the usual expression for local clustering coefficients in unweighted networks [157, 180, 206]. The high values for mean clustering coefficient in both the U.S. Congress and coauthorship networks are unsurprising, given how those networks have been constructed.

Recall that the second smallest eigenvalue λ_2 of the normalised Laplacian provides a qualitative notion of connectivity that can be used to bound the mixing time of diffusion-based dynamics on networks [93] (where larger values of λ_2 imply that

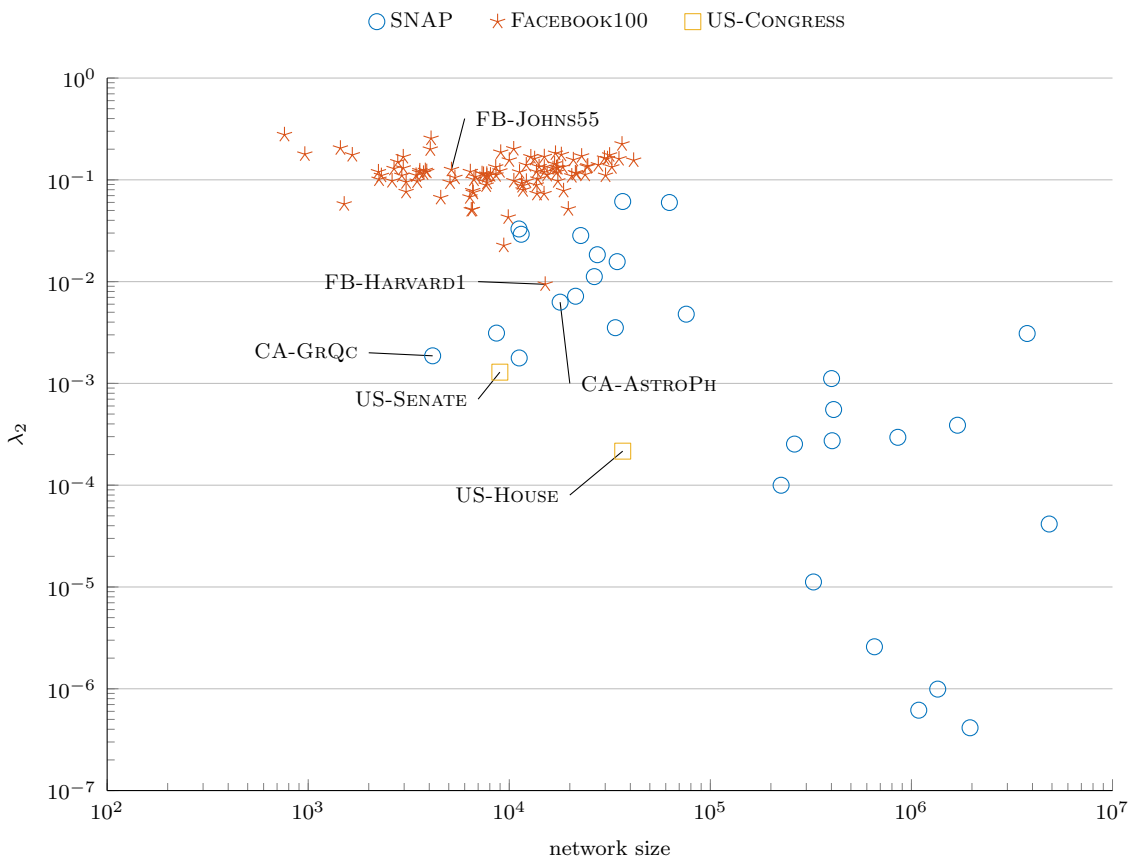


Figure 3.4: Scatter plot of the second smallest eigenvalue (λ_2) of the normalised Laplacian versus size of the network for the networks from the SNAP data that were studied in [127, 128], all 100 networks in the Facebook100 data set [199, 200], and the two US-Congress temporal networks [17, 148, 149].

there are fewer bottlenecks to mixing) and which can also be used to partition a graph into communities [38, 152, 153] (where smaller values of λ_2 correspond to the existence of better communities). We show the values of λ_2 for our six networks in Table 3.1. For comparison, we show in Fig. 3.4 a scatter plot of λ_2 versus the size of the network (i.e., the number of nodes in the network) for these six networks, the remaining networks from SNAP[4] that were also studied in [127, 128], and the remaining 98 networks from the Facebook100 data set [199, 200].

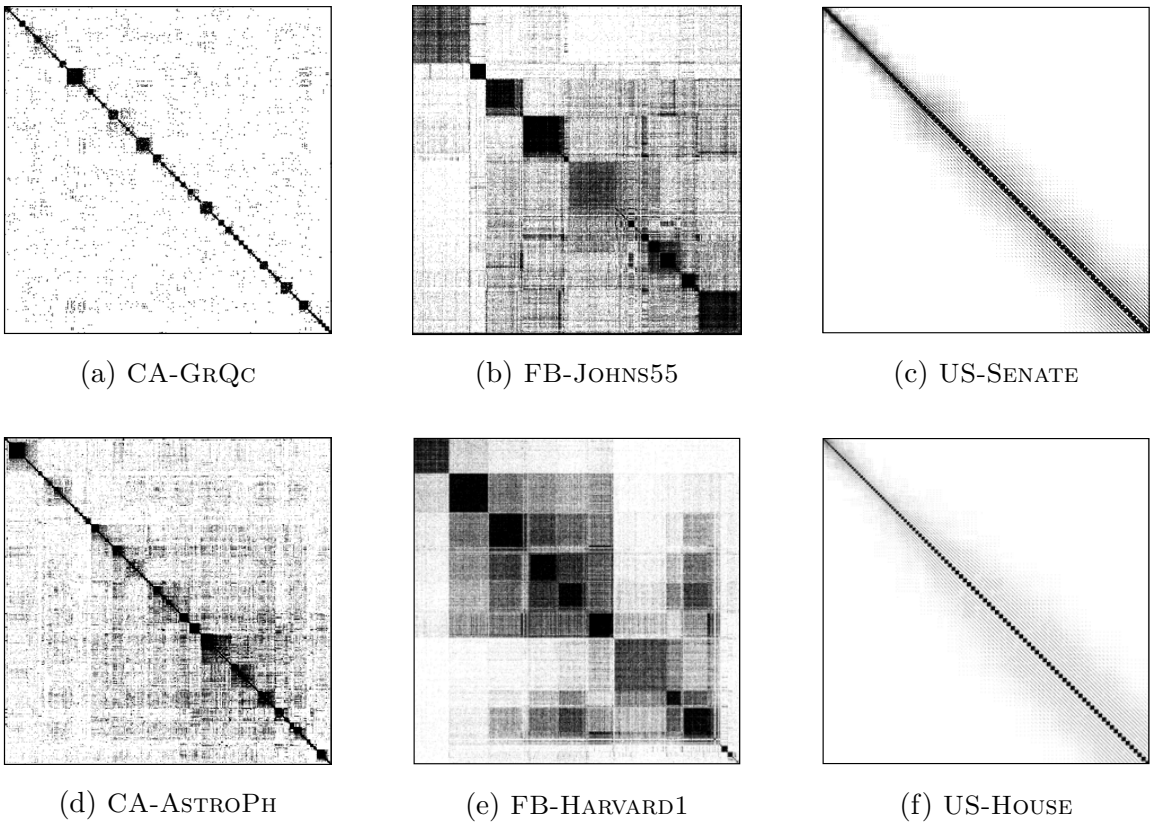
The first point to note about Fig. 3.4 is that λ_2 for nearly all of the Facebook100 graphs is much larger than those for the two collaboration graphs and the two voting graphs. Figure 3.4 and previous empirical results (from Refs. [127, 128]) clearly

[4]available from `http://snap.stanford.edu`

demonstrate that the λ_2 values for the two collaboration graphs are representative of (and, in many cases, higher than) those of the other SNAP graphs studied empirically in Refs. [127, 128]. That is, nearly all of the networks have λ_2 values that are much smaller than those in the Facebook100 graphs. This implies, in particular, that those graphs contain more substantial bottlenecks to mixing. (Note, though, that the value of λ_2 says nothing about the size or cardinality of the set of nodes that achieves the minimum.) In order to understand these differences, we study two networks from the Facebook100 data set in detail: one (FB-Johns55) with a typical value of λ_2 and another (FB-Harvard1) that is an "outlier," in that it has the lowest value of λ_2 in the entire Facebook100 data set. (The FB-Caltech36 network is the smallest network in the Facebook100 data set—it has 762 nodes in its largest connected component (LCC)—and it has the largest value of λ_2.)

The second point to note about Fig. 3.4 and Table 3.1 is that they suggest that FB-Johns55 (and possibly also FB-Harvard1) are better connected than the other four networks, and that the connectivity properties of the two collaboration graphs and the two voting graphs (and perhaps also FB-Harvard1) might be very similar. As we will see below, however, the situation is considerably more subtle.

In Fig. 3.5, we visualise the adjacency matrices of each of these networks using a sparsity-pattern (Spy) plot. We draw the nonzero entries of the adjacency matrix as black dots. The grayscale visualisation in Fig. 3.5 is a result of coarsening the dpi resolution and illustrates the density of connections in an area of the adjacency matrix. This yields a visualisation comparable to the idealised block models in Fig. 3.1. The node order in a Spy plot is arbitrary and, by permuting the nodes, one can sometimes obtain visualisations that are suggestive of structural features in a network. For the coauthorship and Facebook networks, we use results from a single run of an implementation (see Appendix B) of a Louvain-like heuristic [23] for modularity optimisation to partition these networks into communities. We then sort nodes by community assignment: we choose the order of communities manually to suggest potential large-scale structures. For the voting similarity networks, time provides a natural ordering for the nodes. We start with the nodes from the 1st Congress



(a) CA-GrQc (b) FB-Johns55 (c) US-Senate

(d) CA-AstroPh (e) FB-Harvard1 (f) US-House

Figure 3.5: Sparsity-pattern (Spy) plots for the largest connected component of each of our six example networks. The coauthorship networks (CA-GrQc and CA-AstroPh) and Facebook networks (FB-Johns55 and FB-Harvard1) are arranged by communities that we obtained using an implementation of a Louvain-like heuristic (see Appendix B) for modularity optimisation [23]. For US-Congress, we preserve the temporal order of the nodes starting with the first Congress in the top left and ending with the 110th Congress in the bottom right.

and end with the nodes from the 110th Congress. The small blocks on the diagonal are the individual Congresses, which are almost fully connected internally, and the off-diagonals result from the interlayer coupling between the same individuals from different Congresses.

While certainly not definitive, Fig. 3.5 suggests several hypotheses about the relationship between small-scale structure and the large-scale structure—and, in particular, between small communities and large communities—in these six networks. First, from Figs. 3.5cf, it appears that the large-scale structure in US-Senate and US-House corresponds to that of a "banded" matrix[5]. This banded structure is a

[5]Banded matrices arise, for example, in the study of finite-difference equations. In banded matri-

result of the interlayer edges in these networks. Second, from Figs. 3.5ad, it appears that CA-GrQc and CA-AstroPh both have many small-scale communities. It appears that they have a large-scale structure that is roughly banded, but there also appear to be many "long-range" off-diagonal interactions between distant nodes in the depicted ordering. Third, from Figs. 3.5be, we observe that both FB-Johns55 and FB-Harvard1 appear to have roughly 10 communities that are both relatively large and relatively good.

From these visuals, it appears that nearly all of these communities have dense internal connections and sparse connections to other communities. Given the usual notion that communities are sets of nodes with denser connections among its constituent nodes than with the rest of the network, the visualisations in Fig. 3.5 appear to suggest that there may be interesting large-scale structure that might be exploitable in FB-Johns55 and FB-Harvard1 but not in the other networks. In particular, FB-Johns55 and FB-Harvard1 seem to be examples of the case $\alpha_{11} \approx \alpha_{22} \gg \alpha_{12}$ illustrated in Fig. 3.1a.

The focus of the present chapter is to test the extent to which the above hypotheses about the relationship between small-scale structure and large-scale structure in these six networks is correct. As we have discussed, intuition like what we have illustrated in Fig. 3.5 is common in the development and validation of methods for community detection, so it is useful to delve into great depth on a set of networks to explore the connections between small-scale and large-scale connections in networks. As we will see in the next several sections, the situation is considerably more subtle than these figures (and commonly-employed intuition) might suggest. For example, with the exception of the small communities in CA-GrQc/CA-AstroPh and the large-scale structure (i.e., the one-dimensional temporal ordering) in US-Senate and US-House, these intuitive hypotheses about the relationship between the local structure and the global structure in these networks are not unambiguously supported by other evidence. Similarly, many communities that appear to be "good" based on

ces, the non-zero entries are confined to diagonal bands near the main diagonal. Using our informal terminology from Section 3.1, this is an extreme example of a "hot dog".

the usual intuition and visualisations like that in Fig. 3.5 often are judged as largely artefactual from the perspective of quantitative measures of community quality.

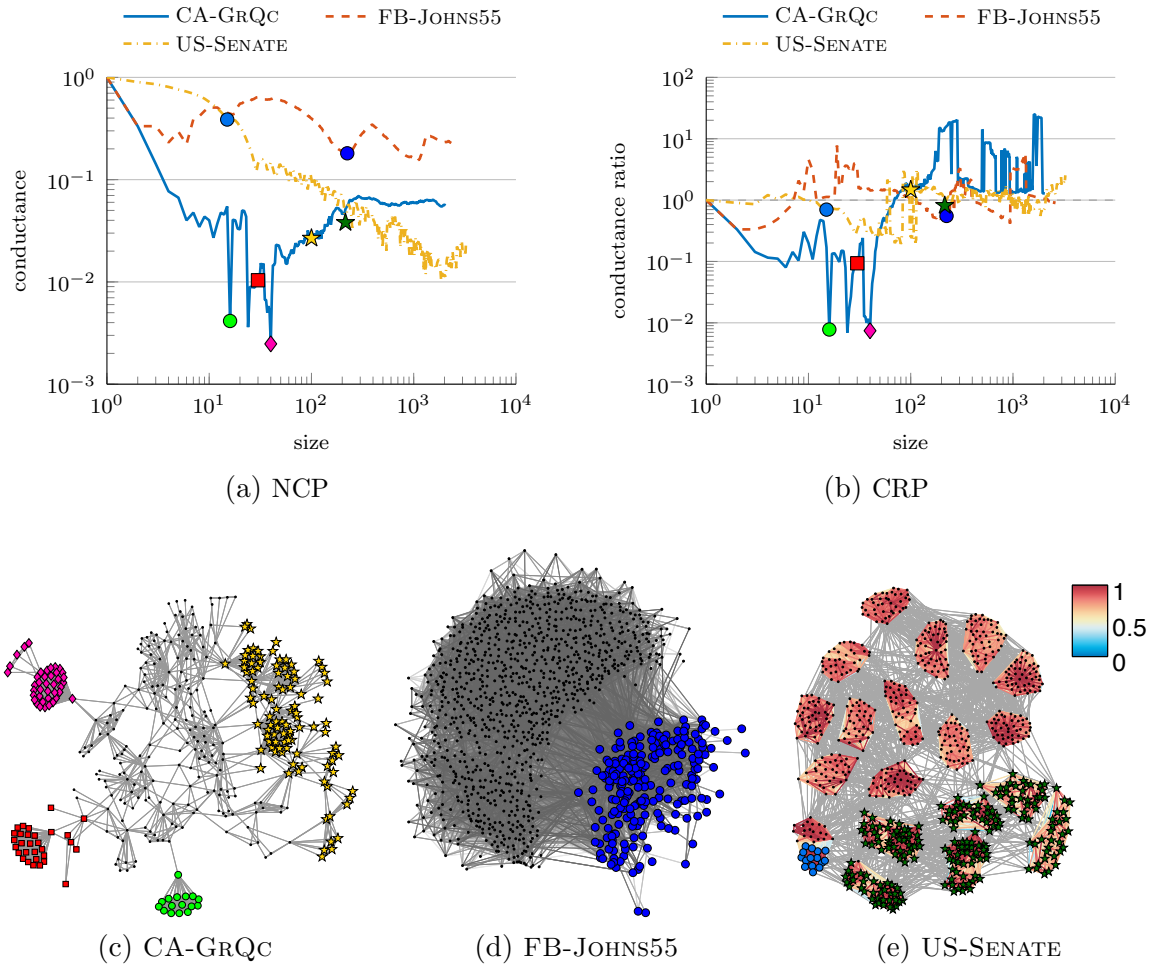
3.4.2 Network Community Profiles (AclCut Method)

We start by presenting our main results from using the AclCut method (see Figs. 3.6 and 3.7). We obtain similar insights about global structure using the MovCut (see Section 3.4.3) and EgoNet (see Section 3.4.4), although they can exhibit rather different local behaviour.

In Fig. 3.6, we show the NCPs and CRPs for the smaller network from each of the three pairs of networks from Table 3.1. In Fig. 3.7, we show the results for the corresponding larger networks. Note the logarithmic scale for both the vertical and horizontal axes in these figures as well as in subsequent NCP and CRP plots. Observe from Figs. 3.6a and 3.7a that the NCPs for networks of the same type are qualitatively similar, whereas NCPs for networks of different types have qualitatively distinct shapes.

- For the co-authorship networks CA-GrQc and CA-AstroPh, the NCPs have a mostly upward-sloping shape, except for the region with fewer than 100 nodes. We conclude that CA-GrQc and CA-AstroPh have good small (e.g., consisting of tens of nodes) communities, but they do not have good large (e.g., consisting of hundreds or thousands of nodes) communities. These results are consistent with the NCPs of LiveJournal from Fig. 3.2b and with the results of [127–129]. Additionally, the high values for the CRPs for the co-authorship networks (see Figs. 3.6b and 3.7b) for communities with hundreds or thousands of nodes reveals that these large communities are loosely connected collections of good, small communities. This feature is also visible in Fig. 3.6c, in which we show selected communities and their neighbourhoods for the CA-GrQc network.

- For the Facebook networks FB-Johns55 and FB-Harvard1, all of the communities at every size (except for two small "communities" with 5 and 10 nodes



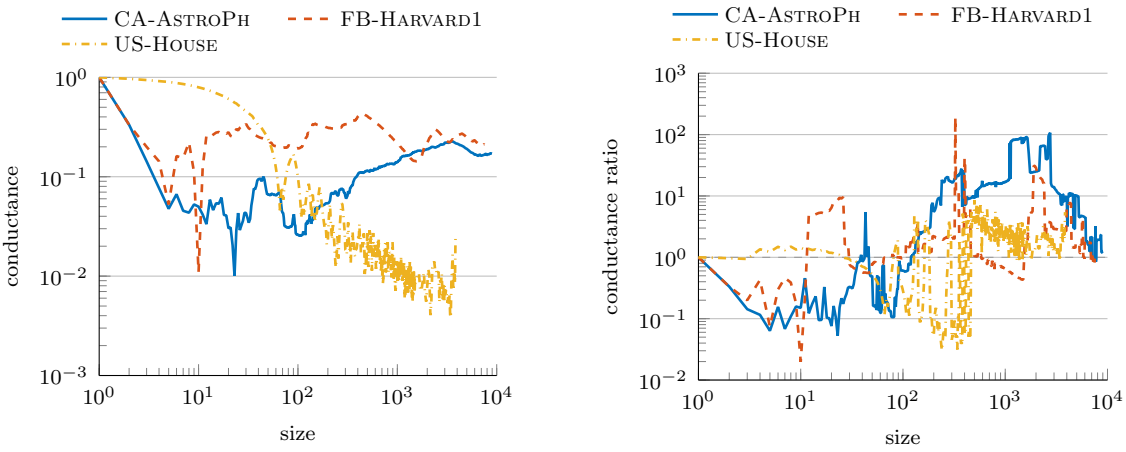
(a) NCP (b) CRP

(c) CA-GrQc (d) FB-Johns55 (e) US-Senate

Figure 3.6: NCP plots [in Panel (a)] and conductance ratio profile (CRP) plots [in Panel (b)] for CA-GrQc, FB-Johns55, and US-Senate (i.e., the smaller network from each of the three pairs of networks from Table 3.1) generated using the AclCut method. In Panels (c)–(e), we show modified Kamada-Kawai [97] spring-embedding visualisations that emphasise community structure (see Appendix C) of corresponding (colour-coded) communities and their neighbourhoods (a2-neighbourhood for CA-GrQc, a 1-neighbourhood for FB-Johns55, and all Senates that have at least one senator in common with those in the communities for US-Senate). We find good small communities but no good large communities in CA-GrQc; some weak large-scale structure in FB-Johns55 that does *not* create substantial bottlenecks to the random-walk dynamics, and signatures of low-dimensional structure (i.e., good large communities but no good small communities) for US-Senate. The low-dimensional structure in US-Senate results from the multilayer structure that encapsulates the network's temporal properties.

in FB-Harvard1[6]) have very large conductances (greater than 10^{-1}). This in-

[6]The two small "communities" in FB-Harvard1 are responsible for the low value of λ_2 for this network compared with all of the other Facebook networks (recall Fig. 3.4). Removing these two communities from the network increases the value of λ_2 by an order of magnitude (from 0.0094 to 0.075), yielding a value that is fairly typical for networks in the Facebook100 data set. In



(a) NCP (b) CRP

Figure 3.7: NCP plots [in Panel (a)] and CRP plots [in Panel (b)] for CA-AstroPh, FB-Harvard1, and US-House (i.e., the larger network in each of the three pairs of networks from Table 3.1) generated using the AclCut method.

dicates that the communities in this network all have very poor community quality, in sharp contrast (though for different reasons) with both the co-authorship and voting networks. The essentially flat shape for the NCPs of the Facebook networks illustrate that these networks have strong expander-like properties (see Section 2.4) and relatedly that there are no substantial bottlenecks to the rapid mixing of random walks on these networks. Both Facebook networks have noticeable dips in their NCPs at larger community sizes (about 220 and 1100 nodes for FB-Johns55 and about 1500 nodes for FB-Harvard1), and the sets of nodes associated with each of these dips correlate strongly with self-reported demographic information[7].

- For the voting networks US-Senate and US-House, the NCP has a predom-

other words, aside from the 15 nodes from these two "communities," FB-Harvard1 looks like FB-Johns55 and the rest of the Facebook100 networks. This is one of the few examples where a feature of an NCP is *not* robust.

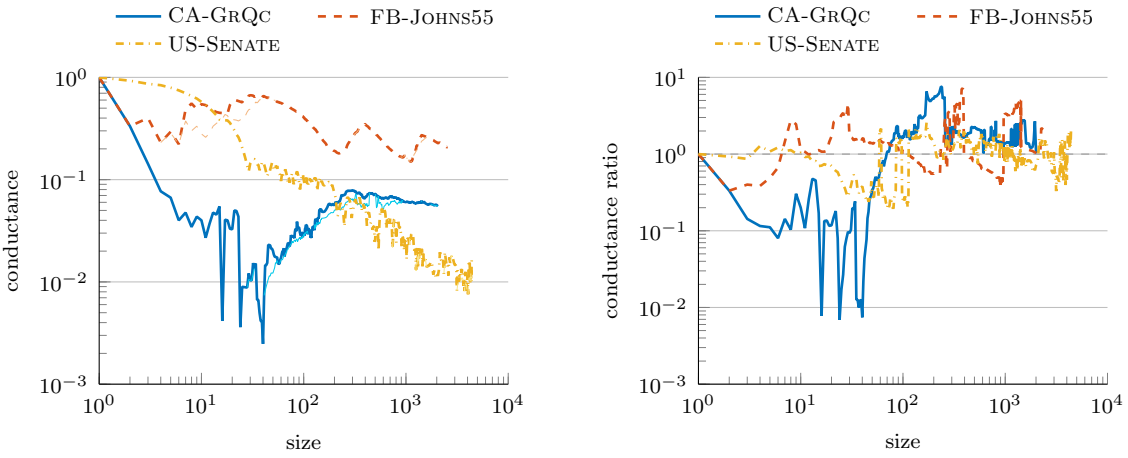
[7]For FB-Johns55, the community of size about 220 corresponds closely to a set of students with the same major, and the community associated with the dip near 1100 corresponds to first-year students. Similarly, the community associated with the dip near 1500 for FB-Harvard1 also corresponds to first-year students. Consistent with the results of Ref. [200], we find similar poorly-connected and moderately-large communities that correlate reasonably well with first-year students in most of the other networks in the Facebook100 data set. (As usual, the most notable exception among the Facebook100 networks is FB-Caltech36, which is known to be influenced much more by dormitory ("House") residence than by class year [200].)

inantly downward-sloping shape. This is characteristic of "low-dimensional" networks, in the sense that we described informally in Section 3.1. Informally, the reason for the downward-sloping shape is that US-Senate and US-House consist of a low-dimensional structure that is evolving along a one-dimensional scaffolding (i.e., time), upon which the detailed structure of individual Congresses (i.e., a good partition that is nearly along party lines) is superimposed. (One can examine such structures by using smaller values of the interlayer coupling parameter; see Ref. [148].) This is consistent with previous results [46].

These results, which illustrate that community quality changes very differently with size in each of the three pairs of networks, also indicate that these three types of networks have very different properties with respect to large-scale versus small-scale community structure. Moreover, the qualitative similarity in behaviour between the two networks in each pair suggests that the coarse behaviour of an NCP (downward-sloping, upward-sloping, or flat) is indicative of large classes of networks and not an artifact of our particular choice of example networks. One obtains similar insights about global structure using the MovCut (see Section 3.4.3) and EgoNet (see Section 3.4.4) methods, although they can exhibit rather different local behaviour. We investigate these differences in local behaviour in Section 3.4.5.

3.4.3 Network Community Profiles (MovCut Method)

The MovCut method provides an alternative way of sampling local community profiles to construct an NCP. Unlike AclCut, which uses *only* local information to obtain good communities, MovCut also incorporates some global information about a network to construct local communities around a seed node. In particular, this implies that there can be sweep sets and thus communities that consist of disconnected components of a network. Such communities have infinitely large conductance ratios. We observe this phenomenon often for the coauthorship and Facebook networks, but it almost never occurs for the Congressional voting networks. These sweep sets consist of several small sets of peripheral nodes, each of which has moderate to very

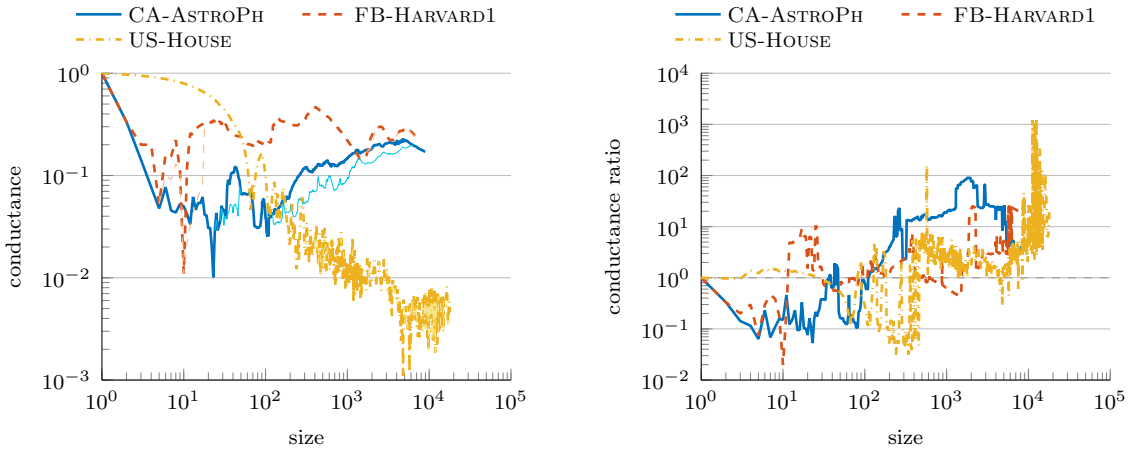


(a) NCP (b) CRP

Figure 3.8: NCP plots [in Panel (a)] and CRP plots [in Panel (b)] for FB-Johns55, CA-GrQc, and US-Senate (i.e., the smaller network in each of the three pairs of networks from Table 3.1) generated using the MovCut method. The thin curves are the NCPs that we obtain when also consider disconnected sweep sets.

low conductance, but which are otherwise unrelated. Although one would not usually think of such a set of nodes as a single good community, optimisation-based algorithms often clump several unrelated communities into a single community for networks with a global core-periphery structure. For completeness and comparison, we include our results both when we keep the disconnected sweep sets and when we restrict our attention to connected communities. As we discuss below, the NCP does not change substantially, although there are some small differences.

The resulting NCPs for the MovCut method (see Figs. 3.8a and 3.9a) are similar to those that we obtained for the AclCut method (see Figs. 3.6a and 3.7a), although there are a few differences worth discussing. The CRP plots are also very similar (compare Figs. 3.8b and 3.9b to Figs. 3.6b and 3.7b). For the coauthorship networks (CA-GrQc and CA-AstroPh), as well as for FB-Harvard1, both MovCut and AclCut identify the same good small communities that are responsible for the spikes in the NCP plots. In addition, the communities that yield the dips in the NCPs for FB-Johns55 near 220 and 1100 nodes, and for FB-Harvard1 near 1500 nodes, all share more than 98% of their nodes. This indicates that both methods are able to find roughly the same community-like structures. However, the results from the MovCut



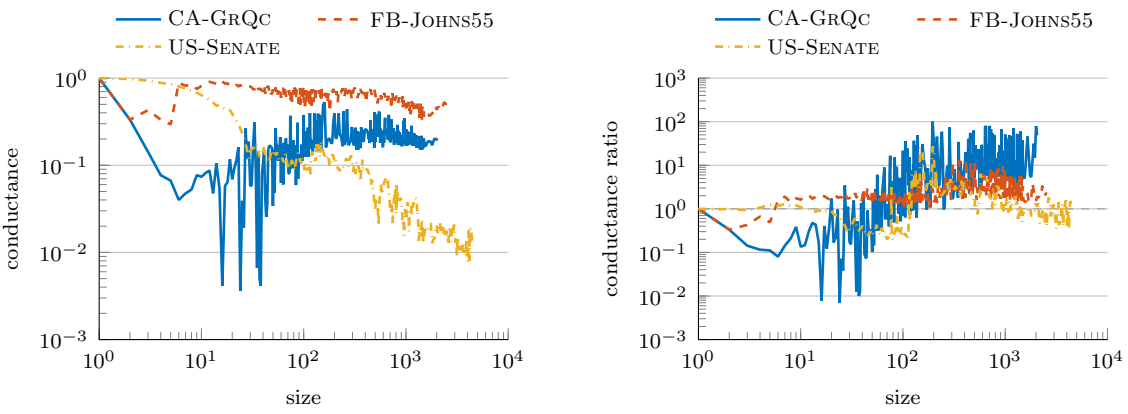
(a) NCP (b) CRP

Figure 3.9: NCP plots [in Panel (a)] and CRP plots [in Panel (b)] for CA-AstroPh, FB-Harvard1, and US-House (i.e., the larger network in each of the three pairs of networks from Table 3.1) generated using the MovCut method. The thin curves are the NCPs that we obtain when we also consider disconnected sweep sets.

NCP for CA-GrQc is higher and less choppy than the one that we computed using AclCut—because the truncation employed by AclCut performs a form of implicit sparsity-based regularisation that is absent from MovCut. See Refs. [73, 140, 141] for a discussion and precise characterisation of this regularisation. For the coauthorship and Facebook networks, we also note that there are regions of the computed NCPs, when using the MovCut method, in which one finds disconnected sweep sets (see the thin curves) with lower conductance than that for the best connected sets of the same size. At other sizes, we see some differences between the NCPs from MovCut and AclCut. This illustrates that the two methods can have somewhat different local behaviour, although both methods produce similar insights regarding the large-scale structure in these networks. In Section 3.4.5, we discuss some of these differences between our results from the two methods in more detail.

3.4.4 Network Community Profiles (EgoNet Method)

Despite its simplicity, and in agreement with Ref. [74], the EgoNet method produces NCPs that are qualitatively similar to those from both the AclCut and MovCut methods, for all of the networks that we consider (see Figs. 3.10 and 3.11). The NCPs



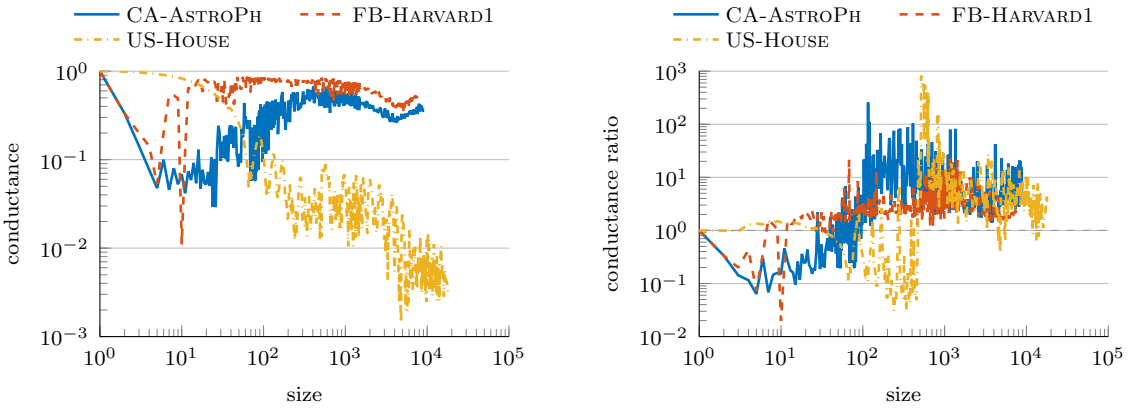
(a) NCP (b) CRP

Figure 3.10: NCP plots [in Panel (a)] and CRP plots [in Panel (b)] for CA-GrQc, FB-Johns55, and US-Senate (i.e., the smaller network in each of the three pairs of networks from Table 3.1) using the EgoNet method. We find qualitatively similar behaviour as with the other two methods, although the NCPs are shifted upwards and some of the large-scale structure is no longer present (especially in the Facebook network).

for the EgoNet method are shifted upwards compared to those for the AclCut and MovCut methods; and this is particularly noticeable at larger community size. This is unsurprising, because the latter two methods more aggressively optimise the conductance objective. However, for all six of our networks, this method preserves an NCP's small-scale structure as well as the global tendency to be upward-sloping, flat, or downward-sloping. This provides further evidence that the qualitative features of an NCP provide a signature of community structure in a network and are not just an artifact of a particular way to sample communities. In Section 3.4.5, we give a more detailed comparison between the results of these methods.

3.4.5 Comparison of Results from AclCut, MovCut, and EgoNet

The NCPs generated using either AclCut or MovCut (see Section 3.4.3), and to a slightly lesser extent using EgoNet (see Section 3.4.4), have similar global features—i.e., they exhibit the same general trends and have dips at small size scales that correspond to nearly identical communities—indicating that we obtain a broadly similar picture of the large-scale community structure by using any of the methods.



(a) NCP (b) CRP

Figure 3.11: NCP plots [in Panel (a)] and CRP plots [in Panel (b)] for CA-AstroPh, FB-Harvard1, and US-House (i.e., the larger networks in each of the three pairs of networks from Table 3.1) using the EgoNet method. We find qualitatively similar behaviour as with the other two methods, although the NCPs are shifted upwards and some of the large-scale structure is no longer present (especially in the Facebook network).

However, the detailed local behaviour of the three methods can differ considerably. Such behaviour depends sensitively on the choice of seed node, the choice of the parameters in the different methods, and the specific details of each method. In this section, we discuss the similarities and differences in the results from these methods. In this section, we only do calculations for the smaller networks from each of the three network pairs in Table 3.1 (but we have observed similar results on the larger networks).

All three methods return a vector that can be used to rank the nodes, and it is this ranking that determines the community memberships. Recall from Section 3.3.4 that both the AclCut and the MovCut methods return a variant of a PPR vector, which we parametrise using the teleportation parameter α. Additionally, the approximate calculation performed by the AclCut method depends on the truncation parameter ϵ. The EgoNet method returns an EgoRank vector that only depends on the seed node and does not have any parameters.

To compare different methods, we note that any meaningful difference between them should manifest as a difference in the rank order of nodes, as this determines the assignment of nodes to local communities. We quantify rank differences by com-

puting the Spearman rank correlation [191] between the (generalised for MovCut and approximate for AclCut) PPR and EgoRank ranking vectors. We also restrict all comparisons to the support of the corresponding approximate PPR vector that we obtain using the AclCut method. This induces an indirect dependency of the results from MovCut and EgoNet on α and ϵ (in addition to the direct dependency of MovCut on α).

In Tables 3.2–3.4, we show the results of our calculations of Spearman rank correlations. For each of the three networks, we select 50 seed nodes by sampling uniformly without replacement. (Recall that a single run of a method uses a single seed node.) We then compute PPR vectors for these seed nodes using the AclCut and MovCut method for different values of the truncation parameter ϵ and teleportation parameter α, and we also compute the EgoRank vector for each of the seed nodes. Recall that *smaller* values of α correspond to *more local* versions of the procedures, but *larger* values of ϵ correspond to *more local* versions of the procedures.

The AclCut and MovCut methods give very similar results for most of the 50 seed nodes in our sample, although some seed nodes do yield noticeable differences. The two methods give the most similar results for FB-Johns55 (a mean Spearman correlation of 0.92, a minimum of 0.43), whereas we find larger deviations in both CA-GrQc (mean 0.85; minimum -0.13) and US-Senate (mean 0.86; minimum -0.44). Note that we calculate the mean, maximum, and minimum over all sampled seed nodes and parameter values.

Interestingly, the largst deviations between the two methods for CA-GrQc and US-Senate occur at different values of the truncation parameter ϵ. For CA-GrQc (and, to a lesser extent, for FB-Johns55), we obtain the largest deviations for smaller values (e.g., $\epsilon = 10^{-6}$). For US-Senate, however, we obtain the largest deviations for $\epsilon = 10^{-4}$. (See the bold values in Tables 3.2–3.4.) This is consistent with the very different isoperimetric properties of these three networks, as revealed by their NCPs, as well as with well-known connections between conductance and random walks.

There are two potential causes for the differences between the AclCut and MovCut method. First, there is a truncation effect, governed by the parameter ϵ,

			α													
			0.6			0.7			0.8			0.9			0.99		
			A-M	A-E	M-E	A-M	A-E	M-E	A-M	A-E	M-E	A-M	A-E	M-E	A-M	A-E	M-E
ϵ	10^{-3}	max	1.00	0.99	0.99	1.00	0.98	0.98	1.00	0.97	0.97	1.00	0.95	0.95	0.98	0.89	0.86
		mean	0.98	0.78	0.77	0.98	0.76	0.73	0.98	0.72	0.68	0.97	0.68	0.62	0.91	0.60	0.48
		min	0.92	0.26	0.23	0.91	0.18	0.14	0.94	0.21	0.15	0.85	−0.01	−0.05	0.74	0.25	0.06
	10^{-4}	max	1.00	0.97	0.97	1.00	0.97	0.97	1.00	0.94	0.94	0.99	0.89	0.85	**0.92**	0.69	0.53
		mean	0.99	0.74	0.72	0.98	0.73	0.68	0.97	0.70	0.64	0.95	0.63	0.54	0.89	0.51	0.36
		min	0.96	0.16	0.10	0.91	−0.05	−0.19	0.84	0.35	−0.02	0.88	0.43	0.25	0.85	0.32	0.18
	10^{-5}	max	1.00	0.96	0.95	0.97	0.92	0.87	0.94	0.81	0.67	0.89	0.73	0.55	0.93	0.75	0.60
		mean	0.91	0.74	0.58	0.89	0.69	0.51	0.85	0.65	0.42	0.78	0.62	0.33	**0.84**	0.63	0.36
		min	0.24	0.21	−0.20	0.42	0.30	−0.10	0.42	0.39	−0.13	0.25	0.43	−0.10	**0.49**	0.44	0.05
	10^{-6}	max	**0.84**	0.85	0.68	**0.79**	0.81	0.49	**0.70**	0.79	0.39	**0.80**	0.81	0.47	0.93	0.75	0.60
		mean	**0.62**	0.72	0.25	**0.57**	0.69	0.17	**0.51**	0.69	0.12	**0.51**	0.72	0.13	0.85	0.63	0.37
		min	**0.01**	0.57	−0.36	**0.07**	0.52	−0.30	**−0.06**	0.50	−0.27	**−0.13**	0.57	−0.32	0.52	0.46	0.08

Table 3.2: Pairwise comparison of the three methods using the Spearman rank correlation between the rank vectors from the AclCut (A), MovCut (M), and EgoNet (E) methods for CA-GrQc. We use a uniform random sample of 50 nodes for each of several values for the teleportation parameter α and truncation parameter ϵ. We take the maximum, mean, and minimum over the seed nodes. Bold values highlight the largest deviations between AclCut and MovCut methods for a given value of α.

			α													
			0.6			0.7			0.8			0.9			0.99		
			A-M	A-E	M-E	A-M	A-E	M-E	A-M	A-E	M-E	A-M	A-E	M-E	A-M	A-E	M-E
ϵ	10^{-3}	max	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		mean	1.00	0.85	0.84	1.00	0.82	0.81	0.99	0.79	0.79	0.98	0.78	0.77	0.98	0.78	0.75
		min	0.95	0.61	0.56	0.94	0.54	0.47	0.80	0.55	0.53	0.80	0.55	0.49	0.77	0.40	0.39
	10^{-4}	max	1.00	0.89	0.89	1.00	0.93	0.93	1.00	0.91	0.91	1.00	0.89	0.89	1.00	0.88	0.88
		mean	0.98	0.48	0.47	0.97	0.45	0.44	0.96	0.43	0.41	0.95	0.41	0.37	0.94	0.38	0.33
		min	0.81	0.06	0.05	0.78	0.05	0.04	0.74	0.07	0.06	0.69	0.07	0.07	0.69	0.07	0.00
	10^{-5}	max	1.00	0.85	0.85	1.00	0.86	0.85	1.00	0.84	0.83	0.99	0.84	0.82	1.00	0.81	0.78
		mean	0.98	0.58	0.54	0.97	0.59	0.54	0.97	0.63	0.57	0.96	0.61	0.54	0.95	0.53	0.46
		min	0.90	0.24	0.18	0.91	0.26	0.17	0.91	0.23	0.14	0.89	0.22	0.07	0.88	0.18	0.04
	10^{-6}	max	**0.99**	0.75	0.46	**0.97**	0.69	0.46	**0.91**	0.71	0.39	**0.90**	0.73	0.47	**0.97**	0.69	0.57
		mean	**0.79**	0.41	0.20	**0.75**	0.45	0.13	**0.72**	0.56	0.20	**0.77**	0.62	0.32	**0.88**	0.59	0.41
		min	**0.57**	0.23	−0.07	**0.49**	0.24	−0.06	**0.43**	0.32	−0.02	**0.49**	0.32	0.01	**0.60**	0.26	0.07

Table 3.3: Pairwise comparison of the three methods using the Spearman rank correlation between the ranking vectors from the AclCut (A), MovCut (M), and EgoNet (E) methods for FB-Johns55. We use a uniform random sample of 50 nodes for each of several values for the teleportation parameter α and truncation parameter ϵ. We take the maximum, mean, and minimum over the seed nodes. Bold values highlight the largest deviations between AclCut and MovCut methods for a given value of α.

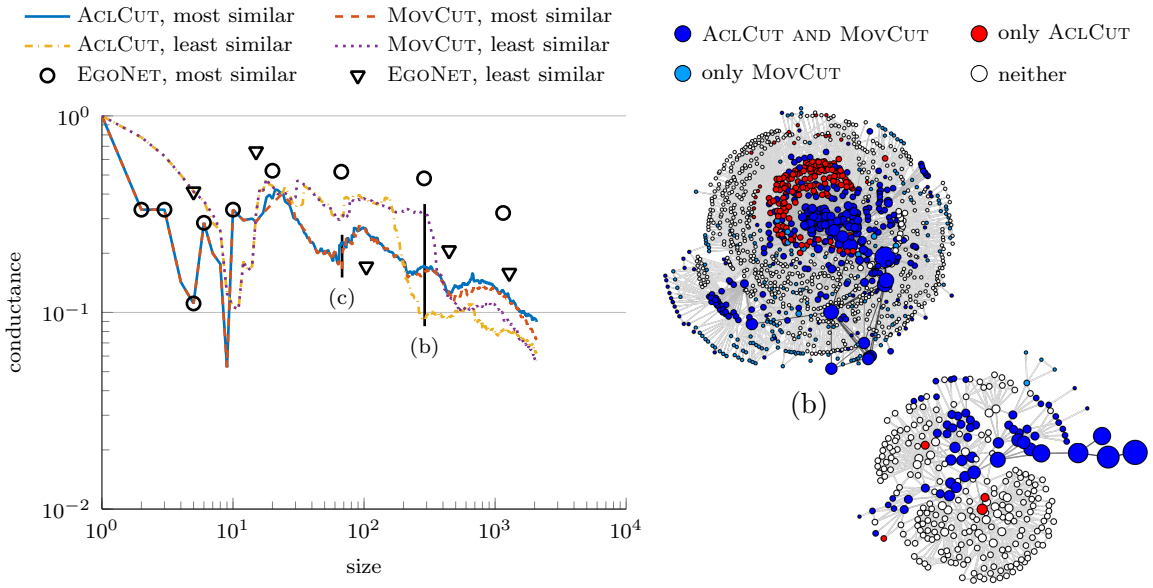
		α														
		0.6			0.7			0.8			0.9			0.99		
		A-M	A-E	M-E	A-M	A-E	M-E	A-M	A-E	M-E	A-M	A-E	M-E	A-M	A-E	M-E
ϵ	max	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10^{-3}	mean	**0.65**	0.66	0.40	0.68	0.71	0.30	0.74	0.77	0.46	**0.74**	0.85	0.57	**0.80**	0.80	0.56
	min	0.02	0.50	−0.05	0.38	0.32	−0.24	0.46	0.53	−0.07	0.50	0.68	−0.02	**0.51**	0.62	0.00
	max	1.00	0.93	0.87	1.00	0.92	0.77	1.00	0.93	0.82	1.00	0.89	0.75	**0.93**	0.81	0.61
10^{-4}	mean	0.75	0.61	0.36	**0.54**	0.59	0.17	**0.67**	0.62	0.34	0.86	0.50	0.34	0.92	0.36	0.23
	min	**−0.03**	−0.16	−0.51	**−0.32**	−0.11	−0.63	**−0.44**	0.15	−0.36	**0.50**	−0.08	−0.26	0.77	−0.20	−0.38
	max	1.00	0.91	0.90	1.00	0.89	0.87	1.00	0.85	0.83	1.00	0.80	0.78	1.00	0.85	0.84
10^{-5}	mean	0.99	0.58	0.54	0.99	0.48	0.46	0.99	0.42	0.40	0.96	0.44	0.40	0.95	0.43	0.39
	min	0.86	0.03	0.01	0.88	−0.02	−0.02	0.88	−0.09	−0.13	0.80	−0.17	−0.20	0.80	0.07	0.01
	max	1.00	0.84	0.84	1.00	0.84	0.84	1.00	0.84	0.85	1.00	0.87	0.86	1.00	0.96	0.96
10^{-6}	mean	0.94	0.54	0.48	0.98	0.55	0.50	0.99	0.62	0.60	1.00	0.69	0.68	0.99	0.91	0.89
	min	0.85	0.30	0.22	0.89	0.08	0.19	0.98	0.25	0.21	0.98	0.30	0.29	0.96	0.83	0.73

Table 3.4: Pairwise comparison of the three methods using the Spearman rank correlation between the ranking vectors from the AclCut (A), MovCut (M), and EgoNet (E) methods for US-Senate. We use a uniform random sample of 50 nodes for each of several values for the teleportation parameter α and truncation parameter ϵ. We take the maximum, mean, and minimum over the seed nodes. Bold values highlight the largest deviations between AclCut and MovCut methods for a given value of α.

in approximating the PPR vector using the AclCut method. As ϵ becomes smaller, the approximation in AclCut becomes more accurate and this effect diminishes. Second, the two methods differ in the precise way that they use a seed vector to represent a seed node. Recall that the AclCut method uses an indicator vector $\boldsymbol{s}$ to represent a seed node i; thus, we use $\boldsymbol{s}_i = 1$ whenever i is a seed node, and we set all other entries in that vector to 0. In contrast, the MovCut method projects the indicator vector onto the orthogonal complement of the strength vector to ensure that $\boldsymbol{s}^T D\mathbf{1} = 0$ (see Section 3.3). This effect decreases as $\alpha \to 1$.

The largest deviations between the two methods occur for smaller values of ϵ in CA-GrQc and FB-Johns55; for these, the truncation effect is small, suggesting that the different way of representing a seed node is partially responsible for the difference between the results of the two methods for these networks. For larger values of ϵ (in particular, $\epsilon \geq 10^{-4}$), where the support of the approximate PPR vector from the AclCut method is small, the behaviour of the two methods is very similar. Consequently, the differences in the choice of seed vector become more important for nodes that are "far away" from the seed node, in the sense that they are rarely visited by the personalised PageRank dynamics that underlie these methods. As a result, the "local NCPs" for the two methods in Figs. 3.12a and 3.13a are largely identical for small community sizes but diverge for large community sizes. (We use the term *local NCP* to refer to an NCP that we computed using only a single seed node without optimising over the results from multiple seed choices; see Ref. [142] for details on the construction of local NCPs.)

For US-Senate, the two methods behave almost identically for small ϵ (see Table 3.4), so we conclude that the different ways of representing a seed node have only a small effect on this network. However, the truncation effect is more pronounced in this network compared with CA-GrQc or FB-Johns55. This feature manifests as larger deviations between AclCut and MovCut in Table 3.4 for large ϵ and small α (i.e., where the truncation has the strongest impact). The discrepancy occurs because the AclCut method initially pushes a large amount of probability to the interlayer neighbours of the seed node (i.e., to the same Senator in different Congresses). This



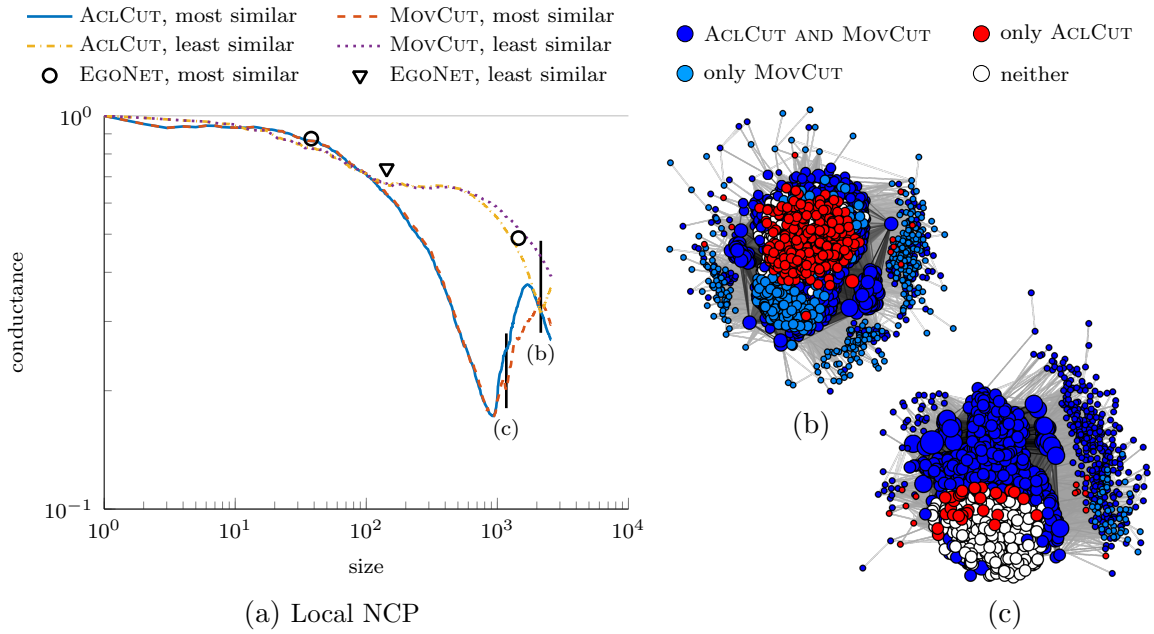
(a) Local NCP

Figure 3.12: Comparison of methods for CA-GrQc. (a) Local NCPs for the seed nodes (of the 50 nodes that we sample) with the highest and lowest mean Spearman correlation over the sampled parameter values. These NCPs highlight the difference in behaviour of the AclCut and MovCut methods for large communities. (b)–(c) Kamada-Kawai-like spring-embedding visualisation (see Appendix C) of (b) the 6-neighbourhood of the seed node with the largest difference between the two methods and (c) the 9-neighbourhood of the seed node with the smallest difference. In these two visualisations, the node size decreases as a function of geodesic distance from the seed node. We colour the nodes according to whether they belong to the local community that we obtain using the AclCut method, the one that we obtain using the MovCut method, the intersection of the two communities, or neither community.

probability does not diffuse to other nodes for sufficiently large values of ϵ.

In Figs. 3.12–3.14, we illustrate the results from Tables 3.2–3.4. In these figures, we plot the local NCPs for CA-GrQc, FB-Johns55, and US-Senate for the seed nodes (from the sample of 50) that yield the highest and lowest mean Spearman rank correlation between the AclCut and MovCut methods. In these figures, we also include visualisations of example communities that we obtain from the AclCut and MovCut methods using a Kamada-Kawai-like spring-embedding visualisation (see Appendix C) of the k-ego-nets of these seed nodes.

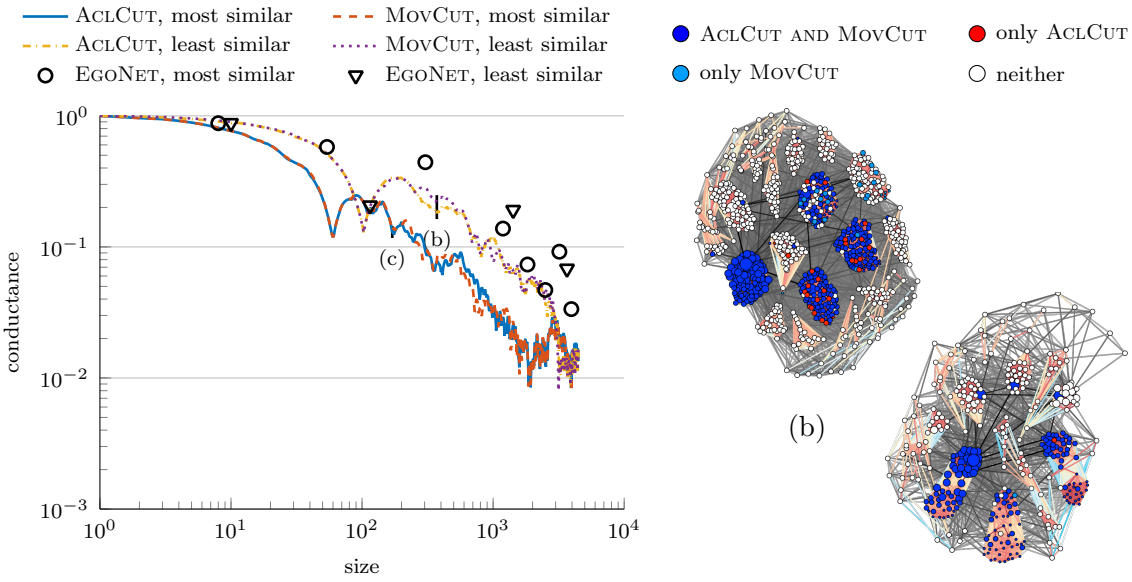
From the visualisations of the local communities, it seems for CA-GrQc (see Fig. 3.12) and FB-Johns55 (see Fig. 3.13) that nodes included in local communities



(a) Local NCP

Figure 3.13: Comparison of methods for FB-Johns55. (a) Local NCPs for the seed nodes (of the 50 nodes that we sample) with the highest and lowest mean Spearman correlation over the sampled parameter values. These NCPs highlight the difference in behaviour for the two methods for large communities. (b)–(c) Kamada-Kawai-like spring-embedding visualisation (see Appendix C) of the 2-neighbourhoods of both seed nodes, (b) the one with the largest difference and (c) the one with the smallest difference. In these two visualisations, the node size decreases as a function of geodesic distance from the seed node. The smallest nodes are more than 2 steps away from the seed node, but they appear in at least one of the local communities. We colour the nodes according to whether they belong to the local community that we obtain using the AclCut method, the one that we obtain using the MovCut method, the intersection of the two communities or neither community.

obtained from AclCut tend to be closer in geodesic distance than those obtained from MovCut to the seed node. (To see this, observe that red nodes tend to be larger than light blue nodes in the visualisation of the k-neighbourhoods.) If this observation holds more generally and is not just an artifact of the particular communities that we show in Figs. 3.12 and 3.13, then we should obtain higher Spearman rank correlations between AclCut and EgoNet than between MovCut and EgoNet. Indeed, Tables 3.2–3.4 consistently show this effect for all choices of ϵ and α and for all three networks. Note that this effect is also present in US-Senate, though it is less prominent in its k-neighbourhood visualisation than is the case for the other two networks.



(a) Local NCP

Figure 3.14: Comparison of methods for US-Senate. (a) Local NCP for the seed nodes (of the 50 nodes we sample) with the highest and lowest average Spearman correlation over sampled parameter values. These NCPs highlight the difference in behaviour for the two methods for large communities. (b)–(c) Kamada-Kawai-like spring-embedding visualisation (see Appendix C) of the 3-neighbourhoods of both seed nodes, (b) the one with the largest difference and (c) the one with the smallest difference. In these two visualisations, the node size decreases as a function of geodesic distance from the seed node. We colour the nodes according to whether they belong to the local community that we obtain using the AclCut method, the one that we obtain using the MovCut method, the intersection of the two communities, or neither community.

Figures 3.12–3.14 also reveal that the three networks look very different from a local perspective. For FB-Johns55 (see Fig. 3.13), the two seed nodes (of the sample of 50) that we explore in detail allow us to reach a large fraction of all nodes after just two steps. This is consistent with known properties of the full Facebook graph (circa 2012) of individuals connected by reciprocal "friendships." For example, the mean geodesic distance between pairs of nodes of the Facebook graph is very small. It was recently estimated by Facebook's Data Science Team and their collaborators to be about 4.74 [10]. Additionally, one can view Facebook as a collection of ego networks that have been patched together into a network whose global structure is sparse [201] (and such structure is an important motivation for the locally-biased notion of community structure that we advocate).

For CA-GrQc, we obtain very different neighbourhoods starting from our two different seed nodes. The node in the sample that results in the largest difference in behaviour between the AclCut and MovCut methods appears to be better connected in the network in the sense that the k-neighbourhood (for any k until saturation occurs) is much larger than that of the node that results in the smallest difference. That is, it is more in the "core" than in the "periphery" of the nested core-periphery structure of Refs. [127, 128]. We observe a similar phenomenon for FB-Johns55 and US-Senate. Furthermore, its 1-ego-net and 2-ego-net are highly clustered, in the sense that they contain many closed triangles. For the seed node that that results in the smallest difference between the AclCut and MovCut methods, we need to consider the 6-ego-net (which has 20 nodes) to obtain a network of similar size to the 2-ego-net (which has 15 nodes) for the seed node with the largest difference. In the case of the seed node in our sample that results in the smallest difference between the two methods, even the 6-ego-net appears rather tree-like; it contains few closed triangles and no larger cliques.

For US-Senate, the 1-neighbourhood of any seed node contains only the node itself and those corresponding to the same Senator in different Congresses[8]. As one begins to consider nodes that are further away, one first reaches corresponding Senators in other Congresses before reaching other Senators with similar voting patterns from the same Congress. This behaviour of the EgoNet method contrasts with the (PageRank-based) AclCut and MovCut methods, which tend to initially select all Senators from one Congress before reaching Senators from other Congresses.

3.4.6 Meso-Scale Structure

From the perspective of the locally-biased community-detection methods one can view intermediate-sized (i.e., meso-scale) structures in networks as arising from collections

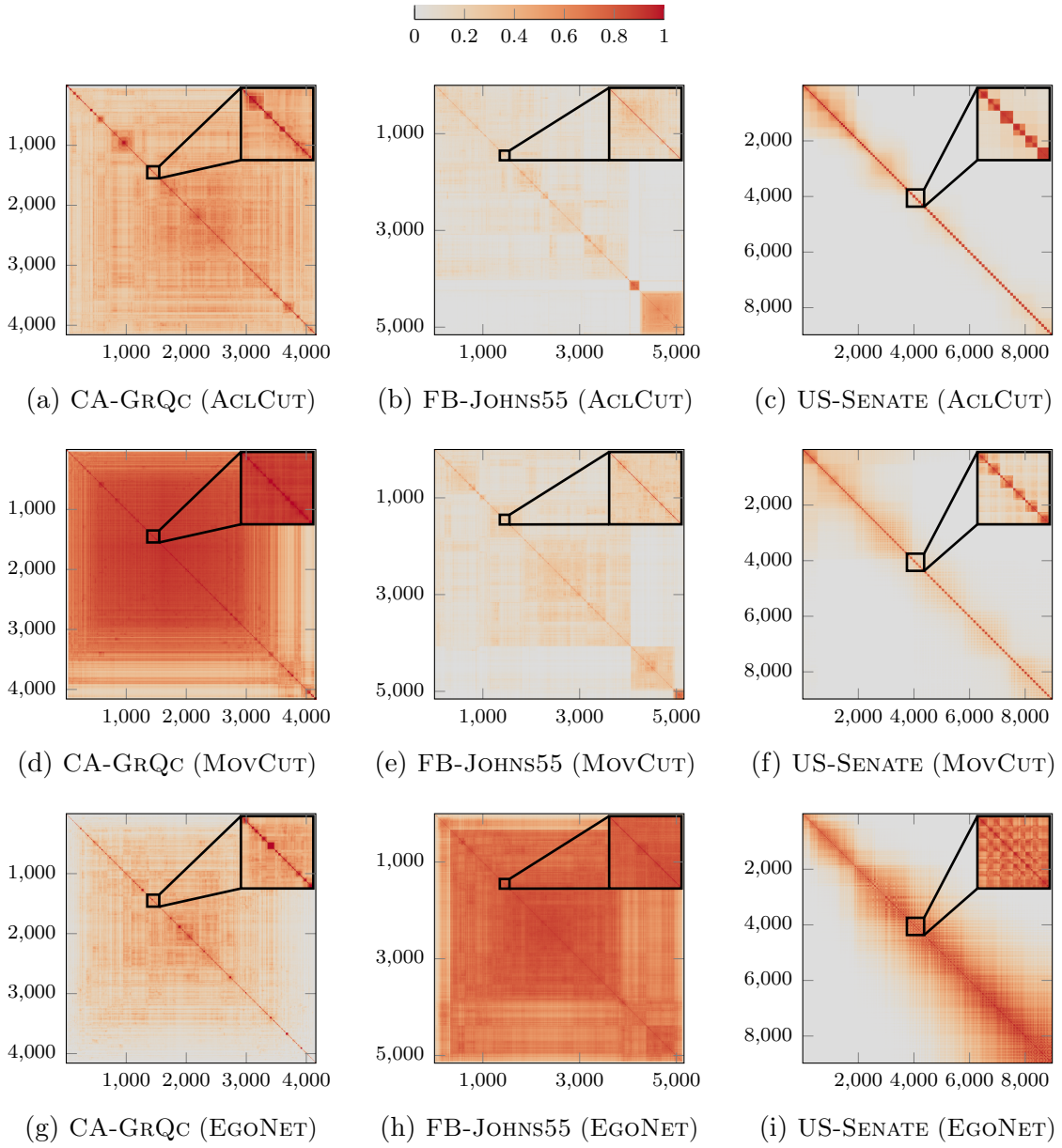
[8]This holds for all seed nodes because the maximum voting similarity between a pair of Senators is 1 (i.e., they voted the same way on every bill) and all interlayer edges (i.e., edges between the same Senator in different Congresses) have a weight of 1. As one reduces the weight of interlayer edges, one needs to chose an increasingly large value of k for nodes from interlayer edges to appear in the k-neighbourhood. Hence, an increasing number of Senators from the same Congress can appear in a k-neighbourhood that does not contain any nodes from interlayer edges.

of local features—e.g., via overlaps of local communities that one obtains algorithmically using locally-biased dynamics such as those that we consider. Such local features depend not only on the network adjacency matrix but also on the dynamical process under study, the initial seed(s) from which one is viewing a network, and the locality parameters (e.g., α and ϵ for AclCut) of the method (which corresponds to the dynamical process) that determine how locally one is viewing the network.

To try to visualise the meso-scale and global network structures that we obtain from the local communities that we identify, we define an $n \times n$ association matrix $\tilde{\boldsymbol{A}}$ (where n is again the number of nodes in the network), which encodes pairwise relations between nodes based on a sample of local communities. For a given sample $\mathcal{S}$ of local communities (obtained, e.g., by running a given method with many seed nodes and values of a locality parameter), the entries of the association matrix are given by the number of times that a pair of nodes appear together in a local community, normalised by the number of times either of them appears. That is, the elements of the association matrix are

$$\tilde{\boldsymbol{A}}_{ij} = \frac{|\{S \in \mathcal{S} : i \in S \text{ and } j \in S\}|}{|\{S \in \mathcal{S} : i \in S \text{ or } j \in S\}|} \,. \tag{3.13}$$

Note that this definition of an association matrix differs from the association matrix we discuss in Section 2.5.2 in the way that it is normalised. The different normalisation in Eq. (3.13) compared to Eq. (2.39) is necessary to correct for the oversampling of large communities relative to small communities which results from sampling nodes uniformly at random. At first glance, association matrices computed by sampling a modularity landscape appear to reveal much clearer community structure in these networks than what we obtain by sampling local communities. However, this is largely an artifact of choosing a fixed resolution when optimising modularity, whereas the local communities we sample span many different size scales. If one uses one of the multi-resolution generalisations of modularity (see the discussion in Section 2.5.1) to sample the modularity landscape across different values of the resolution parameter, one obtains association matrices that are similar in appearance to



(a) CA-GrQc (AclCut) (b) FB-Johns55 (AclCut) (c) US-Senate (AclCut)

(d) CA-GrQc (MovCut) (e) FB-Johns55 (MovCut) (f) US-Senate (MovCut)

(g) CA-GrQc (EgoNet) (h) FB-Johns55 (EgoNet) (i) US-Senate (EgoNet)

Figure 3.15: Visualisations of association matrices (see Eq. (3.13)) for CA-GrQc, FB-Johns55, and US-Senate illustrate how meso-scale and global structures emerge from the superposition and overlap of many local communities. For each of the three networks, we generate the panels using the same three sampling procedures that we use to generate the NCPs: we use AclCut for Panels (a)–(c), MovCut for Panels (d)–(f), and EgoNet for Panels (g)–(i).

the ones that we obtain by sampling local communities.

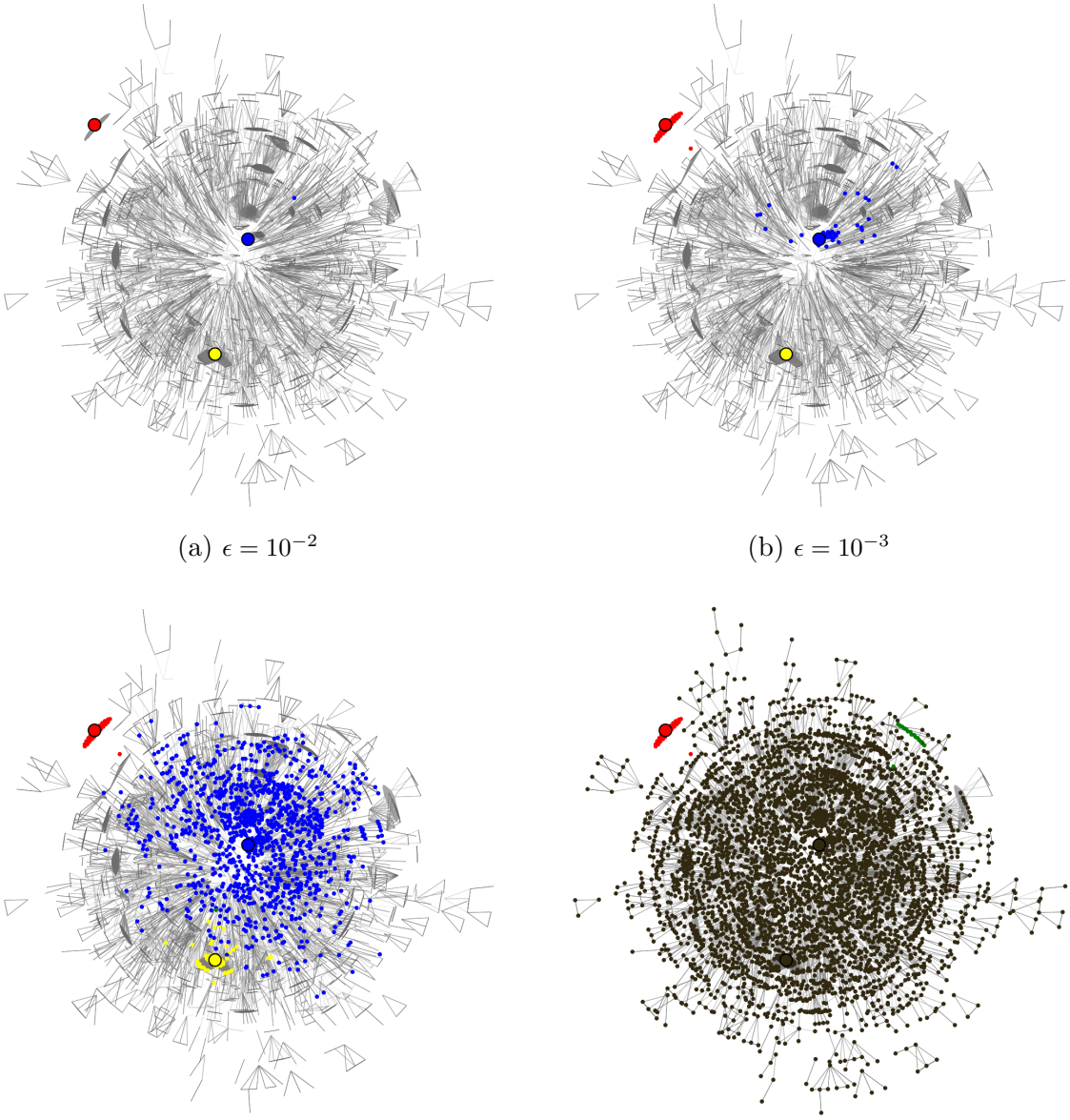
To visualise the association matrices in a way that reveals global network structure, it is important to find a good node order. We found the sorting method suggested in Ref. [179] to be impractically slow for the networks that we study. Instead, we sort the nodes based on the optimal leaf ordering [13] for the average-linkage hierarchical clustering tree of the association matrix. (For US-Senate, we do this procedure within a given Congress, and we then use the natural temporal ordering to define the inter-Congressional ordering.)

In addition, to see small-scale structure using samples $\mathcal{S}$ obtained from MovCut, we use a community-size parameter c (see [142] for details) that limits the volume of the resulting community based on the desired correlation with the seed vector. In this paper, we use $c \in \{10^i : i = 1, \ldots, 5\}$.

We summarise our results in Figs. 3.15–3.18.

In Fig. 3.15, we show the result of applying the above procedure with communities that we sample using the AclCut, MovCut, and EgoNet methods. In each case, we keep only the best conductance community for each sampled ranking vector. The most obvious feature of the visualisations in Fig. 3.15 is that—except for US-Senate, for which there is a natural large-scale global structure defined by the one-dimensional temporal ordering—the visualisations are much more complicated than any of the idealised structures in Fig. 3.1 (which suggests that the visualisations might be revealing at least as much about the inner workings of the visualisation algorithm as about the networks being visualised). The structures in Fig. 3.1 are trivially interpretable, whereas those in real networks (e.g., as illustrated in Fig. 3.15) are extremely messy and very difficult to interpret. In the paragraphs below, we will discuss the structural features in Fig. 3.15 in more detail.

For CA-GrQc (see Fig. 3.16 as well as Fig. 3.15), we observe many small communities that are composed of about 10–100 nodes. These communities, which correspond to the dark red blocks along the diagonal (see the inset in Fig. 3.15a), are responsible for the dips in the NCPs (see Figs. 3.6a, 3.8a, and 3.10a) for this network. However, these small communities do not combine to form large communities, which



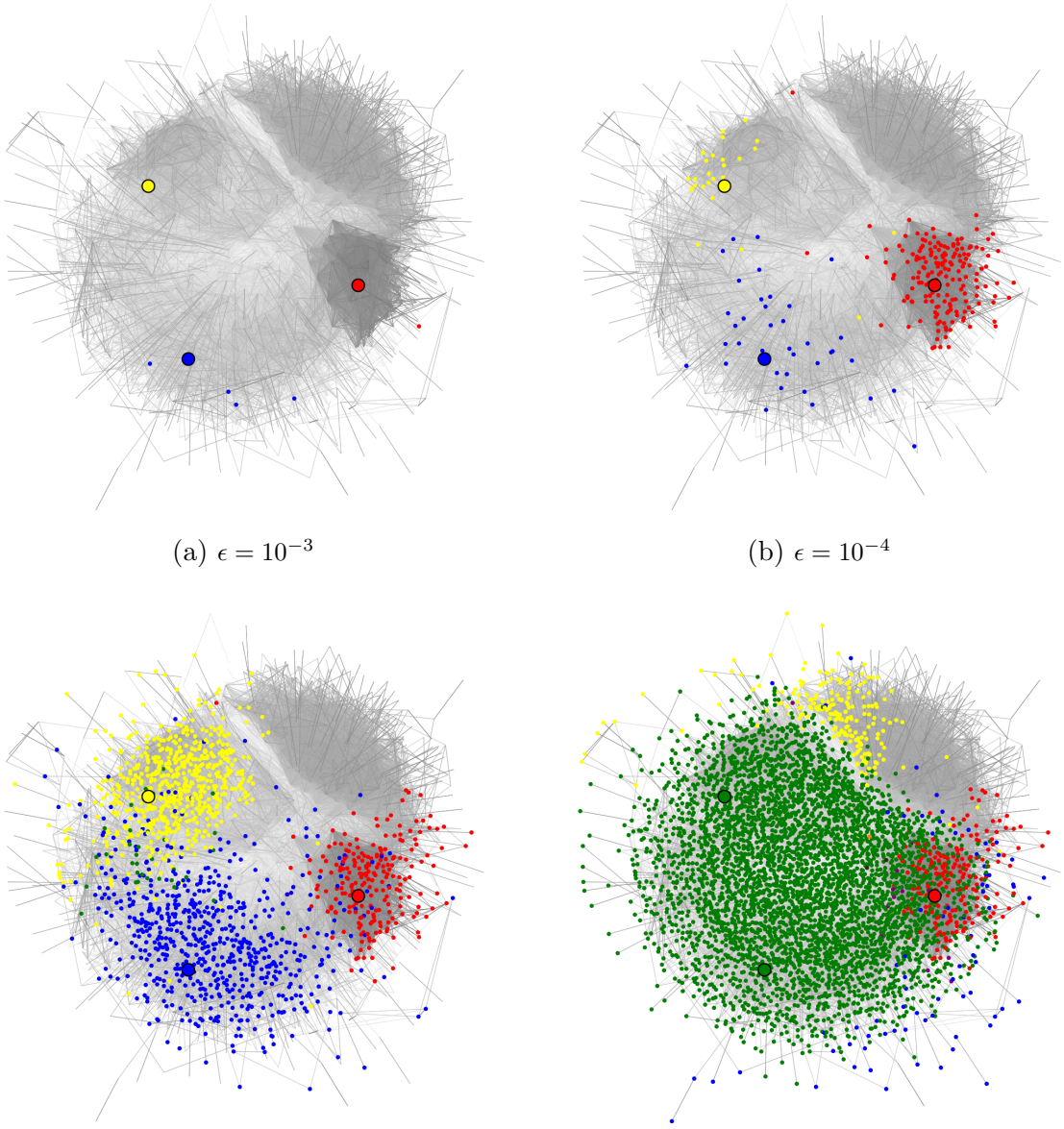
(a) $\epsilon = 10^{-2}$ (b) $\epsilon = 10^{-3}$

(c) $\epsilon = 10^{-4}$ (d) $\epsilon = 10^{-5}$

Figure 3.16: Visualisation of global structure in CA-GrQc. We construct the network layout by weighting each edge using the corresponding entry of the association matrix for the AclCut method (Fig. 3.15a). We then apply the spring embedding visualisation algorithm (see Appendix C) to the resulting weighted network. For ease of visualisation, we only plot edges whose weight is larger than the mean edge weight. Coloured nodes correspond to local communities for three different seed nodes. We draw nodes that are members of more than one community in a mixed colour (e.g., blue and yellow become green; and blue, yellow, and red become blackish). As we decrease the truncation (i.e., locality bias) parameter ϵ, the different communities first explore local structure before merging. In Panel (d), each community covers most of the network.

would result in large diagonal blocks in the association matrices. Instead, the small communities appear to amalgamate into a single large block (or "core"). In Fig. 3.16, we aim to make this observation more intuitive by showing how the local communities for three different seed nodes spread through the network as we change the locality bias parameter ϵ. We construct the weighted network $\tilde{G} = (V, E, \tilde{w})$ in Fig. 3.16 from the unweighted CA-GrQc network $G = (V, E)$ using the association matrix for the AclCut method (see Fig. 3.15a). We assign each edge $(i, j) \in E$ a weight based on the corresponding entry of the association matrix, i.e., $\tilde{w}((i, j)) = \tilde{\boldsymbol{A}}_{ij}$ if $(i, j) \in E$ and $\tilde{w}((i, j)) = 0$ otherwise. Based on our earlier results in Sections 3.4.2–3.4.4 which show that this network has slowly-increasing NCP, as well as previous results in [127–129], we interpret these features shown in Fig. 3.16 in terms of a nested core-periphery structure, in which the network periphery consists of relatively good communities and the core consists of relatively densely connected nodes.

For FB-Johns55 (see Fig. 3.17 as well as Fig. 3.15), we observe two relatively large communities, which correspond to the two large diagonal blocks in Figs. 3.15be and which underlie the dips in the NCPs in Figs. 3.6a and 3.8a. Note, however, from the scale of the vertical axis in Figs. 3.6a and 3.8a that the community quality of these communities is very low, so one should actually construe the visualisation in Figs. 3.15be as highlighting a low-quality community that is only marginally better than the other low-quality communities that are present in that network. Based on this visualisation as well as our earlier results, the remainder of FB-Johns55 does not appear to have much community structure (at least based on using the conductance diagnostic to measure internal versus external connectivity). However, there do appear to be some remnants of highly overlapping communities that one could potentially identify using other methods (e.g., the one in Ref. [211]). The EgoNet method (see Fig. 3.15h) is unable to resolve not only these small communities but also the larger low-quality communities. In Figure 3.17 we show how the local communities for two seed nodes that do not belong to one of the two large communities slowly spread and eventually merge (blue and yellow nodes), whereas the red community (which corresponds to the smaller of the two communities) is quickly identified and



(a) $\epsilon = 10^{-3}$ (b) $\epsilon = 10^{-4}$

(c) $\epsilon = 10^{-5}$ (d) $\epsilon = 10^{-6}$

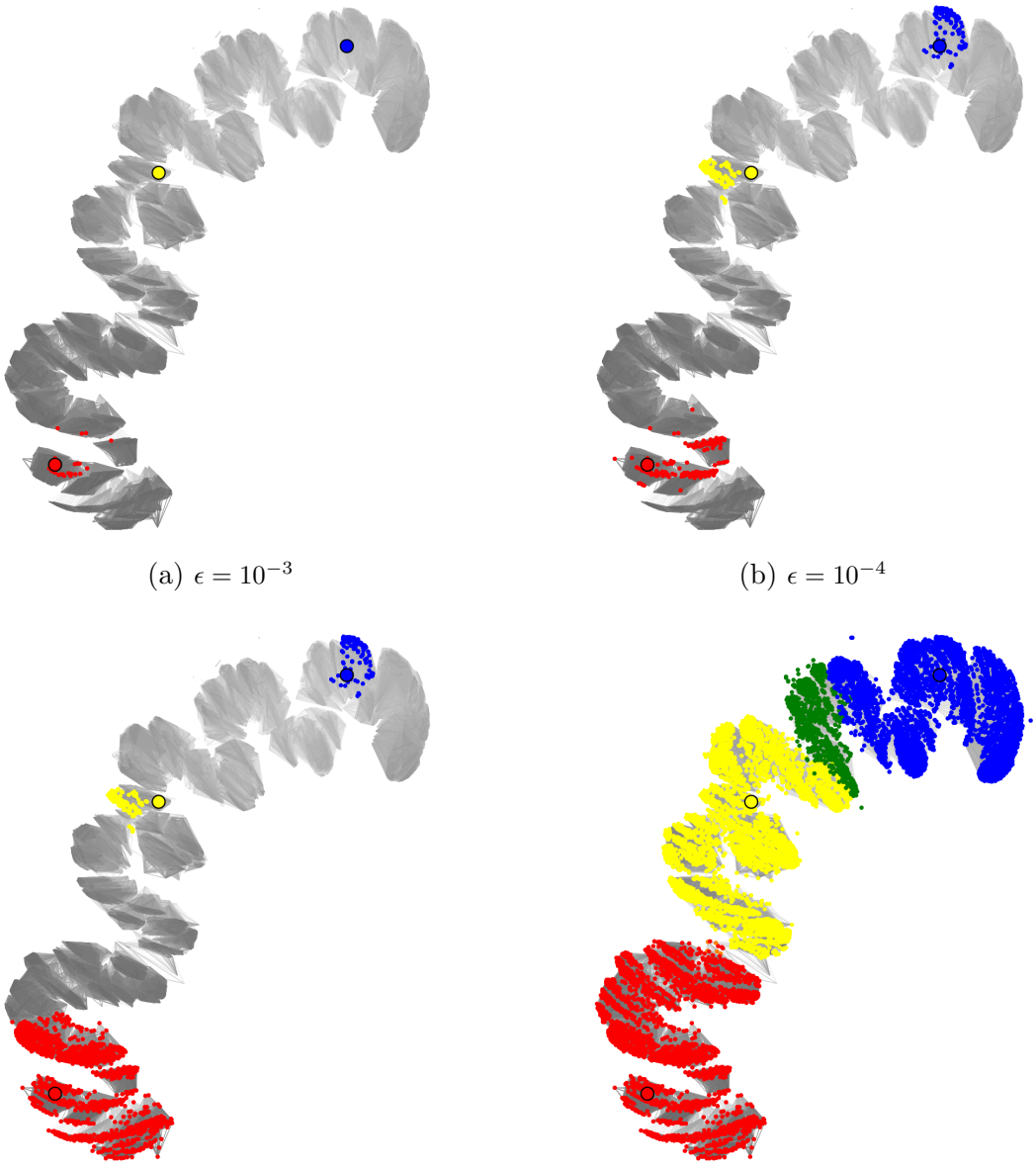
Figure 3.17: Visualisation of global structure in FB-Johns55. We construct the network layout by weighting each edge using the corresponding entry of the association matrix for the AclCut method (see Fig. 3.15b) and then applying the same procedure as in Fig. 3.16. Coloured nodes correspond to local communities for three different seed nodes. Note the difference in behaviour for the red community versus the blue and yellow communities. The blue and yellow communities gradually spread as we decrease ϵ, and they eventually merge to cover a large part of the network. However, the red community initially spreads more quickly than the other two communities as we decrease ϵ but then remains localised as we decrease ϵ further.

remains separate from the other communities.

For US-Senate (see Fig. 3.18 as well as Fig. 3.15), we clearly observe the signature of temporal-based community structure at a large size scale. See Figs. 3.15c, 3.15f, and 3.15i. Using AclCut and MovCut, we also obtain partitions at the scale of individual Congresses (see the insets in Figs. 3.15cf), which sometimes split into two or occasionally three individual communities. These latter partitions have been discussed previously in terms of polarisation between political parties [46, 149, 207]. Because we fixed the temporal order of Congresses for US-Senate and only sort Senators within the same Congress, this visualisation reveals communities within each Senate as well as more temporally-disparate communities. In particular, for the EgoNet method, this ordering introduces a chequerboard pattern that arises from a set of temporal communities that include similar subsets of the senators at different times. (Senators in one such community tend to belong to the same political party, and we observe a chequerboard pattern because there are two parties.) Figure 3.18 clearly shows that this temporal structure also dominates the behaviour of local communities for individual seed nodes.

An important point from these visualisations is that, for both CA-GrQc and FB-Johns55, the meso-scale and large-scale structures that result from the superposition of local communities does *not* correspond particularly well to intuitive good-conductance communities. Relatedly, it also does *not* correspond particularly well to an intuitive low-dimensional structure or a nearly decomposable block-diagonal matrix of community assignments (see our illustration in Fig. 3.1a), one or both of which are often assumed (typically implicitly) by many global methods for algorithmically detecting communities in networks [61, 73, 141, 142, 167]. Of the networks that we investigate, only the temporal structure in US-Senate (as well as in US-House, which is a related temporally-dominant network) closely resembles such an idealisation. This is reflected clearly in its downward-sloping NCP (see Figs. 3.6a, 3.8a, and 3.10a) and in the visualisations in Fig. 3.15.

Instead, in the other (e.g., collaboration, Facebook, and many many other empirical [127, 128]) networks, community structure as a function of size is much more subtle



(a) $\epsilon = 10^{-3}$ (b) $\epsilon = 10^{-4}$

(c) $\epsilon = 10^{-5}$ (d) $\epsilon = 10^{-6}$

Figure 3.18: Visualisation of global structure in US-Senate. We construct the network layout by reweighting each edge using the corresponding entry of the association matrix for the AclCut method (Fig. 3.15c) and then applying the same procedure as in Figs. 3.16 and 3.17. Coloured nodes correspond to local communities for three different seed nodes. The spreading behaviour of the different local communities largely follows the temporal structure of the network.

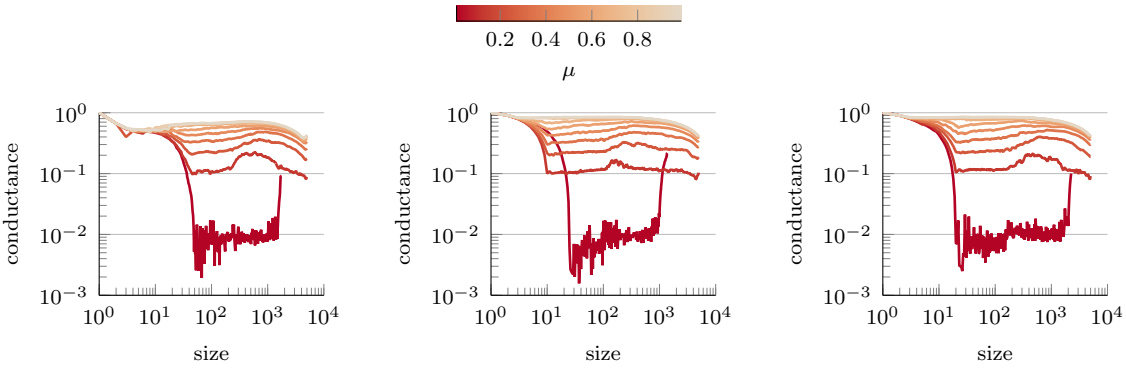
and complicated. Fortunately, our locally-biased perspective provides one means to try to resolve such intricacy. By averaging over results from different seed nodes, a local approach like ours leads naturally to the presence of strongly overlapping communities. Overlapping community structure has now been studied for several years [3, 12, 168], and recent observations continue to shed new light on the ubiquity of community overlap [211]. Overlap of communities in networks is a pervasive phenomenon [123, 211]. Our expectation is that most large realistic networks have communities with significant overlap, rather than merely a small amount of overlap that would amount to a small perturbation of the idealised, nearly decomposable communities in Fig. 3.1a. Additionally, such overlaps imply that larger communities tend to have lower quality in terms of their internal versus external connectivity (i.e., in terms of how much they resemble the intuitive communities that many researchers know and love) than smaller communities—in agreement with our empirical results on both the collaboration networks and Facebook networks, but in strong disagreement with popular intuition. In these latter cases, recent work that fits related networks with upward-sloping NCPs to hierarchical Kronecker graphs resulted in parameters that are consistent with the core-periphery structure that we illustrated in Fig. 3.1b [126].

3.5 Empirical Results on Synthetic Benchmarks

Motivated by our empirical results on networks constructed from real data, we also apply our methods to LFR networks to test the extent to which they are able to reproduce the three classes of NCP behaviour (upward-sloping, flat, and downward-sloping) that we have observed with real networks.

Recall from Section 2.5.4 that one specifies the strength of the planted community structure of an LFR network using a mixing parameter μ, where each node shares a fraction $1 - \mu$ of its edges with nodes in the same community. A simple calculation shows that this definition of the mixing parameter implies that each community in the planted partition has conductance μ (up to rounding effects).

In Fig. 3.19, we show representative NCPs for LFR networks for three choices



(a) AclCut, $\langle k \rangle = 10$, $k_{\max} = 100$, $\tau_1 = -2$, $\tau_2 = -3$, $c_{\min} = 10$, $c_{\max} = 50$

(b) AclCut, $\langle k \rangle = 20$, $k_{\max} = 50$, $\tau_1 = -2$, $\tau_2 = -1$, $c_{\min} = 10$, $c_{\max} = 50$

(c) AclCut, $\langle k \rangle = 20$, $k_{\max} = 50$, $\tau_1 = -2$, $\tau_2 = -1$, $c_{\min} = 20$, $c_{\max} = 100$

Figure 3.19: NCPs of LFR synthetic benchmark networks [118] with $n = 10000$ nodes. Colours correspond to different values of the mixing parameter μ. Our choices for the mean degree $\langle k \rangle$, maximum degree $k_{\max}$, exponent of the degree distribution τ_1, exponent of the community size distribution τ_2, minimum community size $c_{\min}$, and maximum community size $c_{\max}$ correspond to the ones used in Refs. [115, 117] to benchmark community-detection algorithms.

of parameters for the degree distribution and community-size distribution that have been used previously to benchmark community-detection algorithms [115, 117, 118]. (The results presented in Fig. 3.19 use the AclCut method to sample communities, but we obtain nearly identical NCPs using the MovCut method.) The three panels demonstrate that all three parameter choices yield networks with similar NCPs. In particular, we observe that—above a certain critical size—the best communities have comparable quality, as a function of increasing size. That is, above the critical size, the NCP is approximately flat. For $\mu = 0.01$, which is the smallest value of the mixing parameter we show in Fig. 3.19, the networks are starting to fragment into disconnected components. Increasing the topological mixing parameter μ in the LFR network generative mechanism at first shifts the entire NCP upwards because the number of inter-community edges increases. For $\mu \approx 1$, it levels off to the characteristic flat shape for an NCP of a network generated from the configuration model of random graphs. This behaviour should not come as a surprise as it is a direct consequence of the uniformly-random inter-community edges.

Importantly, the behaviour for the LFR benchmark networks from Ref. [118]

that we illustrate in Fig. 3.19 does *not* resemble the NCPs for any of the real-world networks in either the present paper or in Ref. [127, 128]. In addition, we have been unable to find parameter values for which the qualitative properties of realistic NCPs—in particular, a relatively gradually upward-sloping NCP—are reproduced, which suggests that the community structure generated by the LFR benchmarks is *not* realistic in terms of its size-resolved properties.

To verify that this behaviour is not an artifact of the particular choices of parameters shown in Fig. 3.19, we sampled sets of parameters uniformly at random with $n \in \{1000, 10\,000, 50\,000\}$, $\tau_1, \tau_2 \in \{-1, -2, \ldots, -5\}$, $\langle k \rangle \in \{10, 11, \ldots, 100\}$, $k_{\max} \in \{\langle k \rangle, \langle k \rangle + 1, \ldots, 250\}$, $c_{\min} \in \{10, 11, \ldots, 250\}$, and $c_{\max} \in \{\max(c_{\min}, k_{\max}), \ldots, 250\}$. The aggregate trends of the NCPs for the LFR benchmark networks with the different parameters we sample are similar to and consistent with the results shown in Fig. 3.19. Hence, although the LFR benchmark networks are useful as tests for community-detection techniques, our calculations suggest that they are unable to reproduce a fundamental feature of many real networks with respect to variation in community quality (and, in particular, worsening community quality) as a function of increasing community size.

Based on our empirical observations, our locally-biased perspective on community detection suggests a natural approach to determine whether synthetic benchmarks possess small-scale, medium-scale, and large-scale community structure that resembles that of large realistic networks: namely, a family of synthetic benchmark networks ought to include parameter values that generate networks with (robust) upward-sloping, flat, and downward-sloping NCPs (as observed in Figs. 3.2a and 3.6a).

3.6 A Local Perspective on Communities in Multilayer Networks

The perspective of communities as bottlenecks to dynamical processes on networks, extends in a natural way to multilayer networks. In particular, the two US-Congress

networks we analyse in Section 3.4 are examples of flattened multilayer networks. Recall from Section 2.3 that the classical random walk on a multilayer network is equivalent to the usual random walk on the flattened network. Thus, we have already analysed local community structure in a multilayer network using a classical random walk. However, recall from Section 2.3 that the classical random walk is not the only way one can generalise random-walk-like dynamics to multilayer networks. In this section we illustrate how the methodology generalises to an arbitrary discrete-time, ergodic Markov chain on a multilayer network. Such a dynamical process follows the update equation

$$\boldsymbol{p}_{i\boldsymbol{a}}(t+1) = \sum_{j\boldsymbol{b}} \boldsymbol{P}_{i\boldsymbol{a}}^{j\boldsymbol{b}} \boldsymbol{p}_{j\boldsymbol{b}}(t)\,, \tag{3.14}$$

where $\boldsymbol{p}_{j\boldsymbol{b}}(t)$ is the probability for a random walker to be at node j in layer $\boldsymbol{b}$ at time t and $\boldsymbol{P}_{i\boldsymbol{a}}^{j\boldsymbol{b}}$ is the probability for a random walker to transition from node j in layer $\boldsymbol{b}$ to node i in layer $\boldsymbol{a}$ in one time step. We also assume that the random walk is ergodic, so that it has a well-defined stationary distribution $\boldsymbol{p}_{i\boldsymbol{a}}(\infty)$. We then use the stationary distribution to define the *conductance* [93] of a set S of nodes such that $\sum_{i\boldsymbol{a}\in S} \boldsymbol{p}_{i\boldsymbol{a}}(\infty) \leq 1/2$ as

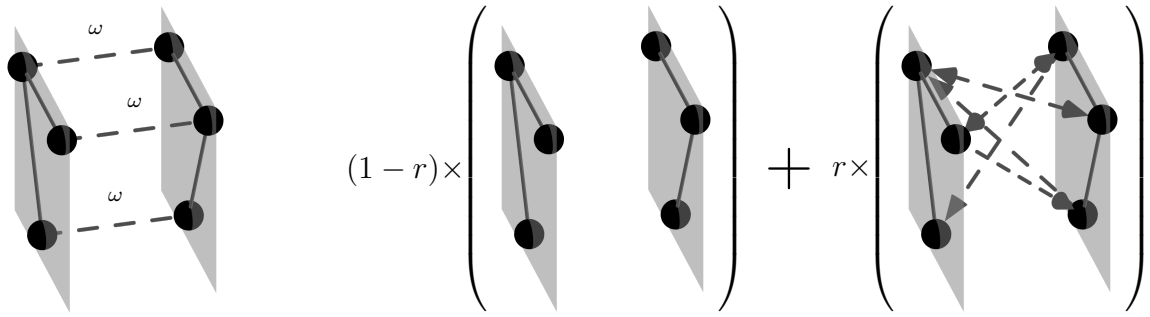
$$\phi(S) = \frac{\displaystyle\sum_{i\boldsymbol{a}\in S} \sum_{j\boldsymbol{b}\notin S} \boldsymbol{P}_{j\boldsymbol{b}}^{i\boldsymbol{a}} \boldsymbol{p}_{i\boldsymbol{a}}(\infty)}{\displaystyle\sum_{i\boldsymbol{a}\in S} \boldsymbol{p}_{i\boldsymbol{a}}(\infty)}\,, \tag{3.15}$$

which we use as a quality measure for local communities. Note that the definition of conductance in Eq. (3.15) is equivalent to the previous definition of conductance in Eq. (3.3) in the case of an unbiased random walk on an undirected network.

Having selected an appropriate random walk, we can define the associated personalised PageRank (pr) of state node $i\boldsymbol{a}$ as the solution to the equation

$$\mathrm{pr}(\alpha, \boldsymbol{s})_{i\boldsymbol{a}} = \alpha \sum_{j\boldsymbol{b}} \boldsymbol{P}_{i\boldsymbol{a}}^{j\boldsymbol{b}} \mathrm{pr}(\alpha, \boldsymbol{s})_{j\boldsymbol{b}} + (1-\alpha)\boldsymbol{s}_{i\boldsymbol{a}}\,, \tag{3.16}$$

where $\boldsymbol{s}$ is a probability distribution over state nodes that determines the seed nodes for the method. Recall from Section 2.3 that if we define the adjacency tensor of the



(a) Classical random walk (b) Relaxed random walk

Figure 3.20: Illustration of two types of random walk on a multilayer network. The walks differ in the way that random walkers switch between layers. (a) *Classical random walk* [50, 149], in which we introduce interlayer edges with weight ω between state nodes (i.e., node-layer tuples) that represent the same physical node in adjacent layers. (b) *Relaxed random walk* [48], in which a random walker is constrained at each step to remain within the same layer with probability $(1-r)$ and is free to choose any edge attached to the same physical node in adjacent layers with probability r. (If changing layers, the walker chooses uniformly at random from the set of all neighbours across all layers.)

multilayer network as

$$\tilde{\boldsymbol{A}}_{\boldsymbol{ia}}^{\boldsymbol{jb}} = \boldsymbol{P}_{\boldsymbol{ia}}^{\boldsymbol{jb}} \boldsymbol{p}_{\boldsymbol{jb}}(\infty), \tag{3.17}$$

then $\boldsymbol{P}$ is the transition tensor of the classical random walk on the multilayer network with adjacency tensor $\tilde{\boldsymbol{A}}$. This means that we can use any of the methods from Section 3.3 (as they can all be applied to directed networks) to estimate an NCP based on an arbitrary ergodic random walk on a multilayer network by simply flattening the adjacency tensor $\tilde{\boldsymbol{A}}$. For non-ergodic random walks, we replace $\boldsymbol{p}(\infty)$ by $\mathrm{pr}(\alpha, \boldsymbol{s})$ in Eq. (3.15) and Eq. (3.17), which corresponds to using unrecorded teleportations [112] to compute the conductance. In the next section, we illustrate some features that one may encounter as a consequence of the particular structure of multiplex networks. Our approach is very general, and we stress that one can similarly extend other dynamical processes on networks [68] to a multilayer setting.

3.6.1 Empirical Results

We illustrate the different behaviour one can expect as a result of different types of random walk on a multilayer network using two different random walks. The first

	nodes	edges	layers
European Airline Network [31]	450 airports	3558 (undirected, unweighted)	37 airlines
Lazega Law Firm Network [121, 122, 190]	71 employees	2223 (directed, unweighted)	3 (advice, friendship, co-work)

Table 3.5: Example network data sets

definition of a random walk we consider is the classical random walk which was used to motivate the definition of multilayer modularity (see [149] and the discussion in Section 2.5.1). The second random walk we consider is the "relaxed random walk", which was introduced in [48] to define multilayer InfoMap. For a multilayer network with adjacency tensor $\boldsymbol{A}$ (where we assume that only the intralayer edges are available and hence $a \neq b \Rightarrow \boldsymbol{A}_{ia}^{jb} = 0$), the transition tensor $\boldsymbol{P}(r)$ for the *relaxed random walk* with *relax rate* r is

$$\boldsymbol{P}_{ia}^{jb}(r) = (1-r)\delta(a,b)\frac{\boldsymbol{A}_{ia}^{ja}}{k_a^{ja}} + r\frac{\boldsymbol{A}_{ia}^{ja}}{k^j},$$

where $k_a^{jb} = \sum_i \boldsymbol{A}_{ia}^{jb}$ and $k^j = \sum_{i,a,b} \boldsymbol{A}_{ia}^{jb}$. We illustrate the effective multilayer networks for the two random walks in Fig. 3.20. Note that the effective multilayer network for the relaxed random walk is in general directed, even if the individual layers of the multilayer network are undirected.

We use two small multiplex networks with some intuitive structural properties to illustrate the difference in behaviour between the two types of random walk. The first network is the European Airline Network of [31], where the edges indicate the presence of a direct flight between two European airports and are grouped into layers based on which airline serves the connection. The second network is the Lazega Law Firm Network [121, 122, 190] where edges indicate different types of connections between employees. We highlight some key properties of these two networks in Table 3.5.

In Figs. 3.21 and 3.22 we show the result of using the AclCut method to estimate NCPs for the two networks based on the classical random walk for different values of the interlayer edge weight ω and the relaxed random walk for different values of the relax rate r. For the Airline Network, both random walks are ergodic (even

though the relaxed random walk is directed) and we use their stationary distribution to define conductance. However, the relaxed random walk is not ergodic for the Lazega Law Firm Network and we have to introduce teleportations. As suggested in [48, 112], we use unrecorded link teleportation to reduce the teleportation bias, i.e., we use $\mathrm{pr}(\alpha, \boldsymbol{s})$ with

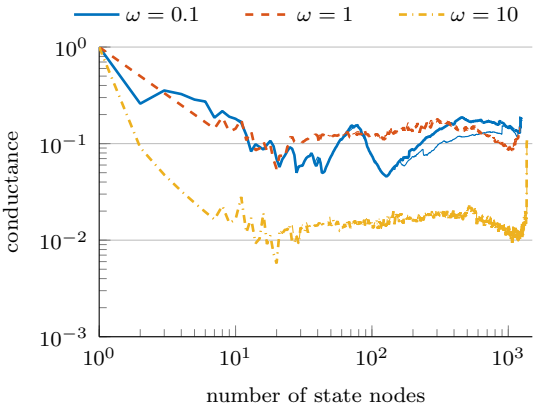
$$\boldsymbol{s}_{ia} = \frac{k^a_{ia}}{\sum_{i,a} k^a_{ia}}, \quad k^a_{ia} = \sum_j \boldsymbol{A}^{ja}_{ia},$$

to replace the stationary distribution. The results in Fig. 3.22 are for the choice of $\alpha = 0.95$, however the NCPs are reasonably robust to changing α. We use this teleportation procedure for both random walks (even though the classical random walk is ergodic on this network) to make results more comparable.

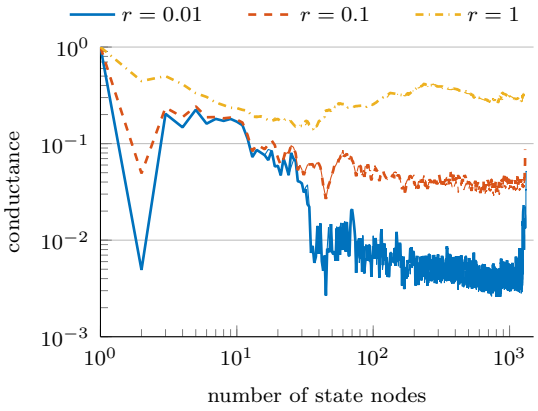
An interesting aspect of multilayer networks is that the physical nodes are a natural way of defining seed sets with more than one (state) node. We show the results of sampling NCPs using either physical or state nodes as seeds in Figs. 3.21 and 3.22. At least in these examples, using physical nodes or state nodes as seeds results in similar NCPs. For the Airline Network (see Fig. 3.21), sampling using physical nodes results in communities of comparable or slightly better quality than using state nodes. However, the situation is reversed for the Lazega Law Firm Network, where sampling using state nodes tends to result in communities of slightly better quality. For both networks the difference between the sampling methods essentially disappears when the coupling between layers is sufficiently strong (i.e., $\omega \to \infty$ and $r \to 1$).

For weak interlayer coupling (i.e, $\omega \to 0$ and $r \to 0$), the layer structure of the multilayer network results in bottlenecks for both the classical and relaxed random walk. In this regime, the notable communities one identifies are groups of layers with similar structure (e.g., groups of similar airlines in Figs. 3.21cd) and one can identify communities that are localised within a single layer (see Figs. 3.22cd).

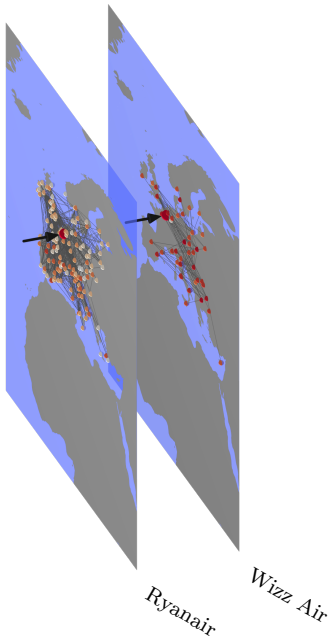
For strong interlayer coupling, the behaviour of the two types of random walk is markedly different. This difference may be best understood if we consider the behaviour of the two random walks at the level of the physical nodes. In the limit as



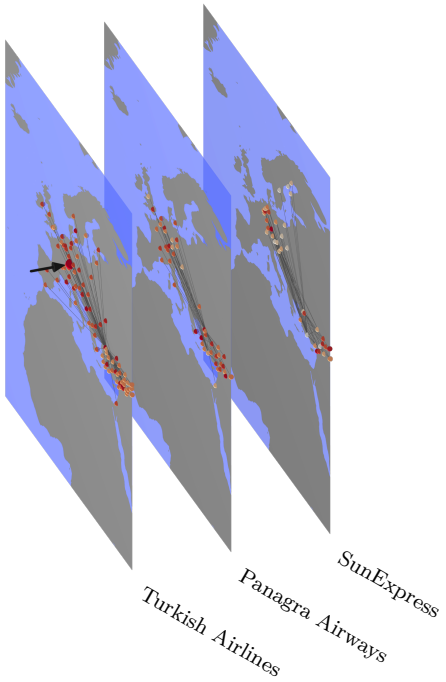
(a) Classical random walk



(b) Relaxed random walk

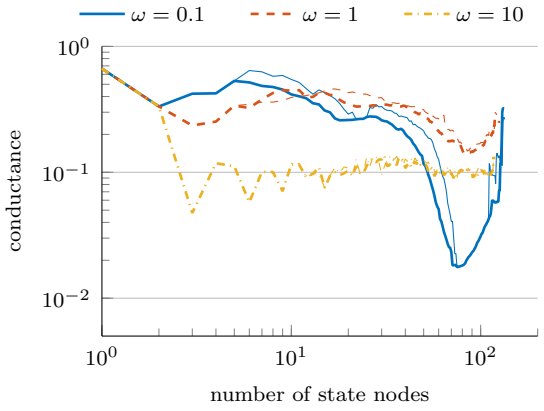


(c) Best community with 173 state nodes for $\omega = 0.1$ (physical node as seed)

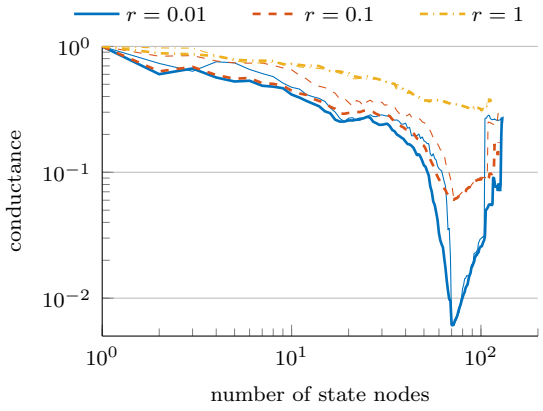


(d) Best community with 169 state nodes for $r = 0.1$ (state node as seed)

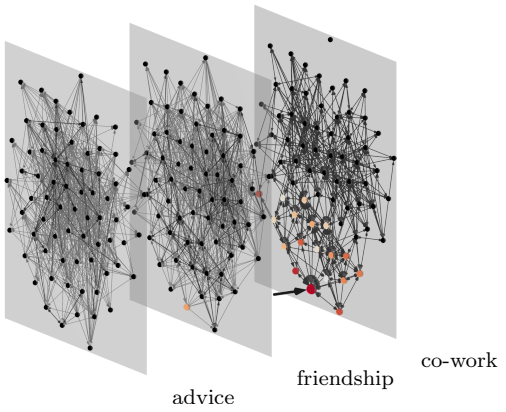
Figure 3.21: *European Airline Network*: Multiplex transportation network with 37 layers, each representing connections of a single airline [31]. Panels (a) and (b) show NCPs for (a) the classical random walk and (b) the relaxed random walk for different strength of the interlayer coupling parameters ω and r. Thin curves indicate communities sampled using physical nodes instead of state nodes as seeds. Notably, we find slightly better large (larger than 100 state nodes) communities by using physical nodes as seeds for the classical walk with $\omega = 0.1$. Panels (c) and (d) illustrate some of the communities that we obtain. We shade the state nodes in a community from dark red to light grey based on their rank within the community. The large arrows point at the seed nodes.



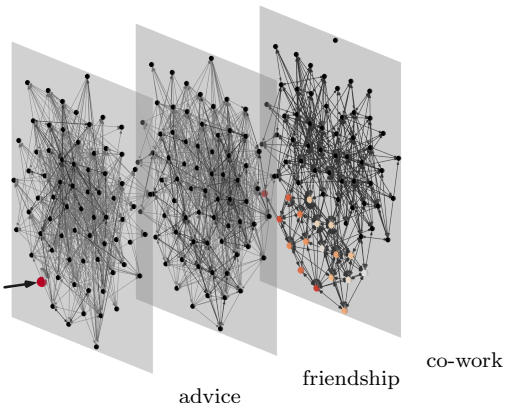
(a) Classical random walk



(b) Relaxed random walk



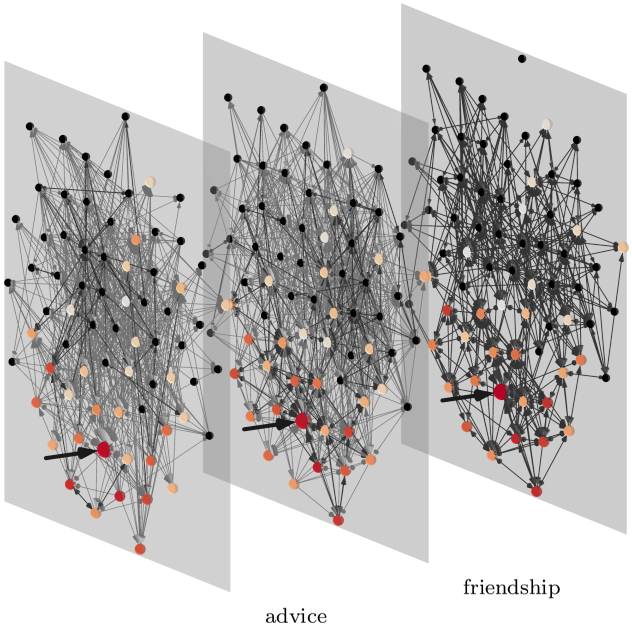
(c) Best community with 19 state nodes for $\omega = 0.1$ (state node as seed)



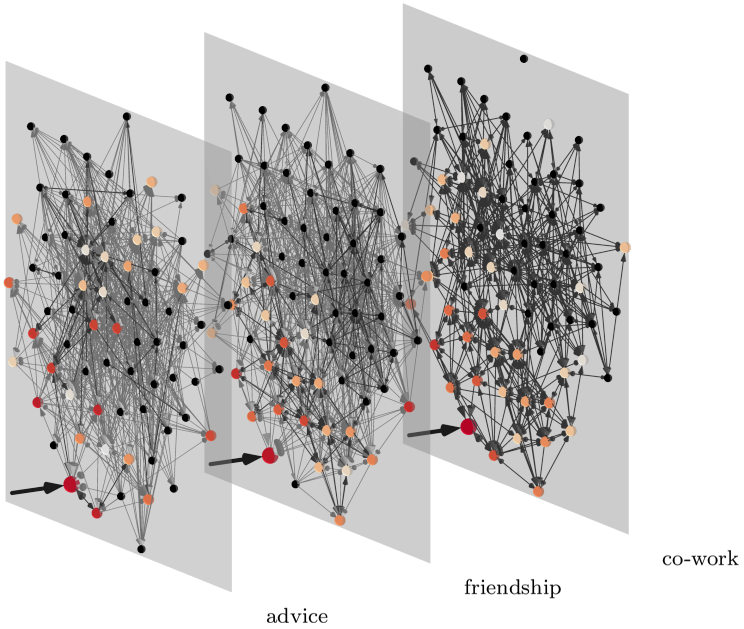
(d) Best community with 19 state nodes for $r = 0.01$ (state node as seed)

Figure 3.22: *Lazega Law Firm Network*: Multiplex social network with three layers representing advice, friendship, and co-work relationships between partners and associates of a corporate law firm [121, 122, 190]. Panels (a) and (b) show associated network community profiles. Using small layer-jumping probability r in the relaxed random walk and small interlayer edge weight ω in the classical random walk yields similar results using the two different dynamics. In each case, the communities tend to be localised to a single layer. The prominent dip in the network community profiles around 71 nodes is the result of a community that contains all state nodes in the "co-work" layer. The communities that we highlight in Panels (c) and (d) are responsible for the less pronounced dips in the network community profiles around 19 nodes. They contain the members of the firm who are based in the Hartford office.

$r \to 1$, the dynamical process induced on the physical nodes by the relaxed random walk is equivalent to an unbiased random walk on the aggregate network. Thus the NCP for the relaxed random walk with $r \approx 1$ tends to resemble the NCP for the aggregate network. The process induced by the classical random walk however



(e) Best community with 91 state nodes for $\omega = 10$ (physical node as seed)



(f) Best community with 91 state nodes for $r = 1$ (physical node as seed)

Figure 3.22 (Cont.): For large w, the classical random walk yields communities that are largely coherent across layers (see Panel (e)), in the sense that if a state node is a member of a community, the other state nodes representing the same physical node also tend to be in the community. Communities from the relaxed random walk (see Panel (f)) tend to be less coherent across layers compared to those from the classical walk.

approaches a random walk on the aggregate network with increasingly strong self-loops at the physical nodes as $\omega \to \infty$. As a result, the classical random walk explores the network more and more slowly as we increase ω and the NCP for the classical random walk thus shifts towards zero as we increase ω (once ω is sufficiently large for this effect to dominate). The two processes also behave differently at the level of the state nodes. For strong interlayer coupling, the classical random walk explores layers uniformly, i.e, the probability for a random walker to visit a particular state node is roughly the same for all state nodes that represent the same physical node. The relaxed random walk does not have this property and in general different state nodes that represent the same physical node will be visited with different probabilities, even when $r = 1$.

For the Airline Network, the best communities we identify using the classical random walk with strong interlayer coupling tend to be geographically localised whereas for the relaxed random walk, the best communities tend to be groups of similar airlines, irrespective of the value of r. For the Lazega Law Firm Network, the difference in behaviour between the two random walks is less pronounced, though we do see that the classical random walk explores the layers in a more uniform manner (see Figs. 3.22ef).

Note that the NCPs in Figs. 3.21ab are roughly flat for communities larger than about 30 nodes irrespective of the interlayer coupling strength. This indicates neither the layer structure nor the aggregate network of the Airline Network is well-embedded in a low-dimensional space. This is in stark contrast with ground-based infrastructure and transportation networks [128] which embed well in a two dimensional space and the US-Congress networks we discuss in Section 3.4 where the interlayer structure is embedded in time. The Lazega Law Firm Network also does not embed in a low-dimensional space, as indicated by the flat NCPs for strong interlayer coupling in Figs. 3.22ab. However, the small number of layers in this network results in a downward sloping trend of the NCPs for weak interlayer coupling.

3.7 Summary

In this chapter we have investigated meso-scale structure in networks from a local perspective based on the behaviour of diffusion-like dynamics on a network. The main assumption behind this work is that meaningful community structure should manifest itself as bottlenecks to dynamical processes on a network, where we have used random walks and geodesic spreading as proxies for information diffusion on real networks. This assumption is the basis for many popular community detection methods (see Section 2.5). By analysing bottlenecks to dynamical processes for different initial conditions and at different size scales, we have shown that community structure in real networks is much more intricate than is suggested by the block-diagonal assumption that is (either implicitly or explicitly) made by most community-detection methods and when developing benchmark networks to test those methods. Community structure in real networks interplays with other meso-scale features, such as core-periphery structure [44, 174, 211], and investigating only community structure without consideration of other structures can lead to misleading results.

Although most algorithmic methods for community detection take a different approach from ours, the observation that network community structure depends not only on network structure per se but also on the dynamical processes that take place on a network and the initial conditions (i.e., seed node or nodes) for those processes is rather traditional in many ways. Recall, for example, Granovetters observation that a node with many weak ties is ideally suited to initialise a successful social contagion process [83]. Our perspective also meshes better than global ones with real-life experience in our own networks. Both of these observations underscore our point that whether particular network structures form bottlenecks for a dynamical process depends not only on the process itself but also on the initial conditions of that process. Furthermore, a local perspective on community detection has the key advantage that it allows pervasive community overlap in a natural way.

The perspective of communities as bottlenecks to dynamics also extends in a natural way to multilayer networks. One can generalise the unbiased random walk

to multilayer networks in many different ways that are all consistent with the single-layer case. These different definitions of a random walk lead to markedly different results when analysing communities in multilayer networks from a local perspective. This highlights that there is even more potential for disagreement between different methods about the kind of structure that should be considered a good community in a multilayer network than in the already ill-defined case of single-layer networks.

Chapter 4

Multilayer Benchmark Networks for Community Detection

Benchmark networks with known structural properties are an important tool for analysing and comparing the performance of different algorithms. The ill-defined nature of the community detection problem makes good benchmark networks particularly crucial. While most would agree that a good community should be a group of nodes that are "surprisingly well-connected", what we mean by "surprising" and "well-connected" will depend on the application at hand, and different algorithms were often developed with different definitions in mind. Further complications arise from the fact that the associated community assignment problem for many notions of community quality is NP-hard [61]. Together with the need for methods to scale to be able to analyse large networks, this means that we often need to use heuristics which only find approximate solutions where we often have little theoretical understanding of how closely they resemble the optimal solution. By comparing the performance of different algorithms on benchmark networks, we can gauge which algorithms are more appropriate in certain situations.

For single-layer networks good benchmark networks exist (see Section 2.5.4), though as we see in Section 3.5 there is still room for improvement. In many applications, single-layer networks are not sufficient to capture the intricacies of connectivity patterns between the entities of interest. Examples include temporal and multiplex

networks, which we represent as multilayer networks (see Section 2.2), where one expects that different layers of a network are correlated in some way. For example, the connectivity patterns in one's Facebook friendship network today are not independent of the connectivity patterns in one's Facebook network next year (temporal). Or, the connectivity patterns in one's Facebook network today are not independent of the connectivity patterns in one's Twitter network today (multiplex). Data sets that have this type of structure are increasingly available, and some methods to detect communities in multilayer networks have been developed [48, 149]. These methods aim to exploit the correlations between layers to identify structure that goes beyond what one can identify by investigating the layers of a multilayer network individually.

Existing benchmark networks for multilayer community detection either assume a temporal structure [81, 183] or a simplified multiplex structure with independent blocks of layers where network layers in the same block have identical community structure [48]. In this chapter we propose a method for sampling random multilayer community structures with arbitrary interlayer correlation structure, that incorporates the temporal structure in [183] and simplified multiplex structure in [48] as special cases. One can then use existing planted partition models to generate networks for each layer that have the desired community structure or one could modify the network generation process to introduce additional correlation between layer networks beyond that induced by community structure. We only use LFR-like networks in this chapter, but the modular nature of this approach easily allows us to generalise other types of benchmarks to multilayer networks.

4.1 Sampling Community Structure

For our benchmark networks we impose the coupling edges by specifying the coupling adjacency tensor $\boldsymbol{A}_C$ of the multilayer network, which determines the correlations between community structure in different layers. We then generate the benchmark networks by proceeding in two steps. First, we sample a multilayer community structure that satisfies the desired correlation properties. Second, we sample random

intralayer networks that have this prescribed community structure. In the following sections we describe this process in detail.

4.1.1 General Multilayer Networks

We assume that the correlations between community structure in different layers are the result of a copying process on the state nodes that is governed by the coupling edges. We assume that each state node has a unique community membership $\boldsymbol{s}_{i\boldsymbol{a}} \in \mathbb{N}$ (note that we could use this framework to represent overlapping communities by identifying multiple state nodes in a single layer with the same physical node), where two state nodes are in the same community if $\boldsymbol{s}_{i\boldsymbol{a}} = \boldsymbol{s}_{j\boldsymbol{b}}$. We use asynchronous updating for the copying process, i.e., at each time step we pick a state node $i\boldsymbol{a}$ uniformly at random and update its community assignment by copying the community assignment of one of its neighbours in the coupling network (where we pick a particular neighbour $j\boldsymbol{b}$ with probability $\boldsymbol{A}_{C_{i\boldsymbol{a}}^{j\boldsymbol{b}}}$) or by resampling its community assignment from a layer-dependent null-distribution $\mathbb{P}_0^{\boldsymbol{a}}$ with probability $(1-\sum_{j\boldsymbol{b}} \boldsymbol{A}_{C_{i\boldsymbol{a}}^{j\boldsymbol{b}}})$. Here we assume that we have specified the coupling edges such that $\sum_{j\boldsymbol{b}} \boldsymbol{A}_{C_{i\boldsymbol{a}}^{j\boldsymbol{b}}} \leq 1$ for each state node $i\boldsymbol{a}$. We can then write the update equation for state node $i\boldsymbol{a}$ at time t as follows:

$$\begin{aligned}\mathbb{P}\left[\boldsymbol{s}_{i\boldsymbol{a}}(t+1) = s|\boldsymbol{s}(t)\right] = &\left(\sum_{j\boldsymbol{b}} \boldsymbol{A}_{C_{i\boldsymbol{a}}^{j\boldsymbol{b}}} \delta(\boldsymbol{s}_{j\boldsymbol{b}}(t), s)\right) \\ &+ \left(1 - \sum_{j\boldsymbol{b}} \boldsymbol{A}_{C_{i\boldsymbol{a}}^{j\boldsymbol{b}}}\right) \mathbb{P}_0^{\boldsymbol{a}}[s].\end{aligned} \tag{4.1}$$

This setup naturally lends itself to using Gibbs sampling [65–67, 193] to sample multilayer community structure that is consistent with this updating process. Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method that allows us to sample from the joint community assignment distribution for all state nodes (which is the community structure distribution that we want to sample) using only the conditional distributions specified by Eq. (4.1). From an initial multilayer partition $\boldsymbol{s}(0)$, Gibbs sampling proceeds by running the updating process described above, i.e., at

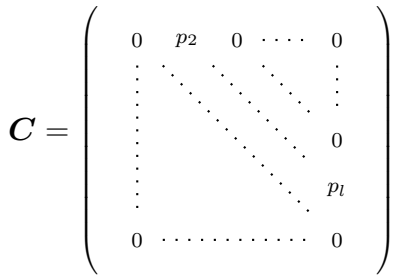
each iteration we pick a state $i\boldsymbol{a}$ uniformly at random and update its community assignment by sampling from the conditional distribution defined in Eq. (4.1). This defines a Markov Chain on the space of all multilayer partitions $\mathbb{S}$ which has our desired joint distribution as its stationary distribution. By running the updating process for sufficiently many iterations (the "burn-in period"), the state of the Markov Chain is approximately sampled from the stationary distribution. We initialise the Markov Chain by sampling from the null-distribution, such that for each state node $\boldsymbol{s}_{i\boldsymbol{a}}(0) \sim \mathbb{P}_0^{\boldsymbol{a}}$. Sampling initial conditions in this way can be important, as the updating process is not necessarily ergodic when $\sum_{j\boldsymbol{b}} \boldsymbol{A}_{C_{i\boldsymbol{a}}^{j\boldsymbol{b}}} = 1$ for some state nodes.

4.1.2 Fully Interconnected, Layer-Coupled Multilayer Networks

In many situations the complexity of allowing arbitrary interlayer edges is unnecessary and even undesirable when designing benchmark networks. A particularly useful restriction which still allows us to represent many situations of interest is to require the multilayer network to be "fully interconnected" and "layer-coupled" [103]. A multilayer network is *fully interconnected* if each node is represented in each layer, and *layer coupled* if we can represent the coupling adjacency tensor $\boldsymbol{A}_C$ using a *layer coupling tensor* $\boldsymbol{C}$, such that $\boldsymbol{A}_{C_{i\boldsymbol{a}}^{j\boldsymbol{b}}} = \delta(i,j)\boldsymbol{C}_{\boldsymbol{a}}^{\boldsymbol{b}}$, i.e., the weight of the interlayer edges only depends on the layers and not the nodes. Thus, for a fully interconnected, layer coupled multilayer network, we can write the update equation (Eq. (4.1)) as

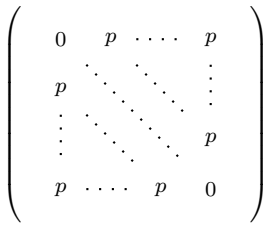
$$\begin{aligned}\mathbb{P}[\boldsymbol{s}_{i\boldsymbol{a}}(t+1) = s|\boldsymbol{s}(t)] = &\left(\sum_{\boldsymbol{b}} \boldsymbol{C}_{\boldsymbol{a}}^{\boldsymbol{b}}\delta(\boldsymbol{s}_{i\boldsymbol{b}}(t), s)\right) \\ &+ \left(1 - \sum_{\boldsymbol{b}} \boldsymbol{C}_{\boldsymbol{a}}^{\boldsymbol{b}}\right)\mathbb{P}_0^{\boldsymbol{a}}(s),\end{aligned} \tag{4.2}$$

which only depends on the layer coupling tensor C and the null-distribution $\mathbb{P}_0$. Note that as before, we require that $\sum_{\boldsymbol{b}} \boldsymbol{C}_{\boldsymbol{a}}^{\boldsymbol{b}} \leq 1$. Each term $\boldsymbol{C}_{\boldsymbol{a}}^{\boldsymbol{b}}$ quantifies the extent to which the community structure in layer $\boldsymbol{a}$ is a direct consequence of the community structure in layer $\boldsymbol{b}$. By changing the structure of $\boldsymbol{C}$, we can generate



$$\boldsymbol{C}_a^b = \delta(a, b+1)p_a, \quad p_a \leq 1$$

(a) Temporal network



$$\boldsymbol{C}_a^b = (1 - \delta(a, b))p, \quad p \leq \frac{1}{l-1}$$

(b) Uniform multiplex network

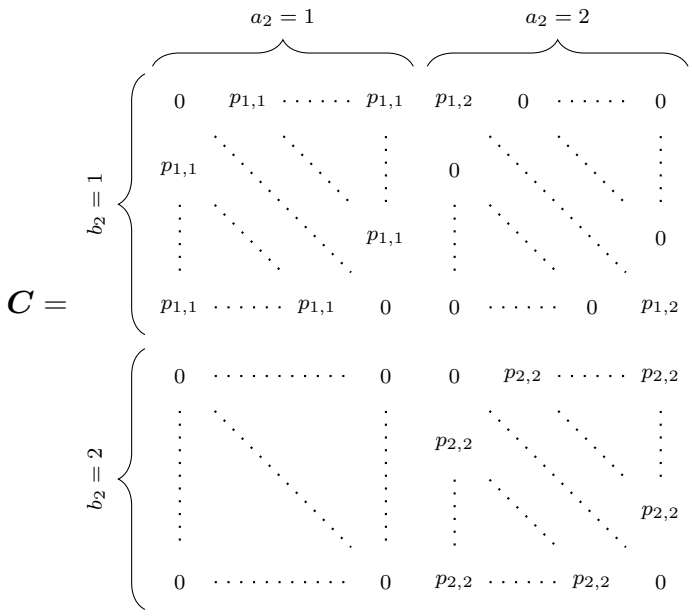
Figure 4.1: Layer coupling tensors (which is a matrix in this case) for different types of multilayer networks with a single aspect. (a) For a temporal network, community structure in a layer only directly depends on community structure in the previous layer, i.e., the only non-zero elements of the layer coupling tensor are in the first superdiagonal. (b) For a multiplex network, community structure in a layer directly depends on community structure in all other layers.

multilayer networks corresponding to different scenarios, e.g., temporal networks, multiplex networks, and multilayer networks with more than one aspect that can have combinations of temporal and multiplex features.

In Fig. 4.1 we show the corresponding coupling tensors for multilayer networks with a single aspect. For a temporal network one usually assumes that the community structure in a layer only directly depends on the community structure in the previous layer, which means that we have $l - 1$ copying probabilities (one for each pair of consecutive layers) that we are free to choose. Typical examples include choosing the same probability for each pair of consecutive layers, giving a uniformly evolving network, or making some of the probabilities significantly smaller than the others, introducing change points that one may want to detect [161].

For a multiplex network, the community structure in a layer can in principle directly depend on the community structure in all other layers, giving us $l(l - 1)$ probabilities to choose. Figure 4.1b illustrates the simplest case, where each layer depends equally on every other layer.

We can also generate multilayer networks with more than one aspect that can combine temporal and multiplex features. We illustrate how to construct an appropriate layer coupling tensor to generate such a network on a simple example with two



$$\boldsymbol{C}^{b_1,b_2}_{a_1,a_2} = (1 - \delta(a_1, b_1))\delta(a_2, b_2)p_{a_1,a_2} + \delta(a_1, b_1)\delta(a_2, b_2 + 1)p_{a_1,a_2}$$

Figure 4.2: Block-matrix representation of a layer coupling tensor for a temporal multiplex network. This is an example of multilayer network with more than one aspect, combining the features of the examples in Fig. 4.1. Community structure in a layer directly depends on the community structure of all other layers in the same time snapshot and on the community structure of the same layer in the previous time snapshot.

aspects, one of which is multiplex and the other temporal, in Fig. 4.2.

4.1.3 LFR-Like Null-Distribution

The community null-distribution in Eqs. (4.1) and (4.2) determines the community structure in the absence of interlayer correlations and is essentially responsible for fixing the number and sizes of the communities in the sampled community structures. We would like to preserve the broad community size distributions of the LFR networks, however the procedure used in [118] of sampling community sizes and then assigning nodes to communities in a way that preserves these community sizes exactly does not mesh well with the copying process we use to induce correlations between community structures in different layers. Instead we modify the approach of [118] to define a null- distribution $\mathbb{P}_0^{\boldsymbol{\alpha}}$ that we can use in Eqs. (4.1) and (4.2).

Independently for each layer, we sample a vector of expected community sizes

Algorithm 3 Sample expected community sizes from power-law distribution with cutoffs.

function SAMPLECOMMUNITYSIZES($\tau_c, c_{\min}, c_{\max}, n$)
  $r = n$
  $i = 1$
  while $r > 0$ **do**
    $\boldsymbol{c}_i$ =POWER($\tau_c, c_{\min}, c_{\max}$) ▷ sample from power law
    if $r - \boldsymbol{c}_i < c_{\min}$ **then** ▷ stop sampling if remainder is too small
      if $r < c_{\max}$ **then**
        $\boldsymbol{c}_i = r$
      else if $r - c_{\min} < c_{\min}$ **then**
        $\boldsymbol{c}_i = r$ ▷ constraints not satisfied
      else
        $\boldsymbol{c}_i$ =POWER($\tau_c, c_{\min}, \min(c_{\max}, r - c_{\min})$)
      end if
    end if
    $r = r - \boldsymbol{c}_i$
    $i = i + 1$
  end while
  $\boldsymbol{c}$ =RANDPERM($\boldsymbol{c}$) ▷ permute community sizes uniformly at random
  $n_c = i - 1$
  return $\boldsymbol{c}, n_c$
end function

$\boldsymbol{c}^{\boldsymbol{a}}$ from a power-law distribution with exponent τ_c, minimum cutoff $c_{\min}$, maximum cutoff $c_{\max}$, and the additional constraint that $\sum_i c_i^{\boldsymbol{a}} = n$ using Algorithm 3, where we use $n_c^{\boldsymbol{a}}$ to denote the number of communities in layer $\boldsymbol{a}$. Using the sampled expected community sizes, we then define the null- distribution such that in the absence of interlayer correlations, the actual community sizes follow a multinomial distribution with the correct expectation, i.e.,

$$\mathbb{P}_0^{\boldsymbol{a}}[s] = \begin{cases} \frac{c_s^{\boldsymbol{a}}}{n}, & 1 \leq s \leq n_c^{\boldsymbol{a}} \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

This procedure keeps community sizes constant as we increase the number of nodes in the network. Together, Eq. (4.2) and Eq. (4.3) provide us with the necessary tools to sample multilayer community structures with the desired properties.

4.2 Sampling Networks

Given a multilayer partition $\mathcal{S}$, we want to sample the intralayer connections $\boldsymbol{A}_L$ in a way that reflects the desired community structure. Here we assume that all correlations between the different layers of the multilayer network are a result of correlations between the community structure in the different layers such that we can generate edges independently for each layer using a single-layer block model.

For the experiments in Section 4.3 we use a model that resembles the LFR benchmark (see Section 2.5.4) for single-layer networks. We quantify the strength of the community structure using a community mixing parameter μ, such that each state node is connected to $\mu k_{i\boldsymbol{a}}^{\boldsymbol{a}}$ state nodes in the same layer that are in a different community, where $k_{i\boldsymbol{a}}^{\boldsymbol{a}} = \sum_j \boldsymbol{A}_{i\boldsymbol{a}}^{j\boldsymbol{a}}$ is the *layer-internal degree* of state node $i\boldsymbol{a}$. We sample the layer-internal degrees for the state nodes from a power law with exponent τ_k, minimum cut-off $k_{\min}$, and maximum cut-off $k_{\max}$ conditional on the community membership of the state node, i.e.,

$$k_{i\boldsymbol{a}}^{\boldsymbol{a}} \sim \textsc{Power}\Big(\tau_k, k_{\min}, \min\big(k_{\max}, (1-\mu)(|S_{\boldsymbol{s}_{i\boldsymbol{a}}}|-1)\big)\Big). \tag{4.4}$$

We then split the stubs for each node into $(1-\mu)k_{i\boldsymbol{a}}^{\boldsymbol{a}}$ community internal and $\mu k_{i\boldsymbol{a}}^{\boldsymbol{a}}$ community external stubs. We connect community internal and community external stubs separately using appropriately constrained configuration models [153].

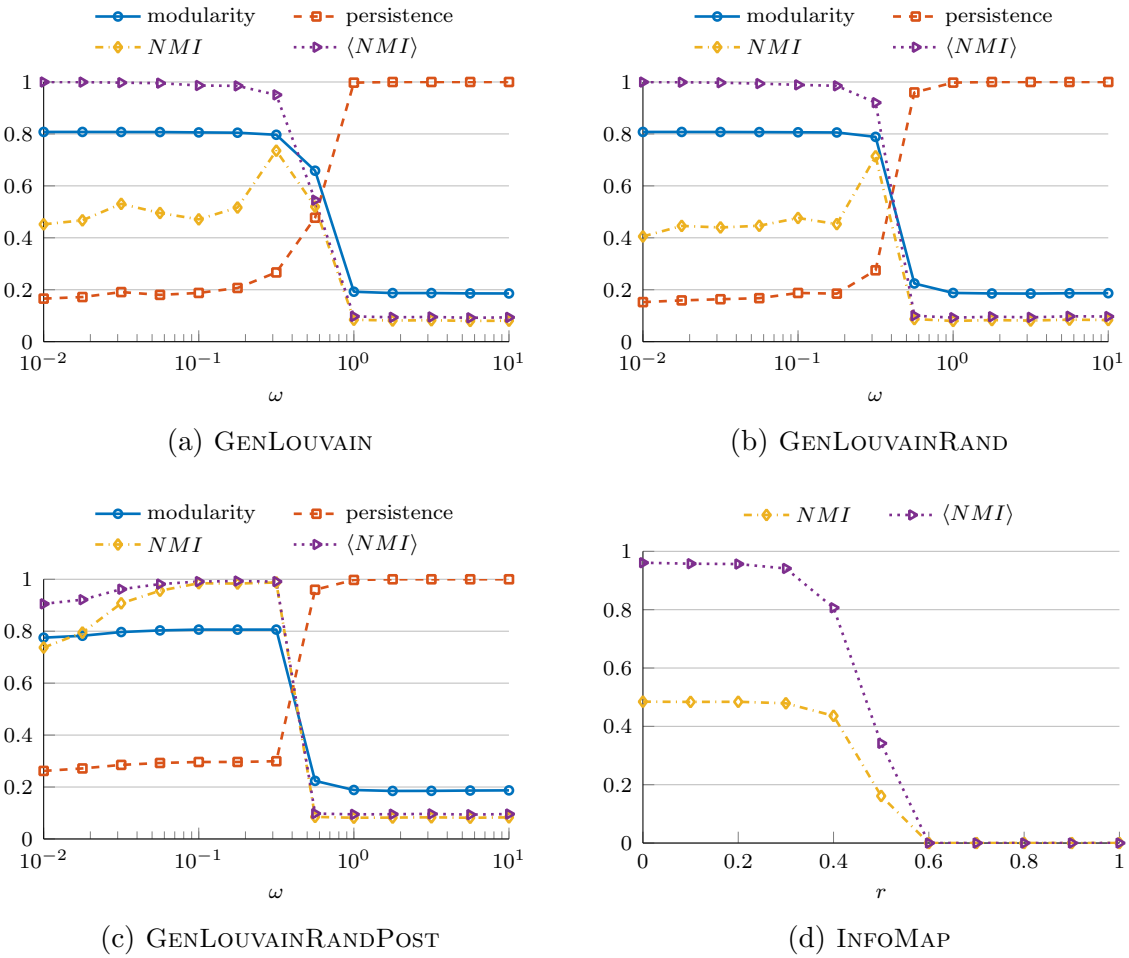
4.3 Numerical Examples

In this section we use the benchmark networks to compare the behaviour of the different variants of the Louvain-like [23] heuristic to optimise multilayer modularity (Eq. (2.35)) that we discuss in Appendix B. We also compare multilayer modularity with multilayer InfoMap[1] [48].

The variants of the Louvain algorithm we consider are based on GenLouvain[2],

[1] using version 0.18.1 of the code available from `http://mapequation.org/code`

[2] code available from `http://netwiki.amath.unc.edu/GenLouvain/GenLouvain`



(a) GenLouvain

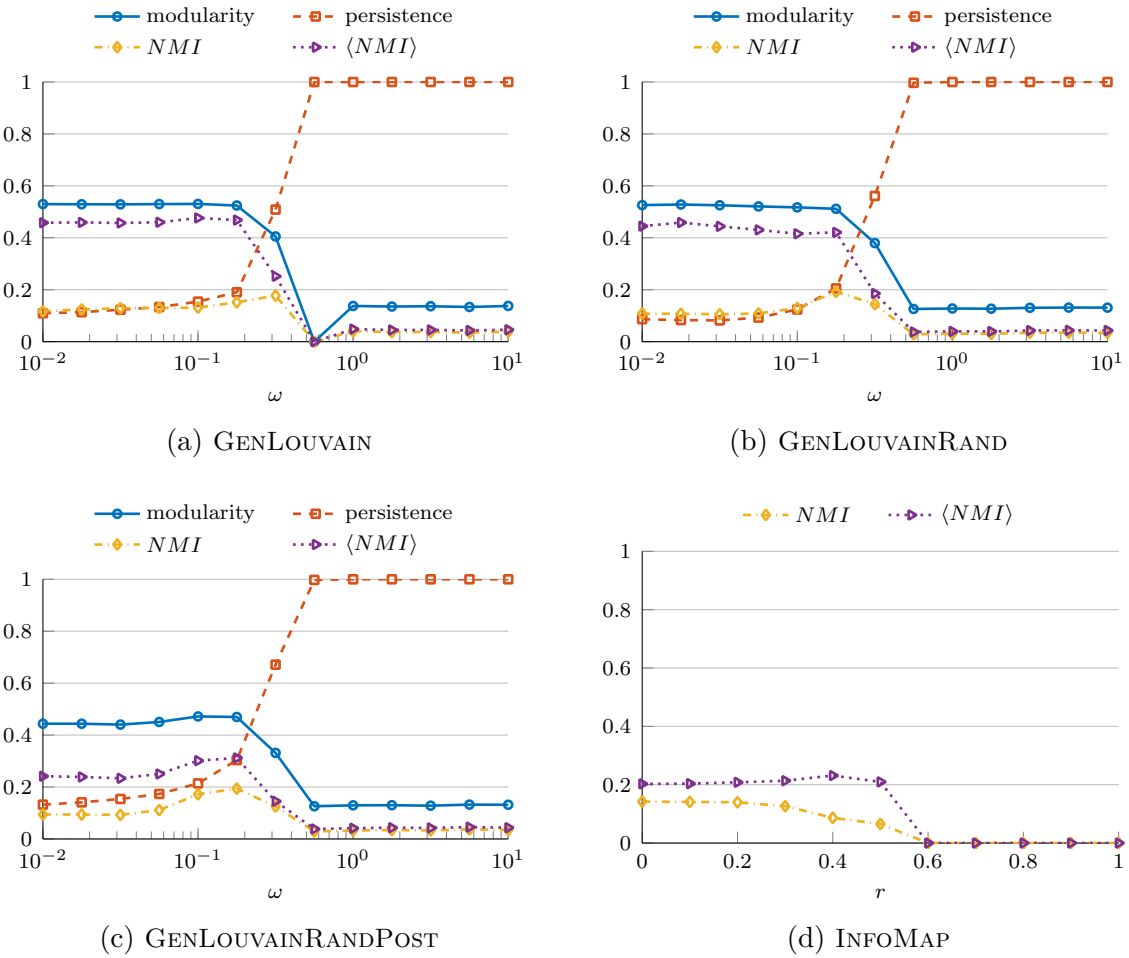
(b) GenLouvainRand

(c) GenLouvainRandPost

(d) InfoMap

Figure 4.3: Benchmark network with $p = 0.08$ (30% probability of re-sampling from the null-distribution) and $\mu = 0.1$

which implements the algorithm of [23] for a general modularity matrix. It is a two phase, locally greedy algorithm. During the first phase, we iterate over all nodes and change their community assignment to maximally increase modularity at each step. Once modularity can no longer be improved by reassigning single nodes, we aggregate nodes that are assigned to the same community into supernodes in the second phase. The algorithm proceeds by iterating the first and second phase with the supernodes until no further improvement of modularity is found. At this point we restart the algorithm with the output partition as the new initial partition and repeat until convergence. The greedy nature of GenLouvain can result in undesirable behaviour in some situations (such as abrupt transitions in behaviour as a function of the interlayer

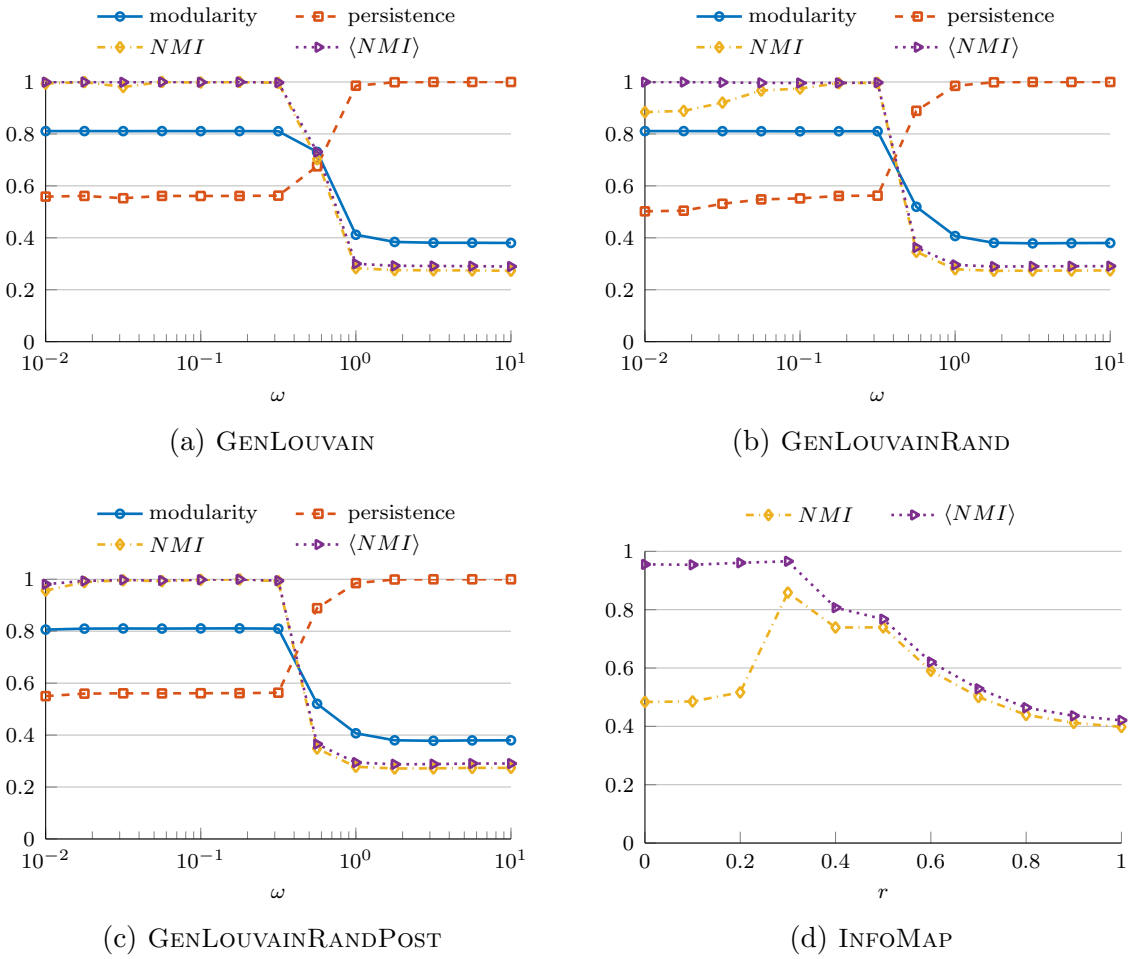


(a) GenLouvain (b) GenLouvainRand

(c) GenLouvainRandPost (d) InfoMap

Figure 4.4: Benchmark network with $p = 0.08$ (30% probability of re-sampling from the null-distribution) and $\mu = 0.3$

coupling parameter ω [21]). To mitigate this type of behaviour, GenLouvainRand chooses modularity-increasing moves at random (with probability proportional to the increase in modularity) during the first phase. Another problem of both GenLouvain and GenLouvainRand is that they tend to get stuck in local optima of multilayer modularity that have suboptimal persistence, i.e., we can change community labels in a way that increases the persistence of the partition without changing the partition of any layer of the multilayer network. GenLouvainRandPost uses a postprocessing step after the first pass through the first phase of GenLouvainRand that changes community labels to improve the persistence of the partition (see Appendix B for details).



(a) GenLouvain

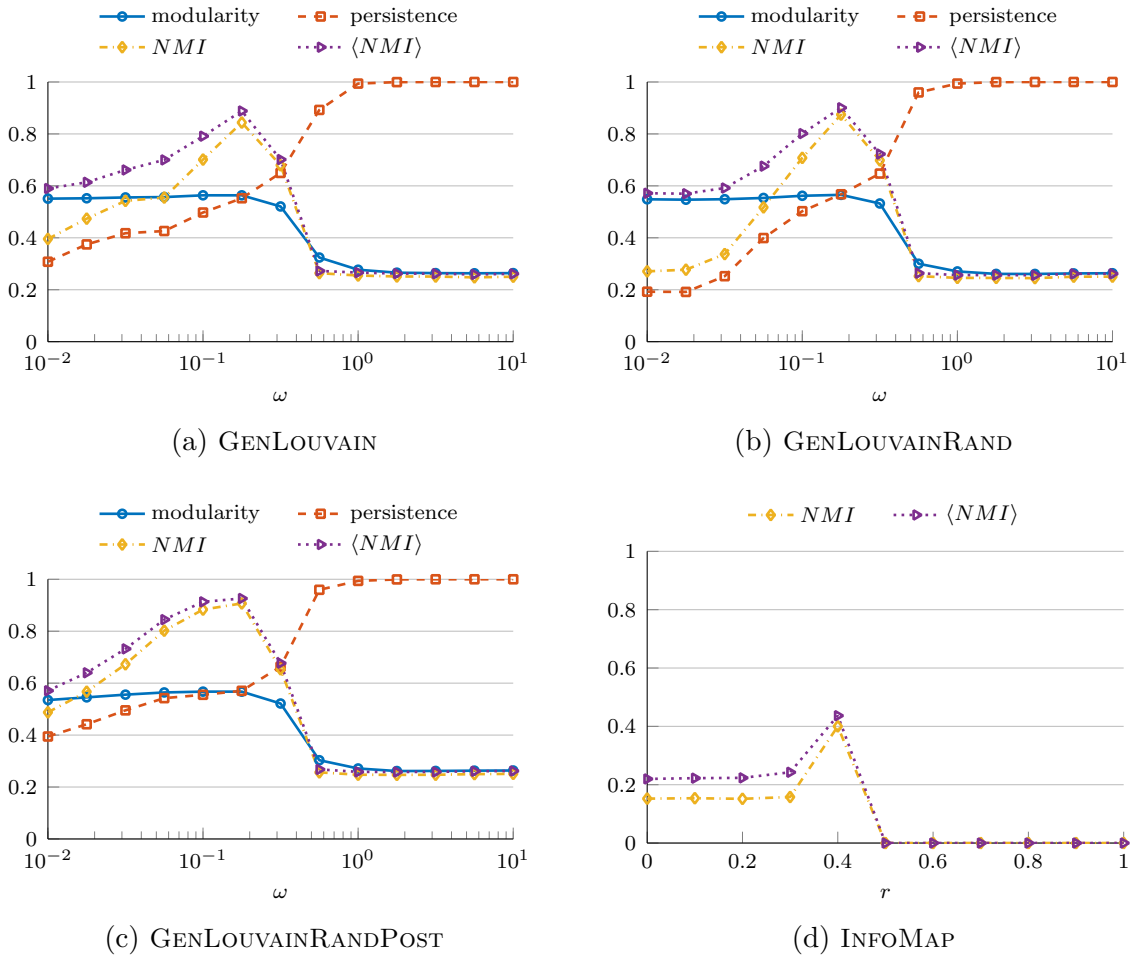
(b) GenLouvainRand

(c) GenLouvainRandPost

(d) InfoMap

Figure 4.5: Benchmark network with $p = 0.1$ (10% probability of re-sampling from the null-distribution) and $\mu = 0.1$

Each of Figs. 4.3–4.7 corresponds to a realisation of the benchmark with uniform multiplex structure (Fig. 4.1b) with different choices for the copying probability p and community mixing parameter μ. Each of the networks has 1000 nodes and 10 layers (for a total of 10 000 state nodes). The parameters for the community null distribution are $\tau_c = 1$, $c_{\min} = 50$, and $c_{\max} = 300$. The parameters for the degree distribution are $\tau_k = 3$, $k_{\min} = 3$, and $k_{\max} = 50$. We use the normalised mutual information between multilayer partitions (NMI) (see Section 2.5.3) and normalised mutual information between single-layer partitions averaged over all layers ($\langle NMI \rangle$) to compare the multilayer partitions identified by the different algorithms with the planted partition. For the multilayer modularity based algorithms we also show the



(a) GenLouvain

(b) GenLouvainRand

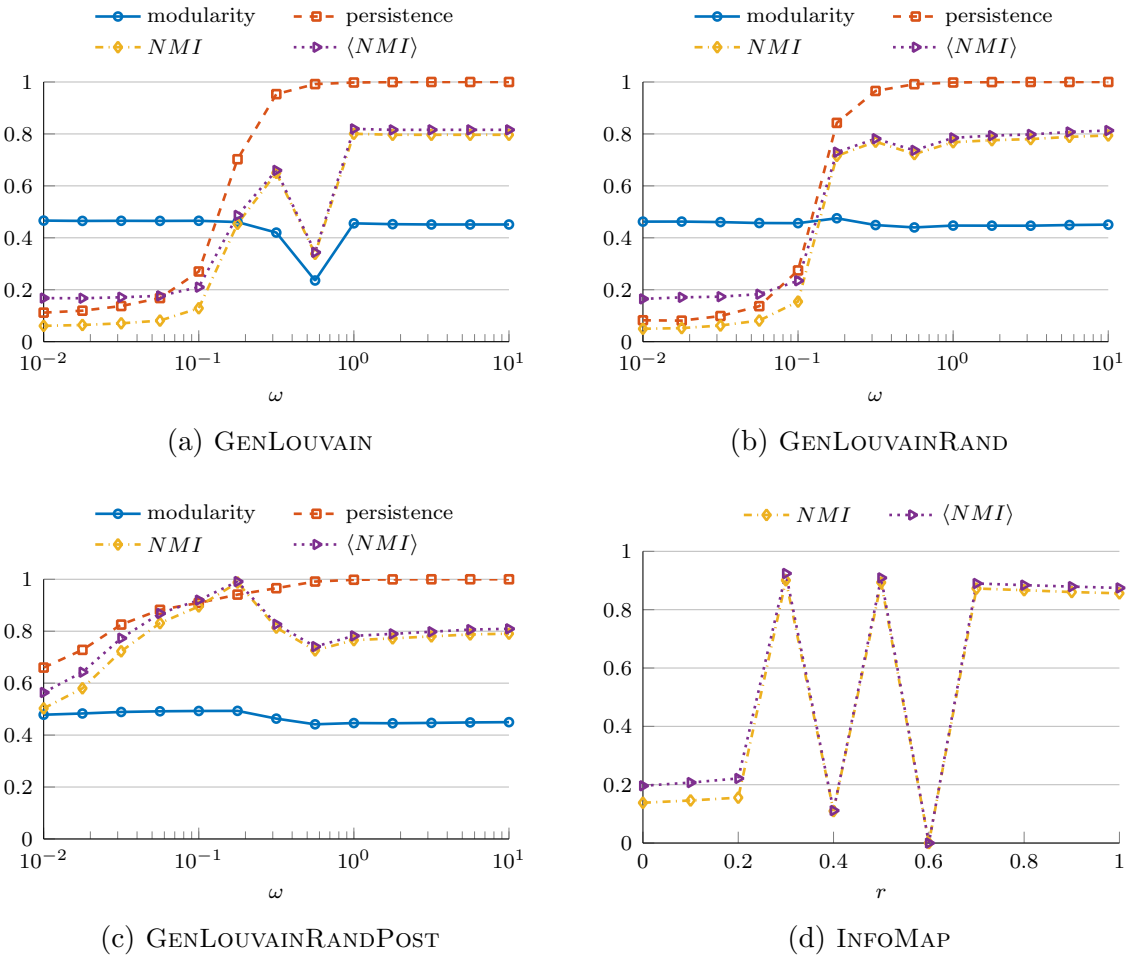
(c) GenLouvainRandPost

(d) InfoMap

Figure 4.6: Benchmark network with $p = 0.1$ (10% probability of re-sampling from the null-distribution) and $\mu = 0.3$

(intralayer) modularity (Eq. (2.35) with $\omega = 0$) and persistence of the multilayer partitions. All the results are averages over 10 runs of the algorithms.

The different benchmark networks illustrate different situations. In Figs. 4.3 and 4.5 the networks have very clear community structure that one can easily identify by analysing individual layers separately. One would expect any reasonable community detection algorithm to be able to recover the community structure in these networks for at least some parameter choices. We refer to this type of network as an *easy* benchmark. The more interesting case from a multilayer networks perspective are Figs. 4.4, 4.6, and 4.7 where the networks are sufficiently well mixed such that the community structure is not easily recoverable by analysing single layers. Here,



(a) GenLouvain

(b) GenLouvainRand

(c) GenLouvainRandPost

(d) InfoMap

Figure 4.7: Benchmark network with $p = 0.11$ (1% probability of re-sampling from the null-distribution) and $\mu = 0.4$

one would hope that by exploiting correlations between the layers of the network, we might still be able to recover the community structure. We refer to this type of network as a *hard* benchmark.

GenLouvain and GenLouvainRand mostly behave in a similar way on these benchmarks, however, we see evidence of strange behaviour as $\omega \to 1$ for GenLouvain in Figs. 4.4 and 4.7 (where the intralayer modularity contribution drops off and then recovers again) which is mitigated by the randomisation in GenLouvainRand. Both recover the planted partition on the individual layers for the easy benchmarks but do not always identify communities optimally across layers (note the discrepancy between the NMI and $\langle NMI \rangle$ in Fig. 4.3 in particular).

GenLouvainRandPost is particularly interesting. It generally does best on the hard benchmarks, except for Fig. 4.4 where there is little correlation between the partitions in different layers. In particular, unlike GenLouvain or GenLouvainRand, it can identify the planted partition in Fig. 4.7 where the correlation between layers is high but the structure in the individual layers is weak. However, as might be expected based on its emphasis on optimising persistence, it does worse than GenLouvain and GenLouvainRand when there is little correlation between layers (Figs. 4.3 and 4.4).

Multilayer InfoMap often shows strange behaviour (notably in Fig. 4.7 but also on many other networks not shown here) that seems to point more towards issues with the implementation rather than necessarily the quality function itself. Even on the easy benchmarks (Figs. 4.3 and 4.5), where it does give reasonable results, it is outperformed by the modularity-based methods. Notably, one could have also achieved better results by using InfoMap to identify communities on each layer separately.

Chapter 5

Applications to Survey Data

One of the questions that arose in discussions with our industrial collaborators is whether we can use network-based community detection methods to extract interesting information from data sets that do not have an immediately apparent network structure. The particular application we had in mind was analysing surveys on consumer opinions, beliefs, and characteristics. We start by computing similarity matrices between survey participants based on their answers to different survey questions. From these similarity matrices we can then construct networks that we can analyse using the community detection methods from Chapters 2 and 3. In this context, one would more commonly refer to communities as clusters, and there is some evidence that network-based community detection methods are well-suited for data clustering [45, 47, 82].

In this chapter we analyse a survey that explores the hair characteristics of females form different countries (Brazil, Japan, Thailand, UK, US), with an emphasis on hair dryness, damage, and the use of artificial hair colours. The survey questions are split into multiple modules which we can use to construct several distinct similarity matrices between the participants, yielding a multilayer-network representation of the survey.

	Nr. participants	Normal	Dry	Damaged	Coloured
Brazil	500	100	200	229	200
Japan	512	100	200	239	230
Thailand	600	100	291	324	236
UK	572	105	248	298	333
USA	533	104	201	250	252
Combined	2717	509	1140	1340	1251

Table 5.1: Split of the survey participants across countries and hair types

5.1 Overview of the Data Set

The data set we consider in this chapter consists of the set of responses to a survey designed to investigate the relationships between hair characteristics, the use of hair products (in particular conditioners and artificial hair colours), and attitudes and beliefs about hair in different countries. The participants were all female and between the age of 18–45. Table 5.1 shows the split of participants across countries and hair types. Note that a participant can identify their hair as belonging to more than one category (e.g. dry and damaged). The survey used a quota system to select participants with varied hair types, and hence the split across hair type categories is not representative of people in the different countries.

The survey is comprised of different modules, covering general habits, product and appliance usage, hair characteristics, hair colouring, dryness and damage. We mostly focus on analysing hair characteristics and use the answers to some of the other questions to interpret the results.

For the hair characteristics module, participants are asked to rate their hair on 28 different attributes. Each attribute consists of two extremes, and participants are asked to rate their current hair on a scale from 1 to 5 between these two extremes. Participants are also asked to rate their ideal hair on the same attributes and which, if any, attributes they want to change the most. Figures 5.1 and 5.2 show the lower triangular part of the correlation matrices between the different attributes for current and ideal hair. Some of the attributes are highly correlated, which we need to deal

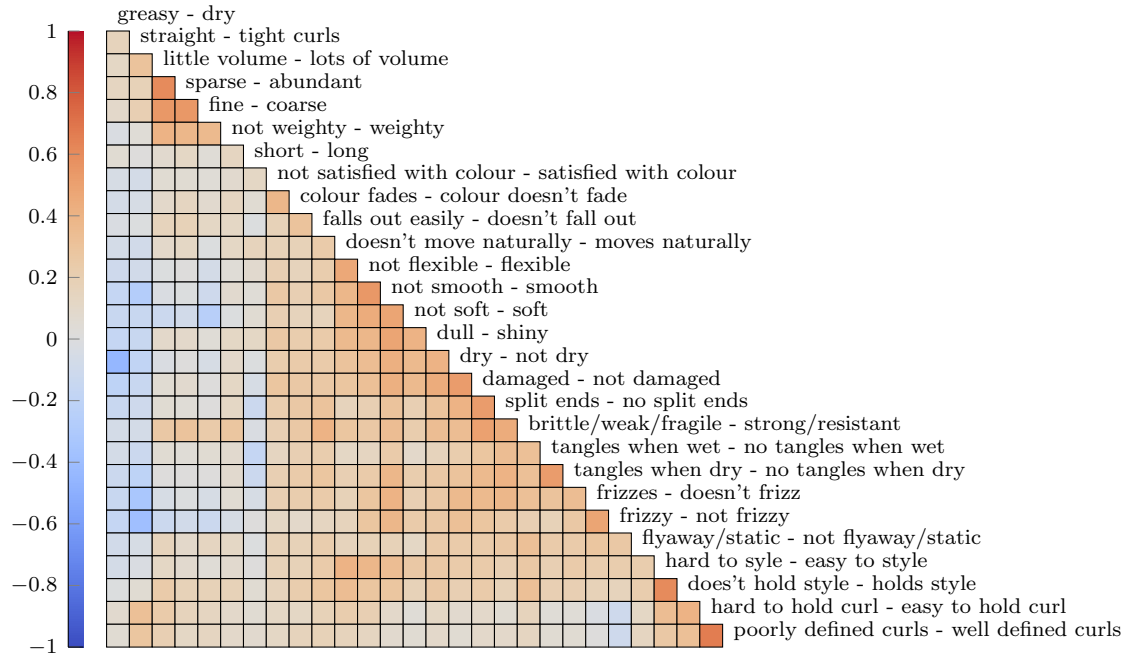


Figure 5.1: Lower triangular part of the Pearson correlation matrix between current hair attributes. The diagonal elements of the correlation matrix have been replaced by attribute labels for the corresponding row and column. As a result, the first square along the diagonal indicates the correlation between "greasy - dry" and "straight - tight curls". Each attribute consists of two extremes and participants rate their hair on a scale of 1 to 5 between the two extremes. The attribute labels indicate the two extremes of the scale, e.g., for "greasy - dry" a 1 indicates very greasy hair and a 5 indicates very dry hair.

with later. The pattern of correlations is also reasonably similar between current and ideal hair.

For the attribute "colour fades - colour doesn't fade" we have systematic missing data, as participants from Brazil were only asked to rate their hair on this attribute if they had artificially coloured hair, whereas all participants from the other countries were asked to rate their hair on this attribute. For the correlation matrices in Figs. 5.1 and 5.2, we estimate the correlation between two attributes based only on those participants that do not have missing values for either attribute. However, for further calculations in the next sections we exclude "colour fades - colour doesn't fade" to avoid any systematic bias.

While most of the analysis in this chapter focuses on current and ideal hair attributes, we also use the modules on dry and damaged hair to construct network

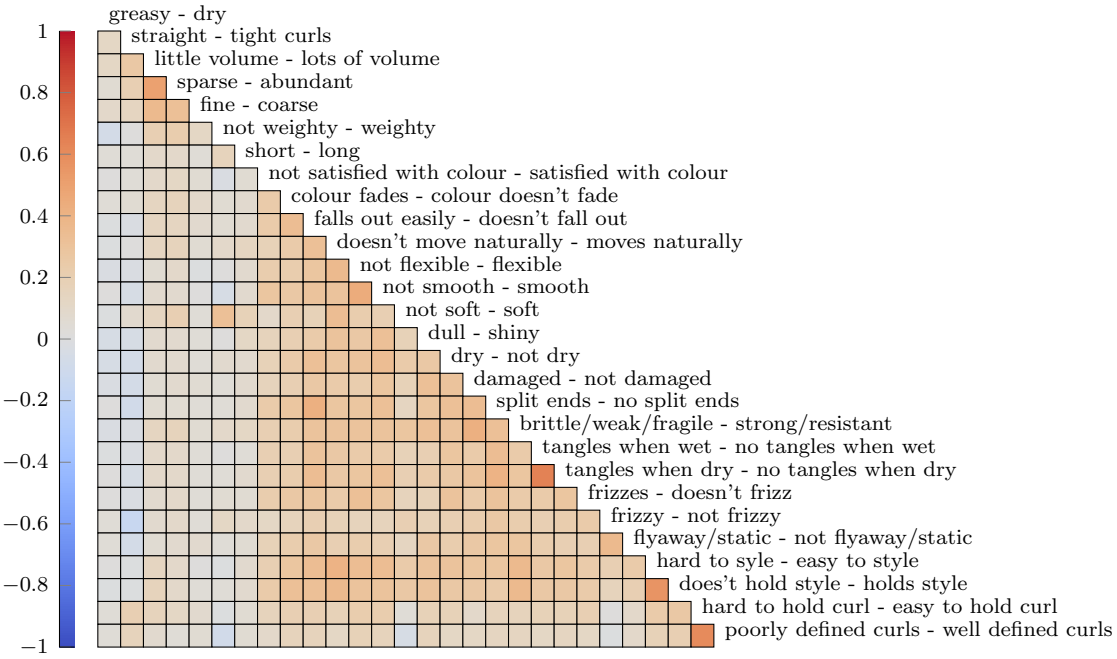


Figure 5.2: Lower triangular part of the Pearson correlation matrix between ideal hair attributes. The diagonal elements of the correlation matrix have been replaced by attribute labels for the corresponding row and column. As a result, the first square along the diagonal indicates the correlation between "greasy - dry" and "straight - tight curls". Each attribute consists of two extremes and participants rate their hair on a scale of 1 to 5 between the two extremes. The attribute labels indicate the two extremes of the scale, e.g., for "greasy - dry" a 1 indicates that the participant wants very greasy hair and a 5 indicates that they want very dry hair.

layers. The data for these two modules is of a rather different nature than that for the hair attributes. For each of dryness and damage, participants are asked to select symptoms and causes from a list of possible answers. We use this data to construct four network layers, one each for symptoms of dryness, causes of dryness, symptoms of damage, and causes of damage. Each of these layers encodes the similarity between participants based on their agreement on the symptoms and causes of dry and damaged hair.

In Section 5.1.1 we discuss current and ideal hair attributes in more detail, using principle component analysis to provide some context for the results we find when analysing the data as a multilayer network. We describe the methods we use to construct this network in Section 5.2, discussing different choices for similarity measures for each of the layers.

5.1.1 Principle Components for Current and Ideal Hair Attributes

One way to address the issue of correlated attributes is to use principle component analysis (PCA) (see e.g. Jolliffe [95]) to generate a set of uncorrelated pseudo-attributes that are linear combinations of the actual attributes. We can represent the survey responses as an $n \times m$ *data matrix* $\boldsymbol{X}$, where n is the number of participants and m is the number of attributes we want to consider. The goal of PCA is to find the orthonormal transformation of the data matrix that best explains its variance. Formally, we write

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{P}, \tag{5.1}$$

where $\boldsymbol{Z}$ is the matrix of principle component scores and $\boldsymbol{P}$ is the matrix of principle component coefficients. We write $\boldsymbol{Z}_{\cdot i}$ for the ith column of $\boldsymbol{Z}$, also called the ith principle component of $\boldsymbol{X}$. Analogously, we use $\boldsymbol{X}_{\cdot i}$ to denote the ith column of $\boldsymbol{X}$ and $\boldsymbol{P}_{\cdot i}$ for the ith column of $\boldsymbol{P}$. The coefficients are chosen to successively maximise the variance of the principle components subject to the constraint that the principle components are uncorrelated. Hence, the $\boldsymbol{P}_{\cdot i}$ satisfy the optimisation problem

$$\boldsymbol{P}_{\cdot i} = \underset{\boldsymbol{p}}{\operatorname{argmax}} \left(\operatorname{var}(\boldsymbol{X}\boldsymbol{p}) : \operatorname{cov}(\boldsymbol{X}\boldsymbol{P}_{\cdot k}, \boldsymbol{X}\boldsymbol{p}) = 0 \ (k < i), \ \boldsymbol{p}^T\boldsymbol{p} = 1 \right). \tag{5.2}$$

One can show that the $\boldsymbol{P}_{\cdot i}$ that satisfy this optimisation problem are the eigenvectors of the covariance matrix $\boldsymbol{\Sigma}$ of $\boldsymbol{X}$, where $\boldsymbol{\Sigma}_{ij} = \operatorname{cov}(\boldsymbol{X}_{\cdot i}, \boldsymbol{X}_{\cdot j})$. Furthermore, the variance of the ith principle component $\boldsymbol{Z}_{\cdot i}$ is given by the ith eigenvalue λ_i of $\boldsymbol{\Sigma}$, where we sort the eigenvalues and corresponding eigenvectors such that $\lambda_1 > \lambda_2 > \cdots > \lambda_m$.

The principle coefficients provide a set of orthonormal directions that best represent the variability in a data set and can often highlight interesting relationships between the original attributes. Keeping only the first k principle components gives the k-dimensional orthogonal projection of the data with the largest variance. Consequently, the projection of the data onto the first two principle components is par-

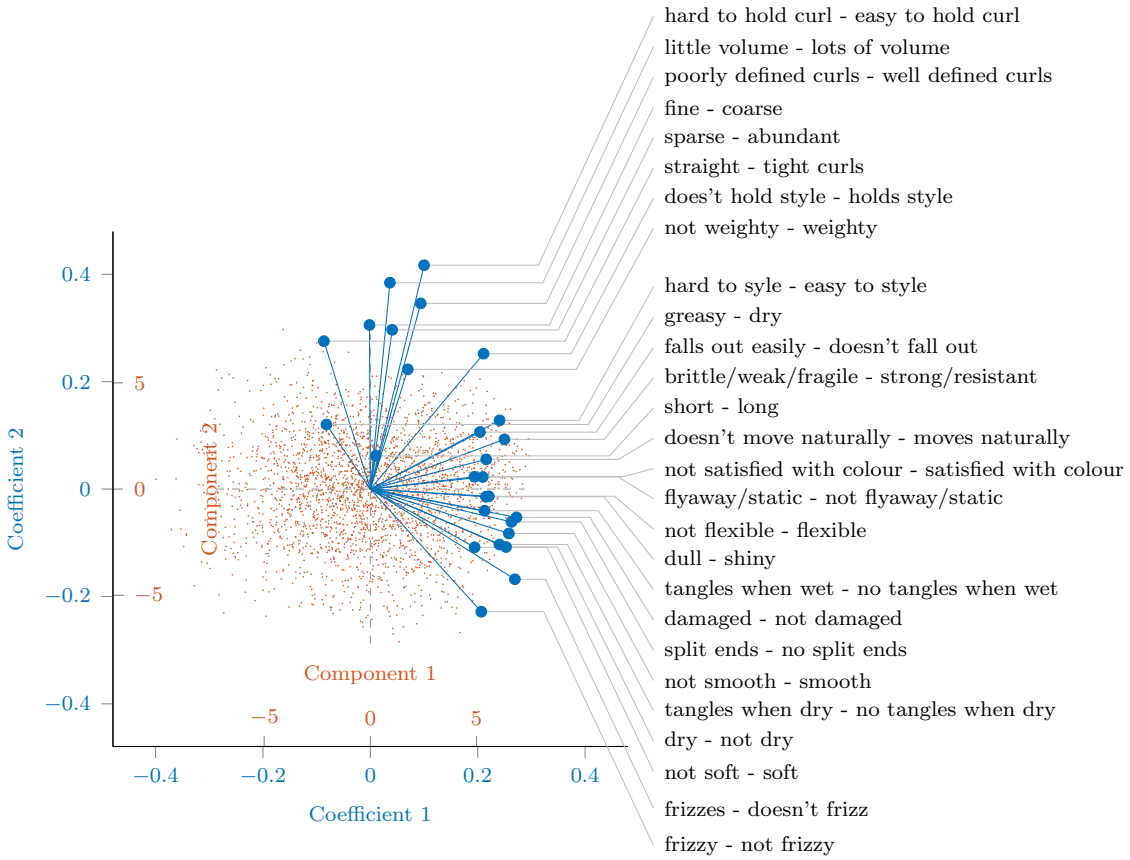


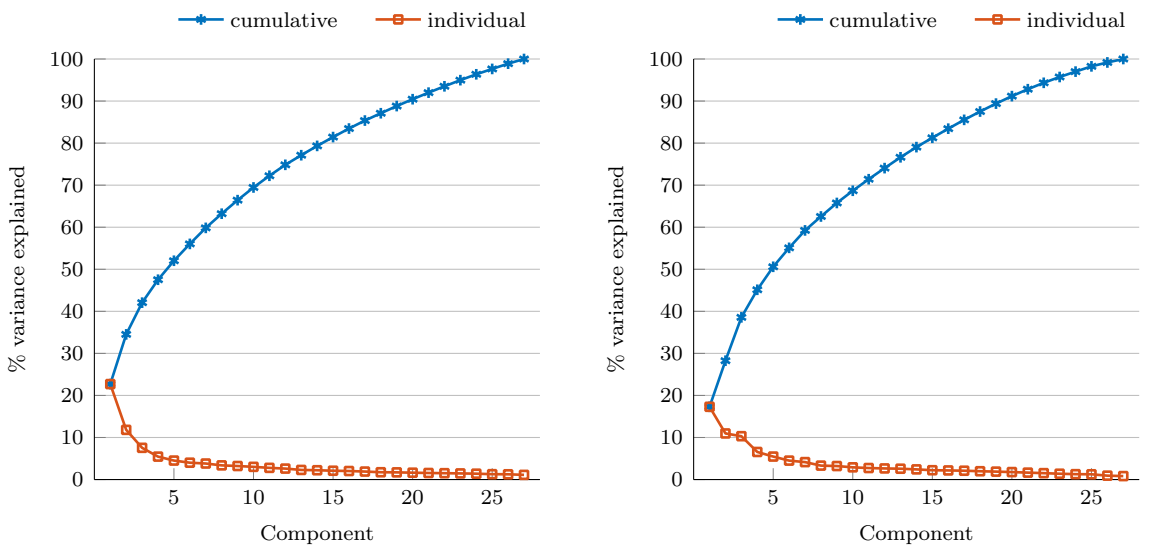
Figure 5.3: Biplot of first and second principle component for current hair attributes, showing the contributions of the attributes to the first and second component (blue vectors), as well as the projected, centred data (red points). From the attributes that have a large contribution to the first component, it appears to mostly indicate how satisfied participants are with their hair. The second component appears to be mostly related to curliness of the participants hair.

ticularly useful for visualisation purposes. Even after removing "colour fades - colour doesn't fade", we still have a small number of missing entries in the data matrix $\boldsymbol{X}$, though these do not appear to be systematic. We use an alternating least squares (ALS) algorithm (Roweis [178], as implemented in Matlab) to estimate the principle components, which can deal with some missing data. We also tried using this algorithm to estimate principle components based on all attributes, but it does not handle the systematic nature of the missing data well, producing results that strongly depend on the initial values for the principle coefficients used to start the ALS iteration.

Figures 5.3 and 5.4 show both the projection of the data onto the first and second

Figure 5.4: Biplot of first and second principle component for ideal hair attributes, showing the contributions of the attributes to the first and second component (blue vectors), as well as the projected, centred data (red points).

principle component as well as the contributions of the original attributes to the first two principle coefficients using a biplot. To read a biplot, note that one can interpret the attribute vectors as potential axes onto which one can project the data points, where the length of the attribute vector is a scaling factor for the axis. This is a simple consequence of the geometric properties of inner products. Let attribute j be represented in the biplot by the vector $\boldsymbol{a}_j = (\boldsymbol{P}_{j1}, \boldsymbol{P}_{j2})$ and data point i by the point $\boldsymbol{b}_i = (\boldsymbol{Z}_{i1}, \boldsymbol{Z}_{i2})$, then the estimated value of attribute j for data point i based on the first two principle components is $\tilde{\boldsymbol{X}}_{ij} = \langle \boldsymbol{a}_j, \boldsymbol{b}_i \rangle$, whereas the length of the orthogonal projection of $\boldsymbol{b}_i$ onto the axis corresponding to attribute j is $\frac{\langle \boldsymbol{a}_j, \boldsymbol{b}_i \rangle}{\|\boldsymbol{a}_j\|} = \frac{\tilde{\boldsymbol{X}}_{ij}}{\|\boldsymbol{a}_j\|}$. Thus one can geometrically reconstruct the estimated value of an attribute for a data point



(a) Current hair

(b) Ideal hair

Figure 5.5: Fraction of the total variance of the data set explained by successive principle components.

by first finding the orthogonal projection of the data point onto the line through the origin in the direction of the attribute vector of interest and then scaling it by the length of the attribute vector. Similarly, the angle between two attribute vectors indicates the correlation between the two attributes, where attributes that correspond to similar directions are highly correlated.

When interpreting these plots it is important to keep in mind that they only show a relatively small fraction of the variance of the original data as we can see from Fig. 5.5. The first two principle components only account for approximately 35% of the total variance for current hair and approximately 28% of the variance for ideal hair and keeping more principle components only slowly increases the fraction of the total variance preserved, indicating that this data set is not particularly well-embedded in a low-dimensional space. This analysis also implicitly assumes that the scale participants use to rate their hair is approximately linear.

Keeping these limitations in mind, the first two principle components nonetheless seem to reveal some intuitive properties of the data set. Looking at the attributes that have a large, positive contribution to the first principle component for current

hair attributes (see Fig. 5.3), we see that they are positive statements about the participants hair (e.g, "moves naturally","no tangles","not damaged", "no split ends"), suggesting that the first principle component gives an indication of how satisfied a participant is with their hair overall. Similarly, looking at the attributes that have a large positive contribution to the second principle component, we see that they seem to be related to curly hair (e.g. "easy to hold curl","lots of volume","well defined curls").

The first two principle components for the ideal hair attributes have a similar interpretation (see Fig. 5.4). This is perhaps a bit surprising, as the attributes most strongly associated with the first component would appear to have an obvious answer when talking about ideal hair. Nonetheless the data still shows large variability in this direction. Repeating the same analysis for each country individually produces mostly similar results (see Appendix A).

5.2 Network Construction

We would like to re-frame the survey data as a multilayer network between participants, so that we can apply the methods from Chapters 2 and 3 to identify groups of people with similar hair types and similar ideas about what they would want their hair to be like. Such groups could potentially help identify target demographics for new hair products.

As before, we represent the survey responses as a set $X = \{\boldsymbol{X}_k\}_{k=1}^{l}$ of $n \times m_k$ data matrices, one for each of l layers, where n is the number of participants and m_k is the number of attributes we want to consider for layer k. From the data matrices X, we want to construct an $n \times n \times l$ adjacency tensor $\boldsymbol{A}$, where $\boldsymbol{A}_{ik}^{jk}$ should be a measure of similarity between participants i and j in layer k. For each of the layers, we should measure similarity in a way that is appropriate for the data in that layer. For current and ideal hair attributes, the entries of the corresponding data matrix take discrete values in $\{1, \ldots, 5\}$, representing a rating between two extremes, whereas for dryness and damage symptoms and causes, the data matrices have binary entries

indicating whether a participant thinks that a particular attribute is a symptom or cause of dryness or damage. In the next section we discuss different possible choices for similarity measures for the different layers

5.2.1 Similarity Measures

There are many different ways one could compute a similarity measure from a data matrix. Similarity measures differ in the assumptions they make about the data and the kind of features they consider to be similar.

One of the key distinguishing factors between similarity measures is whether they treat the data as being measured on a "nominal", "ordinal", "interval", or "ratio" scale. This type of classification was originally proposed by Stevens [192], and in this context, a measurement scale is broadly defined as a rule for assigning numbers to objects or events, i.e., a measurement scale is a mapping $m : \mathbb{E} \to \mathbb{M}$ from an event space $\mathbb{E}$ to a measurements space $\mathbb{M} \subset \mathbb{R}$. Given a measurement scale $m : \mathbb{E} \to \mathbb{M}$, we can then consider a transformed measurement scale $\tilde{m} = t \circ m : \mathbb{E} \to \tilde{\mathbb{M}}$, where $t : \mathbb{M} \to \tilde{\mathbb{M}}$, $\mathbb{M}, \tilde{\mathbb{M}} \subset \mathbb{R}$ is a transformation. We then classify similarity measures based on the types of transformations of the measurement scale that leave the similarity measure invariant. Nominal measures are invariant under any bijective (i.e., invertible) transformation of the measurement scale, ordinal measures are invariant under any monotonically increasing (i.e., rank-preserving) transformation, interval measures are invariant under any linear transformation (i.e., preserving the length of intervals), and ratio measures are invariant under transformation by multiplication by a constant (i.e., preserving ratios).

Note that Stevens originally used this framework to classify measurement scales, which can be problematic (see e.g. Velleman and Wilkinson [203] for some discussion), as we can often meaningfully convert measurement scales to a different type. However, classifying measures in this way does tell us which aspects of the data they consider as meaningful. For example, if we apply a nominal measure to data that one could consider as being ordinal, we loose any information inherent in the order

of the categories, but we can still obtain meaningful results based on the information inherent in category memberships. Similarly, if we apply an interval measure to ordinal data, we can still obtain meaningful results, though we have to worry about the robustness of the results and potential biases. Furthermore, if the event space could conceivably be measured on an interval or ratio scale, we might be able to improve the results by applying an appropriate monotonic transformation to the data, such that the transformed measurements more closely resemble an interval or ratio scale for the event space. In fact, one can often construct an ordinal measure from an interval measure by transforming the data in such a way that the transformed data is invariant under monotonic transformations and then applying an interval measure to the transformed data.

Another key difference between similarity measures is whether they assume that all attributes are measured on the same scale (i.e., they are only invariant when applying the appropriate transformations to the entire data matrix) or not (i.e., they are invariant even when applying different transformations of the appropriate type to each column of the data matrix). In a similar way to the situation for applying interval measures to ordinal scales, we might be able to improve results for measures that assume that attributes are measured on the same scale by applying standardising transformations to each of the attributes.

Whether one can meaningfully consider survey responses on a rating scale (often referred to as a Likert scale [132]) as interval or merely ordinal has been a heavily debated topic [1, 32, 104, 197, 203]. For the survey we are considering in this chapter, one could argue that the rating scale should be a reasonable approximation of an interval scale for the participants' perception of their hair, as, unlike in a traditional Likert scale, the intermediate levels between the two extremes of an attribute do not have a specific label, which should help avoid introducing any systematic bias in the perceived length of the intervals between rating levels. However, if we want to consider the responses as a proxy for some physical properties of the participants hair, the interval assumption is unlikely to hold.

Many different similarity measures have been developed in different fields [143,

194]. In the following sections we discuss some of the more popular measures and the assumptions they make about the underlying data.

Cosine Similarity

The cosine similarity uses the cosine of the angle between rows of the data matrix (considered as vectors in $\mathbb{R}^m$) as a measure of similarity. Note that the cosine of the angle θ between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is related to the inner product of the two vectors by the formula $\cos(\theta) = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|}$. Hence, one way to write the cosine similarity matrix is as

$$\boldsymbol{S}^{\cos}(\boldsymbol{X}) = \hat{\boldsymbol{X}} \hat{\boldsymbol{X}}^T, \tag{5.3}$$

where $\hat{\boldsymbol{X}}$ is a row-normalised data matrix, obtained from $\boldsymbol{X}$ by applying the normalising transformation $\hat{\boldsymbol{X}}_{i\cdot} = \frac{\boldsymbol{X}_{i\cdot}}{\|\boldsymbol{X}_{i\cdot}\|}$ to each row of $\boldsymbol{X}$. One way to interpret this normalisation is that we are treating each row of the data matrix as being measured on an independent ratio scale. Cosine similarity is thus a ratio measure, as it is invariant under global rescaling of the data but not under global linear shifts. It also assumes that each attribute is measured on the same scale.

Pearson Correlation

The classical way to interpret the Pearson correlation coefficient is as a measure of the deviation from a linear relationship between two variables [106]. However, one can also interpret it as the cosine similarity applied to row-centred data, i.e.,

$$\boldsymbol{S}^{\text{corr}}(\boldsymbol{X}) = \boldsymbol{S}^{\cos}(\overset{*}{\boldsymbol{X}}), \tag{5.4}$$

where $\overset{*}{\boldsymbol{X}}$ is a row-centred data matrix, obtained from $\boldsymbol{X}$ by applying the centring transformation $\overset{*}{\boldsymbol{X}}_{i\cdot} = \boldsymbol{X}_{i\cdot} - \langle \boldsymbol{X}_{i\cdot} \rangle$ to each row of the data matrix. This transformation corresponds to treating each row of the data matrix as being measured on an independent interval scale.

Both correlation and cosine similarity are often used in the collaborative filtering approach to recommender systems [28, 135, 170, 182, 187], which often has to work

with rating data that is similar in nature to the kind of data we are considering in this chapter. Correlation and Cosine similarity are examples of an inner product applied to a transformed version of the data, an approach that one can generalise by applying other transformations to the data, e.g., subtracting the column mean before computing the cosine similarity (adjusted cosine similarity, [182]) or subtracting a global central value before computing the cosine similarity [187].

Centred Cosine Similarity

We define an additional inner-product based similarity by transforming the data matrix in a way that combines the properties of the transformations used to define the Pearson correlation and adjusted cosine similarity. We define the centred cosine similarity as

$$\boldsymbol{S}^{\text{cen}}(\boldsymbol{X}) = \boldsymbol{S}^{\cos}(\overset{**}{\boldsymbol{X}}), \tag{5.5}$$

where $\overset{**}{\boldsymbol{X}}$ is the double centred data matrix with entries $\overset{**}{\boldsymbol{X}}_{ij} = \boldsymbol{X}_{ij} - \langle \boldsymbol{X}_{i\cdot} \rangle - \langle \boldsymbol{X}_{\cdot j} \rangle + \langle \boldsymbol{X} \rangle$. Note that, as the double centred data matrix has zero row and column means, when applied to this matrix, cosine similarity, Pearson correlation and adjusted cosine similarity are all equivalent. The double centred data matrix is invariant when applying the same linear transformation to each element of the data matrix, and hence the centred cosine similarity is an interval measure.

Spearman Correlation

Spearman [191] proposed a measure of similarity for ordinal data based on the Pearson correlation. Spearman correlation measures the deviation from a perfect monotonic relationship between two data vectors. To compute the Spearman correlation, we first transform the data by rank-ordering, i.e., we apply the rank-ordering transformation

$$\boldsymbol{R}_{ij} = \frac{|\{k : \boldsymbol{X}_{ik} < \boldsymbol{X}_{ij}\}| + 1 + |\{k : \boldsymbol{X}_{ik} \leq \boldsymbol{X}_{ij}\}|}{2}$$

to the data matrix, where we have resolved ties by using the average rank of the tied values. We then compute the Spearman correlation by computing the Pearson

correlation based on the rank-ordered data, i.e.,

$$\boldsymbol{S}^{\text{spear}}(\boldsymbol{X}) = \boldsymbol{S}^{\text{corr}}(\boldsymbol{R}). \tag{5.6}$$

When computing the Spearman Correlation, we treat each row of the data matrix as being measured on an independent ordinal scale, as the rank-ordering transformation is monotone and non-linear. It is an ordinal measure, as the rank-ordered data is invariant under monotone transformations of the original data.

Information-Theoretic Similarity

A different similarity measure that can be applied to ordinal data was proposed by [133]. Generically, it defines the similarity between two observations a and b as the ratio of the information that a and b have in common to the information needed to describe a and b, i.e.,

$$\text{sim}(a, b) = \frac{I(\text{common}(a, b))}{I(\text{description}(a, b)}, \tag{5.7}$$

where the information of an event e is $I(e) = -\log(\mathbb{P}[e)$. Note that $I(\text{common}(a, b))$ is different from the mutual information [42] which measures the information in common between two distributions (and thus cannot account for any particular features of the random variables we are interested in, such as being measured on an ordinal scale), whereas here we consider $\text{common}(a, b)$ as a single random variable, which provides us with the necessary flexibility to take different features of a and b into account. For two observations from an ordinal scale, [133] suggest defining $\text{common}(a, b) = \{a \in [\min(a, b), \max(a, b)] \wedge b \in [\min(a, b), \max(a, b)]\}$, and $\text{description}(a, b) = \{a \wedge b\}$. Assuming that a and b are realisations of independent random variables with distributions $\mathbb{P}_a$ and $\mathbb{P}_b$ respectively, we then have

$$\text{sim}(a, b) = \frac{\log\left(\sum_{e\in[\min(a,b),\max(a,b)]} \mathbb{P}_a[e]\right) + \log\left(\sum_{e\in[\min(a,b),\max(a,b)]} \mathbb{P}_b[e]\right)}{\log(\mathbb{P}_a[a]) + \log(\mathbb{P}_b[b])}. \tag{5.8}$$

In order to define a similarity matrix based on this measure, we need to generalise

it to vector-valued observations. If we assume that attributes are independent and observations are independent identically distributed, we can write the elements of the similarity matrix as

$$\boldsymbol{S}_{ij}^{\text{its}} = \frac{2\sum_{k=1}^{m} \log\left(\sum_{e\in[\min(\boldsymbol{X}_{ik},\boldsymbol{X}_{jk}),\max(\boldsymbol{X}_{ik},\boldsymbol{X}_{jk})]} \mathbb{P}_k[e]\right)}{\sum_{k=1}^{m} \log(\mathbb{P}_k[\boldsymbol{X}_{ik}]) + \log(\mathbb{P}_k[\boldsymbol{X}_{jk}])}, \tag{5.9}$$

where $\mathbb{P}_k$ is the probability distribution for the k^{th} attribute, which we can estimate using

$$\mathbb{P}_k[e] = \frac{|\{i : \boldsymbol{X}_{ik} = e\}|}{n}. \tag{5.10}$$

This similarity measure is ordinal and does not assume that attributes are measured on the same scale as it is invariant under monotonic transformations of the columns of the data matrix. However, it does assumes that participants use the same scale.

Distance-Based Similarity Measures

Another popular approach for defining similarity measures is to transform a distance metric (i.e. a measure of dissimilarity) into a similarity measure. If $\text{dist}(\boldsymbol{x}, \boldsymbol{y})$ is a distance measure, popular choices for a similarity measure [47, 194] are the multiplicative inverse distance

$$\text{sim}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{1 + \text{dist}(\boldsymbol{x}, \boldsymbol{y})}, \tag{5.11}$$

the additive inverse distance

$$\text{sim}(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{\text{dist}(\boldsymbol{x}, \boldsymbol{y})}{\max_{\boldsymbol{x},\boldsymbol{y}} \text{dist}(\boldsymbol{x}, \boldsymbol{y})}, \tag{5.12}$$

and the exponential inverse distance

$$\text{sim}(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\text{dist}(\boldsymbol{x}, \boldsymbol{y})\right). \tag{5.13}$$

As for similarity measures, there are many possible ways to define the distance between two vectors. Perhaps the most popular choice is the euclidean distance $\text{dist}^{\text{euc}}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\| = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y})}$.

Jaccard Coefficient

The Jaccard coefficient [90] is a measure of similarity between two finite sets, defined as the ratio of the size of the intersection of the two sets to the size of their union, i.e. for two sets S_1 and S_2, the Jaccard coefficient is

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}. \tag{5.14}$$

One can also define the Jaccard coefficient for binary data (i.e, data that only takes values in $\{0, 1\}$) by interpreting a binary vector as an indicator vector for a set. For a binary data matrix $\boldsymbol{X}$, one can write the entries of the Jaccard similarity matrix as

$$\boldsymbol{S}_{ij}^{\text{jac}} = \frac{\boldsymbol{X}_{i\cdot}\boldsymbol{X}_{j\cdot}{}^{T}}{\|\boldsymbol{X}_{i\cdot}\|_1 + \|\boldsymbol{X}_{j\cdot}\|_1 - \boldsymbol{X}_{i\cdot}\boldsymbol{X}_{j\cdot}{}^{T}}. \tag{5.15}$$

Note that the Jaccard coefficient applied to binary data is also known as Tanimoto similarity [172].

5.2.2 Permutation Null Models

When working with similarity measures, we want to be able to distinguish features that could be the result of random noise from those that give meaningful insights about the structure of the data. Most analytic results for the sampling distributions of the similarity measures we discuss in the previous section, where available, require us to assume that the data is normally distributed. An alternative, non-parametric approach is to use permutation tests [30, 57] to estimate null-distributions for the similarity measures. Permutation tests work by recomputing the statistic we want to test for many shuffled samples of the original data to estimate the null distribution, where we randomly shuffle the original data in a way that is consistent with the null hypothesis we want to test. This way we can estimate a p-value for each entry of the similarity matrix which we can use to identify the entries of the similarity matrix that are statistically significant. Here we are interested in identifying pairs of participants that are more similar than would be expected at random. We can estimate the p-value

for this one-sided alternative hypothesis for a pair of participants as the fraction of permuted data matrices for which the two participants are more similar than what we observe for the original data. That is, for a sample T of permutations of the elements of the data matrix, we estimate the p-value for a pair of participants i and j as

$$p(i,j) = \frac{|\{\tau \in T : \text{sim}(\tau(\boldsymbol{X})_{i\cdot}, \tau(\boldsymbol{X})_{j\cdot}) \geq \text{sim}(\boldsymbol{X}_{i\cdot}, \boldsymbol{X}_{j\cdot})\}|}{|T|}, \tag{5.16}$$

where $\tau(\boldsymbol{X})_{ij} = \boldsymbol{X}_{\tau(ij)}$ and $\tau : \{1,\ldots,n\} \times \{1,\ldots,m\} \leftrightarrow \{1,\ldots,n\} \times \{1,\ldots,m\}$. We want to compare the similarity between participants we observe to those we would expect to see if participants were independent. We consider two null models which differ in the other assumptions we make and thus restrict the space of permutations we sample from in different ways.

Column Permutation Null Model

For the first null model we assume that participants are independent and identical and attributes are independent. Under these assumptions, the distribution for the rating of participant i on attribute j only depends on the attribute, i.e., we can state our null hypothesis as $H_0 : \boldsymbol{X}_{ij} \sim \mathbb{P}_j$. Hence, under the null hypothesis, the distribution of the data is invariant if we apply a permutation to each of the columns of the data matrix, i.e., $\boldsymbol{X}_{ij} \sim \boldsymbol{X}_{\tau_j(i)j}$ for all permutations $\tau_j : \{1,\ldots,n\} \leftrightarrow \{1,\ldots,n\}$. Consequently, we can estimate the null distribution for a similarity measure by calculating the similarities for a uniform random sample of column-permuted data matrices. Another important consequence of this null hypothesis is that each row of the data matrix has the same null distribution, and hence each of the off-diagonal entries of the similarity matrix has the same null distribution. Thus, a single random sample of the data matrix yields $n(n-1)/2$ samples from the null distribution for the similarity values (i.e., the null model is uniform), which makes this model easy to estimate.

Row Permutation Null Model

The uniform nature of the column permutation null model makes it efficient to estimate but could potentially mask differences between participants. The second permutation null model we consider assumes that participants are independent and attributes are independent and identical. Under these assumptions, the distribution for the rating of participant i on attribute j only depends on the participant, i.e., we can state our null hypothesis as $H_0 : \boldsymbol{X}_{ij} \sim \mathbb{P}_i$. Hence, under the null hypothesis, the distribution of the data is invariant if we apply a permutation to each of the rows of the data matrix, i.e., $\boldsymbol{X}_{ij} \sim \boldsymbol{X}_{i\,\tau_i(j)}$ for all permutations $\tau_i : \{1, \ldots, m\} \leftrightarrow \{1, \ldots, m\}$. Similarly to the column permutation null model, we can thus estimate the null distribution for a similarity measure under this null hypothesis by calculating the similarities for a uniform random sample of row-permuted data matrices. Unlike for the column permutation null model, we get a different distribution for each entry of the similarity matrix.

5.2.3 Adjacency Tensor Construction

While the order we consider the layers in is arbitrary for this data set, in order to define an adjacency tensor, we need to fix a specific order. We define a total of 6 layers based on the survey data (see Section 5.1), which we consider in the order: current hair attributes, ideal hair attributes, symptoms of dry hair, causes of dry hair, symptoms of damaged hair, and causes of damaged hair.

For each layer we need to specify the intralayer contributions to the adjacency tensor based on a similarity measure that is appropriate for the data in that layer. For the first two layers, one could argue that interval or ordinal measures are appropriate, whereas for the other layers we can use nominal or binary measures (see Section 5.2.1). Given a choice of similarity measure, one could use the values of similarity between participants directly to define the intralayer edges, resulting in a weighted (potentially signed), fully connected network in each layer. However, the structural properties of networks constructed in this way are often not desirable. One particular problem

is that many similarity measures exhibit values that are far away from zero, even when the underlying data is randomly generated in such a way that participants are independent, provided the sample size (i.e. the number of attributes considered) is fairly small. This is the case for all the similarity measures we consider on this data set as we can see from the results for the column and row permutation null models in Figs. 5.6ab, 5.7ab, 5.8ab, and 5.9ab. These large random effects can mask structure that is present in the data. This affects the local methods we discuss in Chapter 3 particularly strongly, as a random walk on these networks looses any local information after a few iterations. Furthermore, the dense nature of a network constructed in this way makes these algorithms inefficient as their computational complexity depends on the size of the neighbourhood of a node (which in this case is the entire network).

Ideally we want to construct a sparse network based on a similarity measure in such a way that the topology of the network reflects the topology of the underlying data. One way to approach this problem is to use a null model, such as the permutation null models from Section 5.2.2, to identify and remove any edges from the similarity network that are not statistically significant. For a given choice of null model and similarity measure for each layer, we then define the intralayer contributions to the adjacency tensor as

$$\boldsymbol{A}_{ik}^{jk} = \begin{cases} \mathrm{sim}_k(\boldsymbol{X}_{ki\cdot}, \boldsymbol{X}_{kj\cdot}), & p_k(i,j) \leq \alpha \\ 0, & \text{otherwise,} \end{cases} \tag{5.17}$$

where $p_k(i,j)$ is the p-value (for a given choice of null model) indicating whether participants i and j are significantly more similar in layer k than under the null model and α is the significance level. Unless stated otherwise, we use $\alpha = 0.05$ to construct adjacency tensors.

For the column null model, this procedure is equivalent to choosing a uniform threshold for each layer and deleting any edges that have a weight that is smaller than the threshold. The main advantage of choosing the threshold in this way is that it provides a consistent way of fixing the threshold for different similarity measures.

For the row null model, this procedure effectively defines a different threshold for each pair of nodes, which should allow it to adapt to local variations in the data.

5.3 Results

Given the breadth of possible choices for similarity measures, and in particular the debate about which measures are appropriate for rating data as we have for current and ideal hair attributes, one of the key questions is how the different choices of similarity measures affect the structure of the networks we construct to represent the data. We are especially interested in the properties of community structure in these networks and to what extent they are robust across different similarity measures and null models.

First, we analyse the intralayer networks for current and ideal hair attributes that we obtain for different similarity measures separately. While the different measures behave in rather different ways, we find that the networks we construct based on the different measures have rather similar properties with respect to their community structure. Next, we look at community structure in the full multilayer network using multiplex modularity and finally we use a consensus clustering approach to identify a small set of representative partitions of the multilayer networks that we analyse in detail to interpret the data.

5.3.1 Comparison of Similarity Measures for Current and Ideal Hair Attributes

As we discuss in Section 5.2.1, for current and ideal hair attributes, we have a choice as to whether we interpret the data as interval (i.e., we assume that the gabs between the different rating levels are roughly equally spaced, e.g., the difference between 3 and 4 is roughly the same as the difference between 4 and 5) or ordinal (i.e., only the order of the data values matters). We consider four different similarity measures in this section, three interval measures (cosine (Eq. (5.3)), centred cosine (Eq. (5.5)), additive inverse euclidean distance (inverse euclidean, Eq. (5.12))) and one ordinal

	cosine		centred cosine		inverse euclidean		info-sim	
	col	row	col	row	col	row	col	row
$C_{\max}$	2629	2693	2693	2693	2693	2693	2693	2693
n_C	65	1	1	1	1	1	1	1
$d_{\max}$	1914	1057	581	582	2671	1564	1041	561
$\langle d \rangle$	729.25	398.22	259.36	268.08	1704.7	692.35	398.21	215.31
$d_{\min}$	1	54	59	61	126	184	21	46
ρ	0.2709	0.1479	0.0963	0.0996	0.6332	0.2572	0.1479	0.0800
$w_{\min}$	0.9093	0.7190	0.2836	0.2673	0.4709	0.0991	0.5592	0.1688

Table 5.2: Basic properties of the networks for current hair attributes based on different similarity measures and null models: number of nodes in the largest connected component $C_{\max}$, number of components n_C, maximum degree $d_{\max}$, mean degree $\langle d \rangle$, minimum degree $d_{\min}$, density ρ, and minimum edge weight $w_{\min}$. (Note that the degrees and density are calculated from the unweighted networks.)

measure (information theoretic similarity (info-sim, Eq. (5.9))). The main purpose of this comparison is to get a better understanding as to what extent the distinction between interval and ordinal measures matters in practise. Additionally, features of the data that can consistently be identified using different similarity measures are less likely to be an artifact of a particular way to analyse the data.

For each of the four similarity measures, we construct four different networks using the method described in Section 5.2.3, using either the column or row permutation null model (see Section 5.2.2) to threshold the similarity values for current or ideal hair attributes. Tables 5.2 and 5.3 show some basic properties of the networks generated in this way. One feature to note is that the column null model has a tendency to fragment part of the network into many small components (in particular when using the cosine similarity), a feature that becomes more pronounced as we decrease the significance level α. Overall, the networks we generate are fairly dense, with more than half of all possible edges present in some cases (most notably for the networks generated using the inverse euclidean similarity) and with the sparsest network having about 8% of all possible edges present. We can also see the local adaptive effect of the row null model, as the minimum edge weight for the row null model tends to be much lower than for the corresponding network generated using the column null

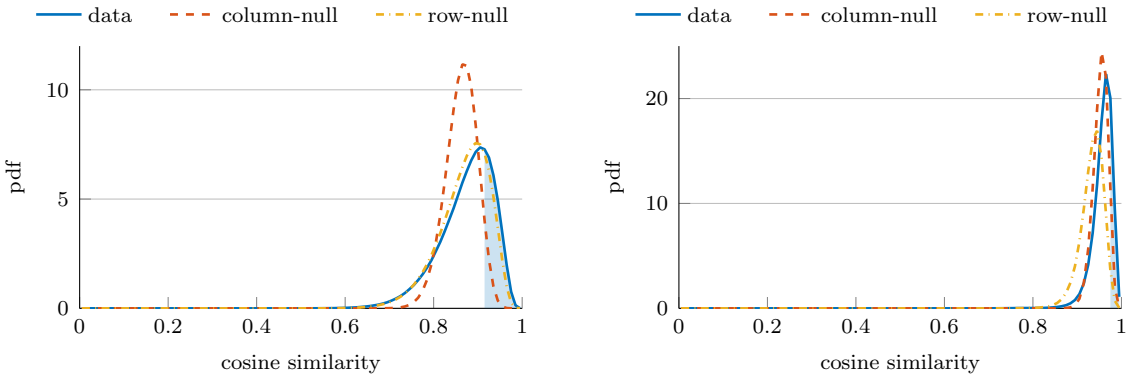
	cosine		centred cosine		inverse euclidean		info-sim	
	col	row	col	row	col	row	col	row
$C_{\max}$	2549	2693	2693	2693	2686	2687	2693	2693
n_C	144	1	1	1	8	3	1	1
$d_{\max}$	1637	2355	466	525	2551	2324	1022	817
$\langle d \rangle$	466.86	1606.6	209.71	282.45	1719.3	1519.0	360.92	242.20
$d_{\min}$	1	10	18	37	1	2	2	4
ρ	0.1734	0.5968	0.0779	0.1049	0.6387	0.5643	0.1341	0.0900
$w_{\min}$	0.9733	0.8010	0.3178	0.2463	0.5739	0.2304	0.5392	0.0936

Table 5.3: Basic properties of the networks for ideal hair attributes based on different similarity measures and null models: number of nodes in the largest connected component $C_{\max}$, number of components n_C, maximum degree $d_{\max}$, mean degree $\langle d \rangle$, minimum degree $d_{\min}$, density ρ, and minimum edge weight $w_{\min}$. (Note that the degrees and density are calculated from the unweighted networks.)

model, with the exception of the networks generated using the centred cosine similarity, where the two networks have very similar properties. Based on their density and degree distributions it appears that the networks generated using the centred cosine and info-sim measures might be more similar to each other than they are to the other networks, which would suggest that the distinction between interval and ordinal measures does not have a strong effect on the structure of the networks we generate and that other aspects of the similarity measures are more important.

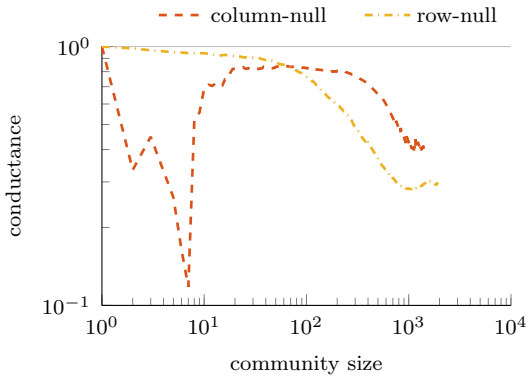
Comparing the distributions of the different similarity measures, we see that the cosine similarity takes values that are close to one for all pairs (see Figs. 5.6ab), which leaves little range to distinguish between pairs of participants that we would consider to be similar and those that we would consider to be dissimilar. Another undesirable feature of both the cosine and additive inverse euclidean similarity (see Figs. 5.8ab) is that the distributions for the data and the different null models have different modes, suggesting that these similarity measures are sensitive to global features of the data that we do not want to take into account. Both the centred cosine (Figs. 5.7ab) and information theoretic similarity (Figs. 5.9ab) are reasonably well behaved with respect to those properties, which should make their values easier to interpret.

We use the local community detection methods from Chapter 3 to compare the

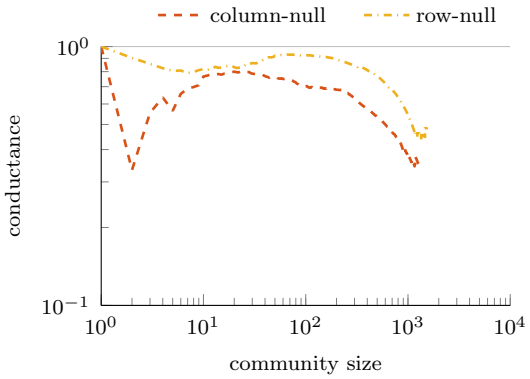


(a) Similarity distributions (current hair)

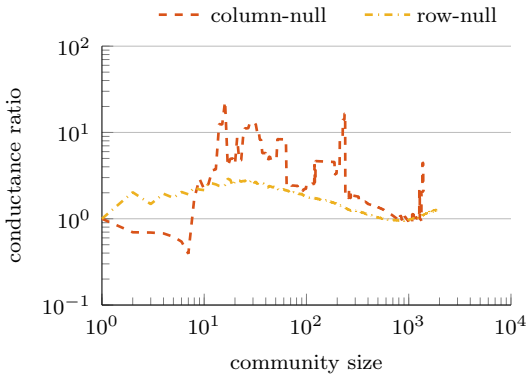
(b) Similarity distributions (ideal hair)



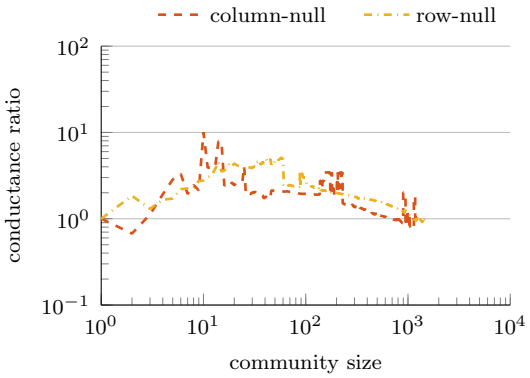
(c) NCPs (current hair)



(d) NCPs (ideal hair)

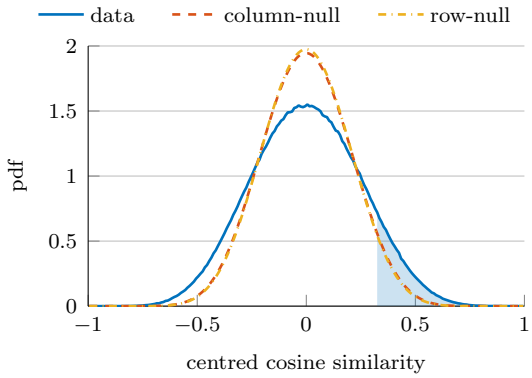


(e) CRPs (current hair)

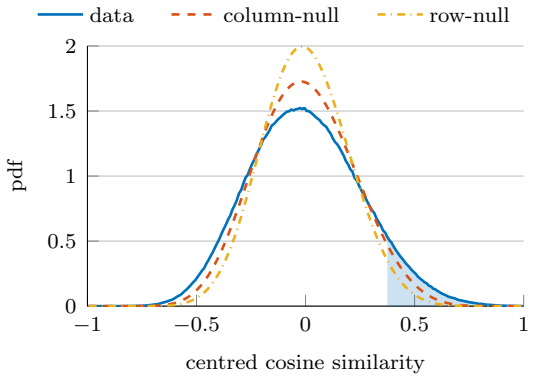


(f) CRPs (ideal hair)

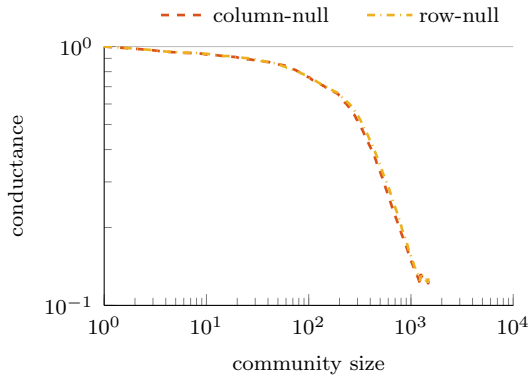
Figure 5.6: Overview of the structure of the networks generated using the *cosine similarity* for current hair attributes (left column) and ideal hair attributes (right column). Panels (a) and (b) show the marginal distributions for the similarity between pairs of participants based on the actual data and the column and row permutation null models. The shaded area indicates the part of the distribution that is statistically significant with $p < 0.05$ based on the column null model. Panels (c) to (f) show the NCP and CRP plots for the networks obtained from the similarity matrix by keeping only edges that are statistically significant with $p < 0.05$ based on the column null model (dashed, red) or row null model (dash-dotted, yellow).



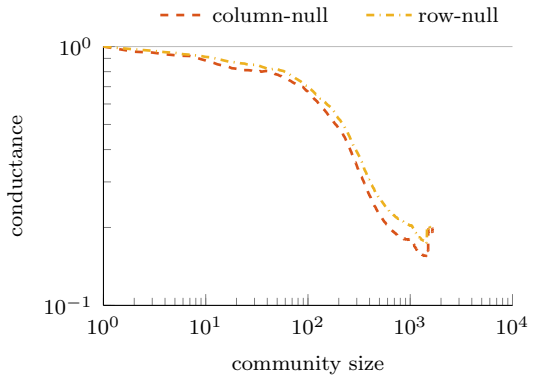
(a) Similarity distributions (current hair)



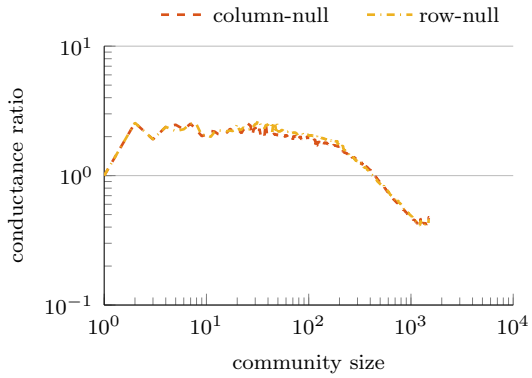
(b) Similarity distributions (ideal hair)



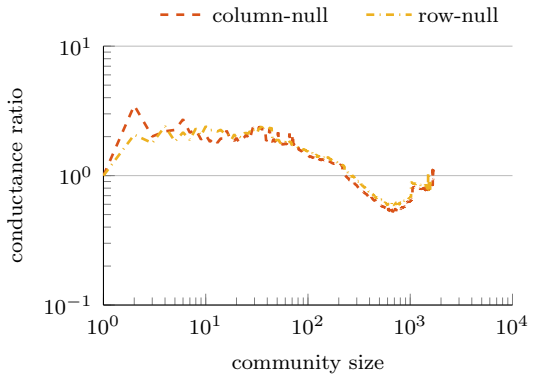
(c) NCPs (current hair)



(d) NCPs (ideal hair)

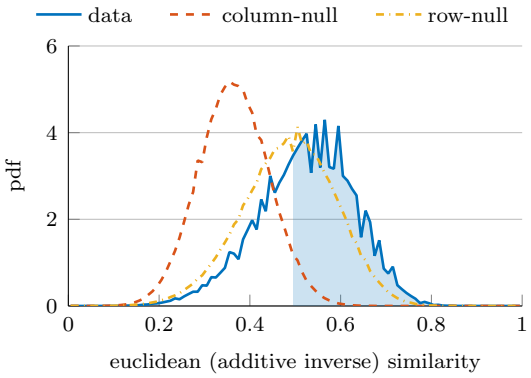


(e) CRPs (current hair)

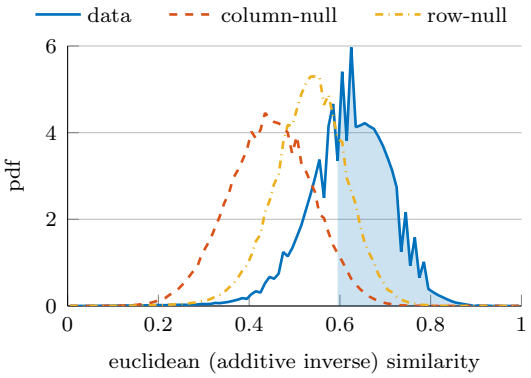


(f) CRPs (ideal hair)

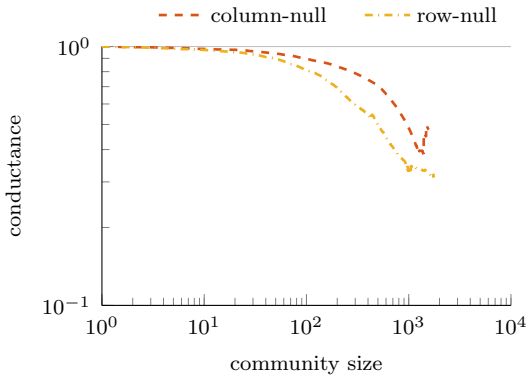
Figure 5.7: Overview of the structure of the networks generated using the *centred cosine similarity* for current hair attributes (left column) and ideal hair attributes (right column). Panels (a) and (b) show the marginal distributions for the similarity between pairs of participants based on the actual data and the column and row permutation null models. The shaded area indicates the part of the distribution that is statistically significant with $p < 0.05$ based on the column null model. Panels (c) to (f) show the NCP and CRP plots for the networks obtained from the similarity matrix by keeping only edges that are statistically significant with $p < 0.05$ based on the column null model (dashed, red) or row null model (dash-dotted, yellow).



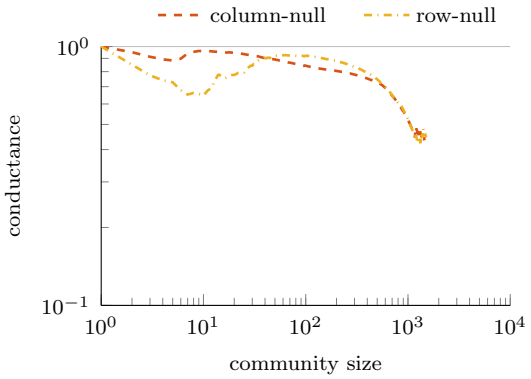
(a) Similarity distributions (current hair)



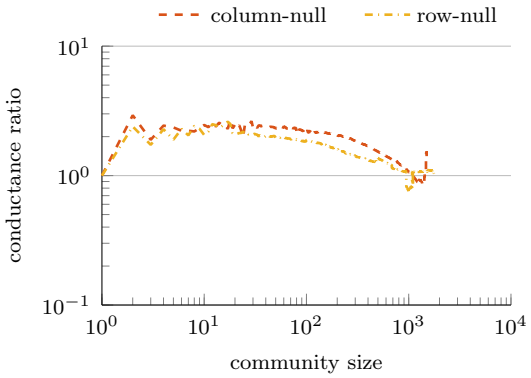
(b) Similarity distributions (ideal hair)



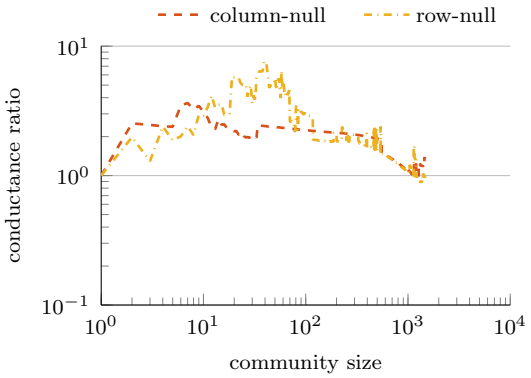
(c) NCPs (current hair)



(d) NCPs (ideal hair)

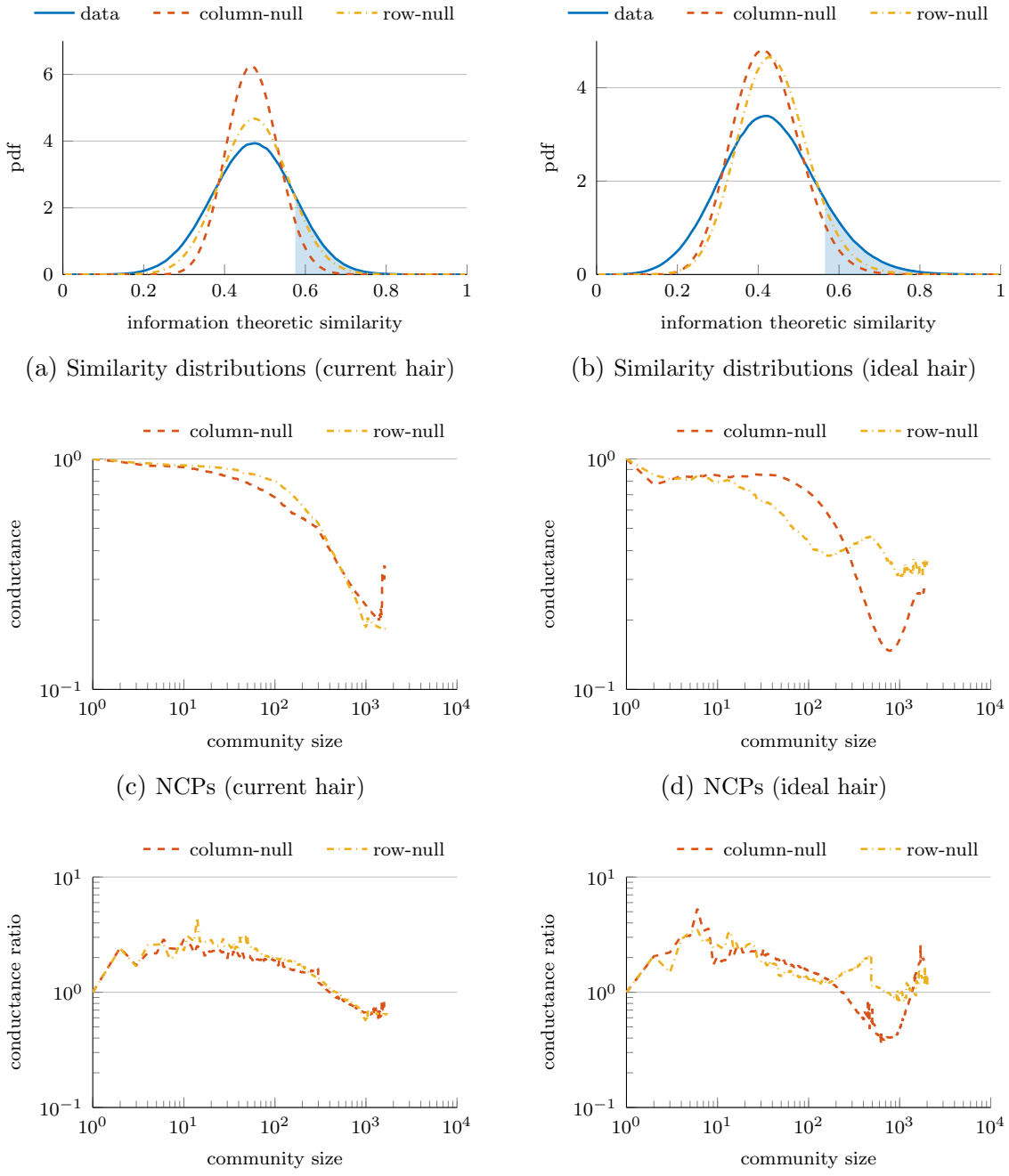


(e) CRPs (current hair)



(f) CRPs (ideal hair)

Figure 5.8: Overview of the structure of the networks generated using the *Additive inverse euclidean similarity* for current hair attributes (left column) and ideal hair attributes (right column). Panels (a) and (b) show the marginal distributions for the similarity between pairs of participants based on the actual data and the column and row permutation null models. The shaded area indicates the part of the distribution that is statistically significant with $p < 0.05$ based on the column null model. Panels (c) to (f) show the NCP and CRP plots for the networks obtained from the similarity matrix by keeping only edges that are statistically significant with $p < 0.05$ based on the column null model (dashed, red) or row null model (dash-dotted, yellow).



(a) Similarity distributions (current hair)

(b) Similarity distributions (ideal hair)

(c) NCPs (current hair)

(d) NCPs (ideal hair)

(e) CRPs (current hair)

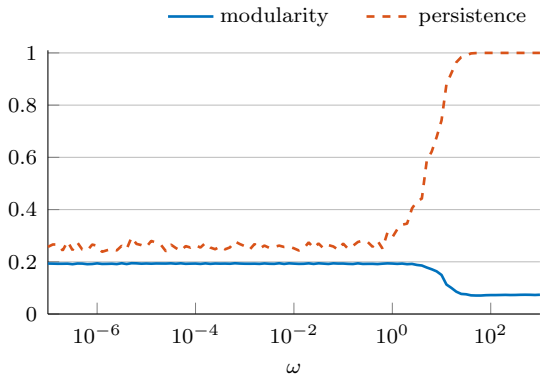
(f) CRPs (ideal hair)

Figure 5.9: Overview of the structure of the networks generated using the *information theoretic similarity* for current hair attributes (left column) and ideal hair attributes (right column). Panels (a) and (b) show the marginal distributions for the similarity between pairs of participants based on the actual data and the column and row permutation null models. The shaded area indicates the part of the distribution that is statistically significant with $p < 0.05$ based on the column null model. Panels (c) to (f) show the NCP and CRP plots for the networks obtained from the similarity matrix by keeping only edges that are statistically significant with $p < 0.05$ based on the column null model (dashed, red) or row null model (dash-dotted, yellow).

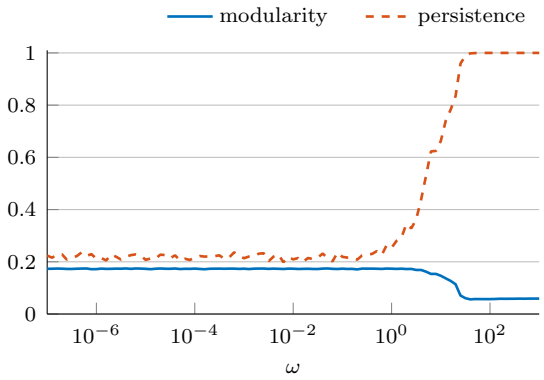
structure of these networks in more detail and to explore their community structure across different size scales. Comparing the network community profiles and conductance ratio profiles for the networks based on different similarity measures and null models (Figs. 5.6c–f, 5.7c–f, 5.8c–f, and 5.9c–f), we do not see any good small communities in these networks, with the exception of some extremely small communities in the case of the network generated using the cosine similarity and column null model (Figs. 5.6cd). Note that these networks also have many small components (see Tables 5.2 and 5.3), suggesting that these extremely small communities should be interpreted as another symptom of the tendency for uniform thresholding to fragment part of the network, especially as they are not present in the network when using the row null model to threshold the similarity matrix. For all of the networks we see some evidence of large-scale community structure at community sizes of around 1000 nodes per community, as indicated by a dip in the NCP and CRP. For most of the similarity measures, the NCPs for the two null models have rather different detailed structure with the exception of the centred cosine similarity, where the two NCPs are almost identical.

5.3.2 Multiplex Modularity

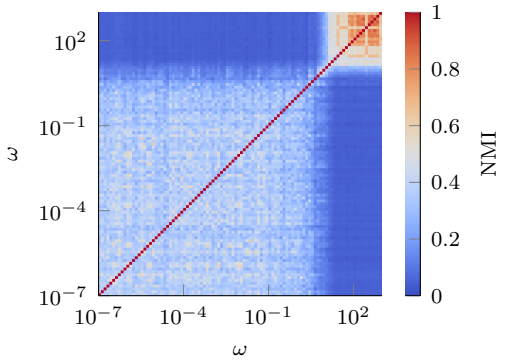
The results from the previous section suggest that there may be some large-scale, weak clusters in this data set for both current and ideal hair attributes. However, we do not see evidence of small-scale clusters based on the conductance measure. In this section we investigate the community structure in the full six-layer multiplex network using multiplex modularity (Eq. (2.35)). For layers three to six, where the underlying data is binary, indicating whether a participant agrees that a particular statement is a symptom or cause of dryness or damage, we use the Jaccard coefficient to measure similarity between participants. The Jaccard coefficient is a measure of similarity between sets, given by the ratio of the size of the intersection of two sets divided by the size of there union (see Eq. (5.15)). As for current and ideal hair attributes, we can use the row or column permutation null model to construct the



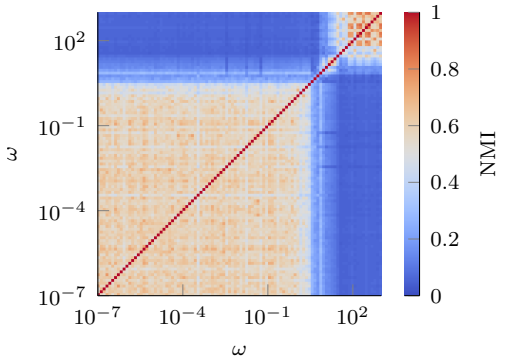
(a) column null model



(b) row null model



(c) NMI between multilayer partitions for different values of omega (column null model)



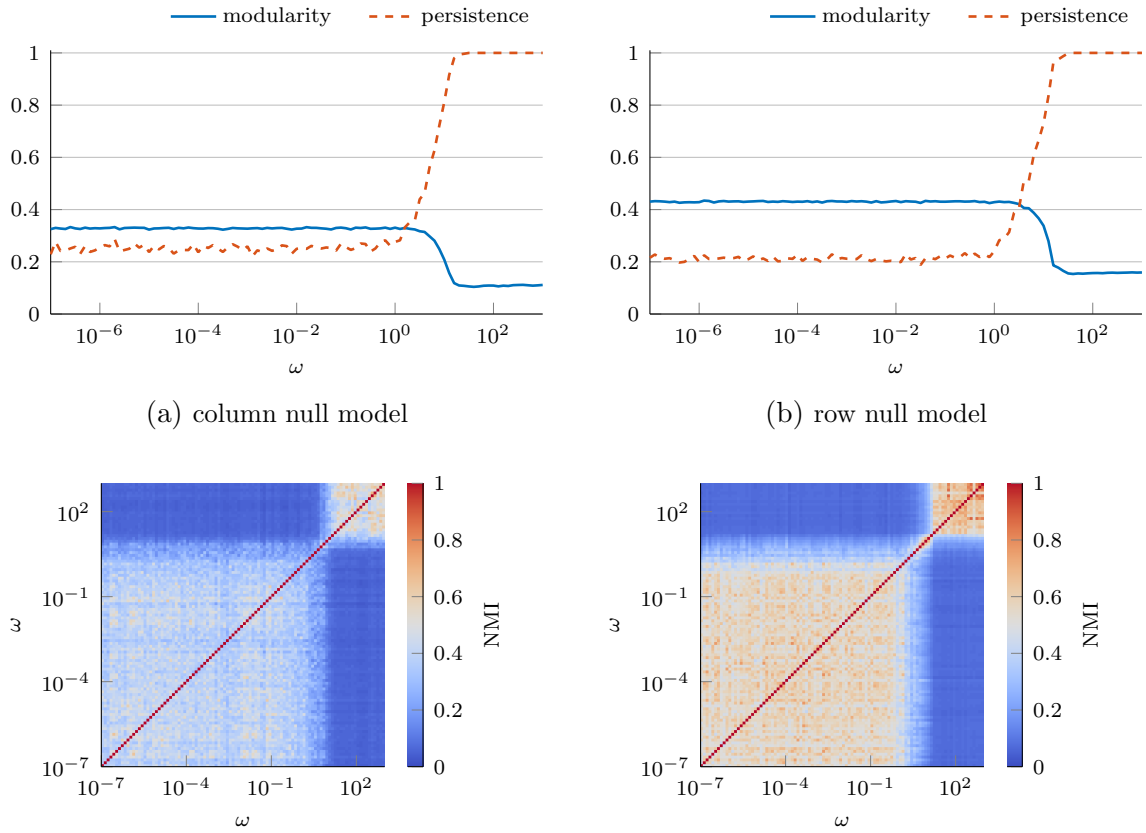
(d) NMI between multilayer partitions for different values of omega (row null model)

Figure 5.10: Cosine similarity

intralayer networks for these layers (see Section 5.2.3).

Recall from Section 2.5.1 that multilayer modularity rewards consistency between community structure in different layers of a multilayer network using a persistence contribution, where the strength of the persistence contribution is governed by a coupling parameter w. The persistence of a multilayer partition $\mathcal{S}$ ranges between 0 and 1, where persistence is equal to one if all state nodes representing the same physical node have the same community assignment.

For $\omega = 0$, optimising multiplex modularity is equivalent to optimising modularity separately for each layer. As we increase ω, the persistence contribution becomes increasingly important, enforcing consistency between the partitions in different layers. In the limit as $\omega \to \infty$, partitions only depend on the aggregate structure of the network.



(a) column null model

(b) row null model

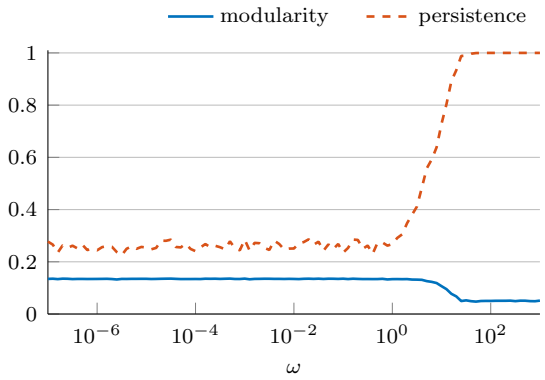
(c) NMI between multilayer partitions for different values of omega (column null model)

(d) NMI between multilayer partitions for different values of omega (row null model)

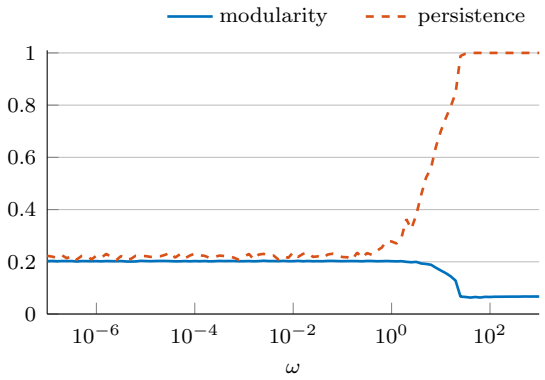
Figure 5.11: Centred cosine similarity

Figures 5.10–5.13 explore the behaviour of multiplex modularity as a function of ω for the networks based on different similarity measures for current and ideal hair attributes and thresholded using either the column or row null model. For each combination of similarity measure and null model we sample 101 multilayer partitions for different values of ω logarithmically spaced between 10^{-7} and 10^{3} using GenLouvainRand with multiplex postprocessing and restarting (see Appendix B).

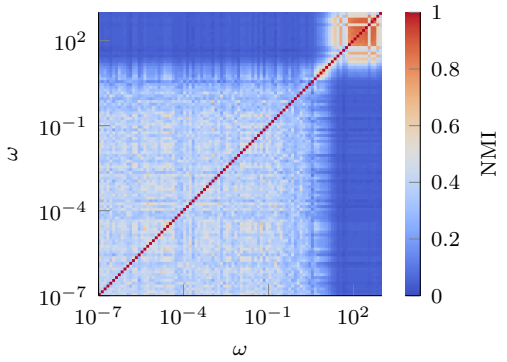
Panels (a) and (b) explore the relative importance of persistence and intralayer modularity contributions to the multiplex modularity as a function of ω. For each of the partitions we find, we plot the value of its multiplex modularity evaluated with $\omega = 0$ and its persistence. The behaviour of modularity and persistence contributions is consistent across similarity measures and null models. For $\omega \ll 1$ modularity and persistence contributions are approximately constant as a function of ω, and



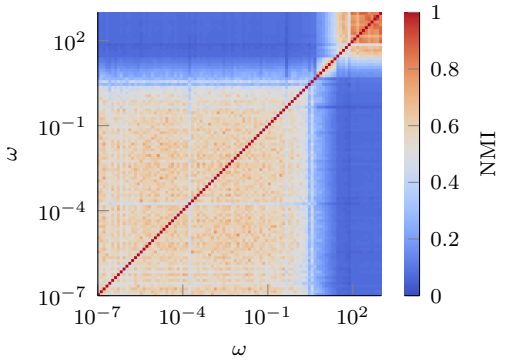
(a) column null model



(b) row null model



(c) NMI between multilayer partitions for different values of omega (column null model)

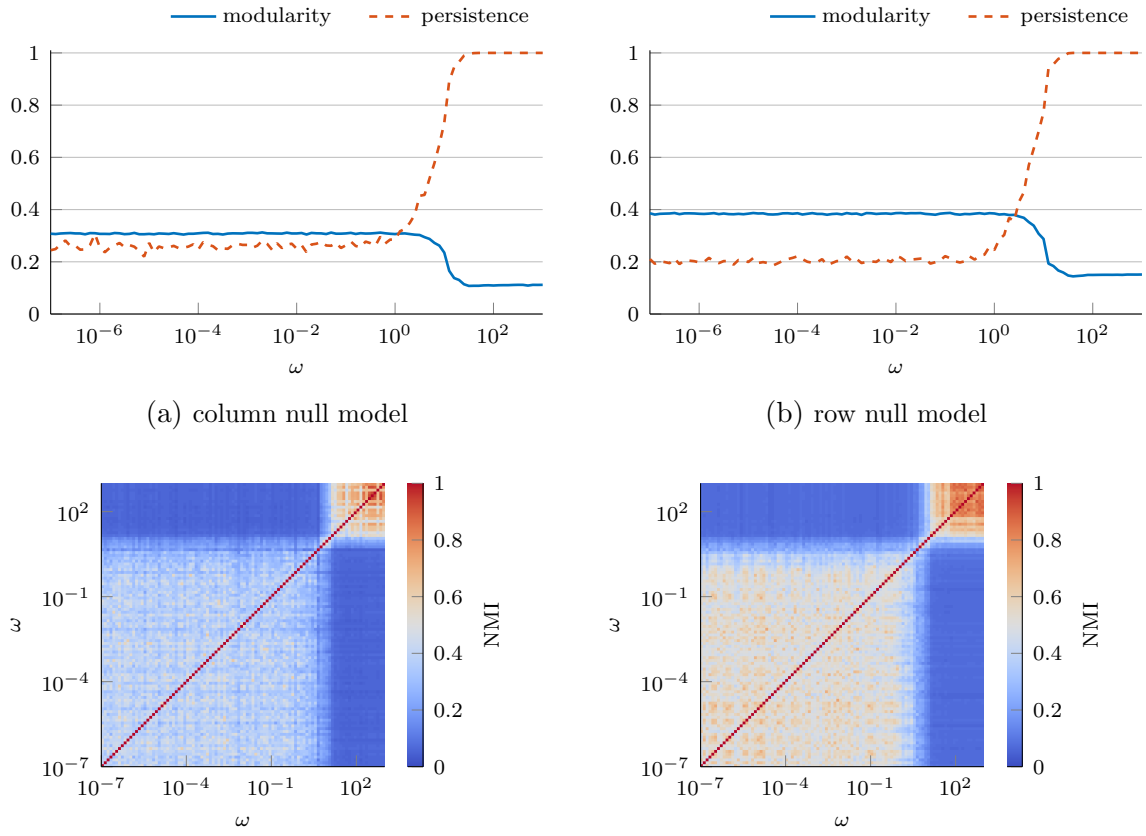


(d) NMI between multilayer partitions for different values of omega (row null model)

Figure 5.12: Euclidean (additive inverse) similarity

the persistence is fairly small, suggesting that the partitions we find are essentially equivalent to the results we would obtain by analysing each layer separately. For $\omega \gg 10$ modularity is again approximately constant and persistence is equal to one, indicating that we only see aggregate behaviour. The results for intermediate values of ω are of particular interest from a multilayer networks perspective, as it is here that we could potentially see non-trivial effects of the multiplex structure that are not detectable by either treating the multilayer network as l independent single layer networks or by aggregating all layers together.

In panels (c) and (d) we compare the partitions we find for different values of ω directly. We compute the normalised mutual information (NMI, Eq. (2.51)) for each pair of multilayer partitions. Recall that a value for the NMI close to one indicates similar partitions and a value for the NMI close to zero indicates partitions that are



(a) column null model

(b) row null model

(c) NMI between multilayer partitions for different values of omega (column null model)

(d) NMI between multilayer partitions for different values of omega (row null model)

Figure 5.13: Information theoretic similarity

independent. For all similarity measures and null models, the multilayer partitions form two clear blocks of similar partitions, one for partitions at small values of ω and one for partitions at large values of ω, with a small transition region in between, confirming the observations above.

Comparing Figs. 5.10–5.13, we see that the behaviour of multiplex modularity based community detection is very similar across networks generated using different similarity measures. To see whether the same is true as well for the partitions we find, we directly compare partitions for different similarity measures and null models in Fig. 5.14. We again use NMI to compare the multilayer partitions. The central matrix plot of Fig. 5.14 show the NMI between all pairs of partitions across different similarity measures and null models. To reveal the structure of the matrix, we sort the partitions based on the optimal leaf order of the average-linkage hierarchical clustering

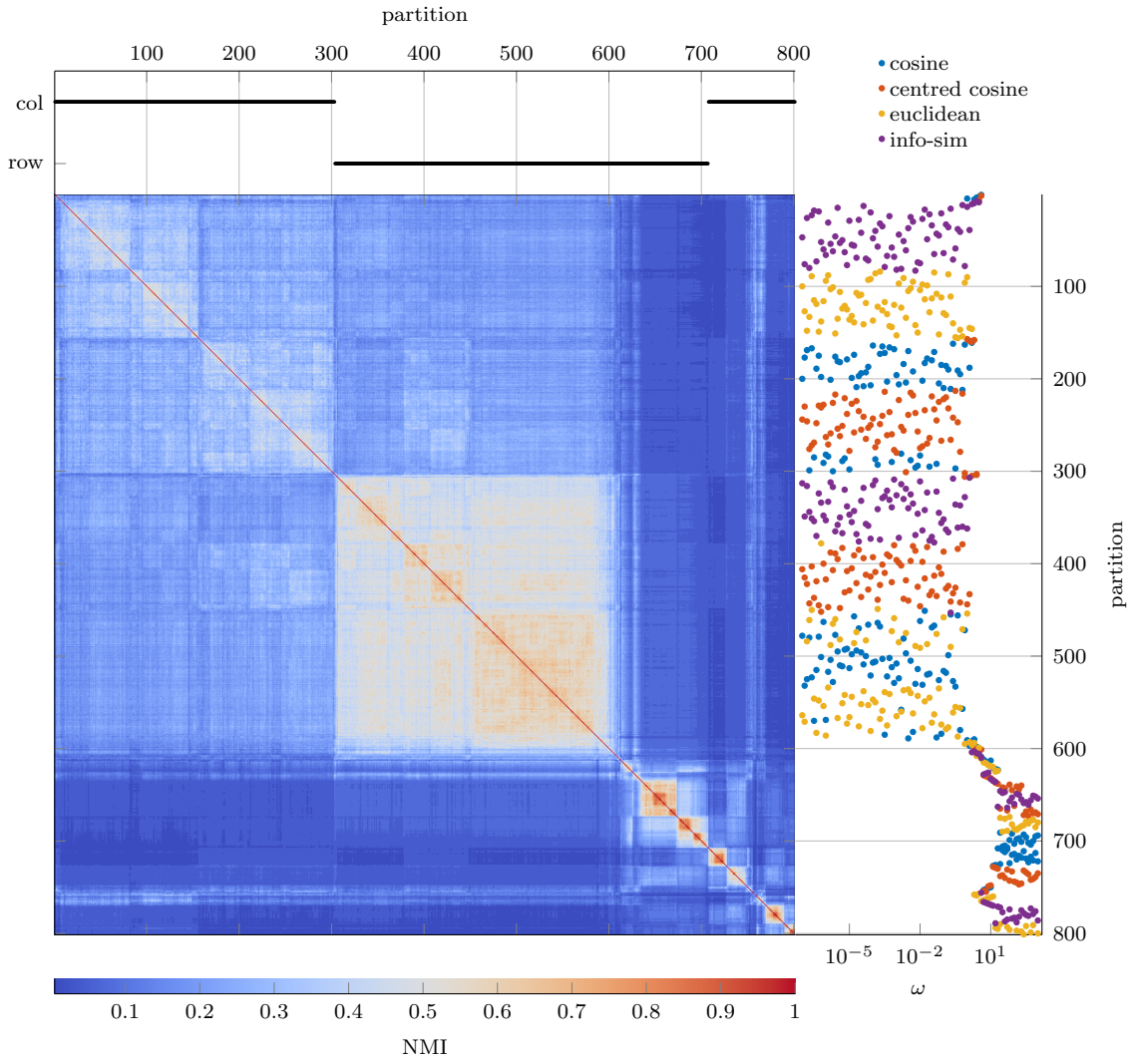


Figure 5.14: Comparison of multilayer partitions for networks generated using different similarity measures and null models. Each row and column of the main matrix plot corresponds to a multilayer partition. The scatter plot on the right shows the value of the interlayer coupling ω that was used to generate the partition (where there is one point for each row of the matrix), colour-coded to indicate the similarity measure that was used to generate the network. The plot above the matrix indicates the null model used to threshold the network for the corresponding partition.

tree of the matrix [13]. The plot on the right of the matrix shows the value of ω used to generate the partition corresponding to each row of the matrix, colour coded to also show the similarity measure used. The plot on top of the matrix indicates whether a partition corresponding to a particular column of the matrix was generated using the row or column null model. We again see a clear split between partitions for small and

large values of ω, with a transition region starting at $\omega \approx 1$ and ending at $\omega \approx 10$. The different null models and similarity measures do have a noticeable influence on the partitions we find. In particular, it appears that networks thresholded based on the row null model have clearer community structure which results in less variability between partitions from different runs of the community detection algorithm.

5.3.3 Consensus Community Structure

From the results in the previous section we have a large number of different partitions based on different similarity measures and different values of ω. Given that all of these partitions are coarse-grained representations of the same underlying data, we are particularly interested in their common features, as these should be more representative of the underlying data, rather than an artifact of a particular way to construct a network from this data. Consensus clustering is one way to identify such features. From the results in the previous section and in particular from Fig. 5.14, we see that we have a clear division between partitions for small values of ω and large values of ω, with a transition region for intermediate values of ω. To investigate this behaviour, we compute three separate consensus partitions, one for $\omega \in [10^{-7}, 1)$ which represents partitions where the different layers are essentially independent, one for $\omega \in [1, 10]$ where we see multilayer interactions, and one for $\omega \in (10, 10^3]$ where we see only aggregate structure. For each of the three sets of partitions, we compute an association matrix between all the state nodes of the multilayer network, where an entry of the association matrix gives the fraction of the sampled partitions in which two state nodes are grouped in the same community. From the association matrices we then identify consensus partitions using modularity maximisation with a null model that keeps the community sizes for each layer of the sampled partitions fixed but otherwise assumes that nodes are assigned to communities at random. We discuss this procedure in detail in Section 2.5.2. As before we use GenLouvainRand to optimise modularity. However there is no need for postprocessing at this stage. Different runs of the algorithm produce almost identical partitions and we use the

partition with the highest modularity based on ten runs of GenLouvainRand as a representative partition.

Given a representative partition of the multilayer network, we still need a way to interpret the community structure to identify what, if anything, it reveals about patterns in the data. From the original survey data, we have a large number of potential variables that could help explain the community structure, only a subset of which we use to generate the different layers of the multilayer network. In total we have 137 potential explanatory variables (396 after expanding categorical variables using dummy variables), some of which have categorical responses and some of which use an ordinal response scale. Given the large number of potential explanatory variables, we need a method that can identify only the most relevant variables, takes correlations between variables into account, and can handle both categorical and ordinal responses. The idea behind the method we propose here is that we can treat community assignments as a coarse-grained observable variable and try to predict community membership based on the different potential explanatory variables. A standard approach to this type of problem, where the dependent variable to be explained is categorical, is to use multinomial logistic regression. One way to interpret the model underlying multinomial logistic regression is as a latent variable model, where each community has a different associated latent variable which is a linear combination of the observed variables [139]. The probability under the model of being assigned to a particular community then depends on the relative sizes of the associated latent variables. Formally, given an $n \times p$ data matrix $\boldsymbol{X}$ of explanatory variables, we assume that the value of the latent variable associated with a community k and a participant i has the form

$$\tilde{\boldsymbol{Y}}_{ik} = \boldsymbol{a}_k + \boldsymbol{b}_{k\cdot}\boldsymbol{X}_{i\cdot}^T + \boldsymbol{\epsilon}_k, \tag{5.18}$$

where c is the number of observed communities, $\boldsymbol{a}$ is a $c \times 1$ vector of intercepts and $\boldsymbol{b}$ is a $c \times p$ matrix of coefficients that we estimate and the error variables $\boldsymbol{\epsilon}_k$ follow a standard type-1 extreme value distribution. The observed community assignment $\boldsymbol{Y}_i$ of participant i is that with the largest associated latent variable, thus the probability

under the model of observing that participant i is assigned to community k is

$$\mathbb{P}\left[\boldsymbol{Y}_i = k\right] = \mathbb{P}\left[\max\left(\tilde{\boldsymbol{Y}}_{i1}, \ldots, \tilde{\boldsymbol{Y}}_{ic}\right) = \tilde{\boldsymbol{Y}}_{ik}\right]. \tag{5.19}$$

One can write down the probabilities for community assignments in closed form (see [139]) as

$$\mathbb{P}[\boldsymbol{Y}_i = k|\boldsymbol{X}_{i\cdot}] = \frac{e^{\boldsymbol{a}_k + \boldsymbol{b}_{k\cdot}\boldsymbol{X}_{i\cdot}^T}}{\sum_{l=1}^{c} e^{\boldsymbol{a}_l + \boldsymbol{b}_{l\cdot}\boldsymbol{X}_{i\cdot}^T}}. \tag{5.20}$$

From Eq. (5.20) one could estimate the coefficients using maximum likelihood. However, unless we add additional constraints, the maximum likelihood problem specified by Eq. (5.20) does not have a unique solution as the probabilities in Eq. (5.19) only depend on the differences between the latent variables, and we could hence shift all coefficients for each variable by a constant without affecting the probabilities. More importantly, a maximum likelihood solution will in general have non-zero coefficients for all explanatory variables which would make the results difficult to interpret given the number of variables we are considering.

One way to identify sparse solutions where most of the coefficients are zeros is to use elastic net regularisation [63, 214] to select only the most relevant variables for each community. To select only the most relevant variables, the algorithm maximises the likelihood of the model subject to a regularisation penalty on the coefficients, i.e. it solves

$$\max_{\boldsymbol{a},\boldsymbol{b}} \left[\frac{1}{n}\sum_{i=1}^{n} \mathbb{P}[\boldsymbol{Y}_i|\boldsymbol{X}_{i\cdot}] - \lambda \sum_{k=1}^{c} \frac{1-\alpha}{2}\|\boldsymbol{b}_{k\cdot}\|_2^2 + \alpha\|\boldsymbol{b}_{k\cdot}\|_1\right] \tag{5.21}$$

The algorithm in [63] allows one to quickly find the entire regularisation paths, i.e. coefficients for "all" sensible values of λ. The parameter α interpolates between an l_2 and an l_1 penalty and controls the sparsity of the solution, where values of α close to one give sparser solutions. For the calculations on the survey data we use $\alpha = 1$ which corresponds to LASSO regression [195]. Note that from a Bayesian perspective, the l_1 penalty is equivalent to assuming independent double-exponential priors for the coefficients, whereas the l_2 penalty corresponds to independent normal priors. One of the main reasons for using an l_1 penalty to introduce sparsity rather

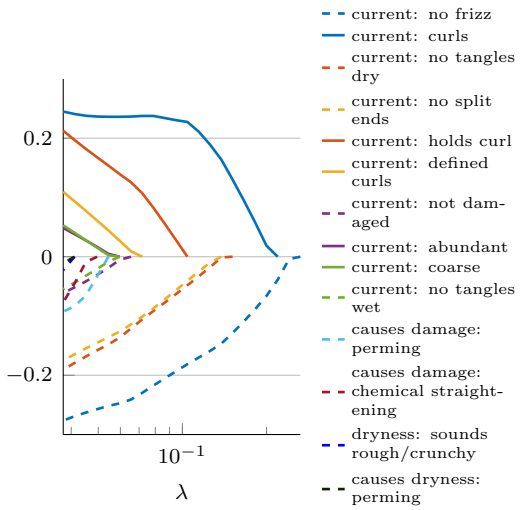
than a more direct way, e.g., by constraining the number of non-zero coefficients, is that it is a convex constraint, which has major computational advantages. That the l_1 penalty produces sparse solutions is a direct consequence of the geometry of the unit ball under the l_1 norm, which has sharp corners at the coordinate axes (see Fig. 2 of [195]).

Figures 5.15–5.20 show the coefficients for the multinomial logistic model as a function of the LASSO regularisation parameter λ for the consensus partition based on intermediate values of ω, where we expect to see multilayer effects. Note that in the limit as $\lambda \to 0$, LASSO regression tends to ordinary logistic regression and all coefficients are non-zero. As we increase λ, more and more coefficients are driven to zero, and for sufficiently large λ all coefficients are zero. The idea behind the LASSO regression is that the variables that are most important for explaining the observed behaviour should still have non-zero coefficients for relatively large λ whereas coefficients for unimportant variables are driven to zero quickly. For visualisation purposes, we truncate the plots for each community in Figs. 5.15–5.20, such that we have no more than 15 non-zero coefficients for each community.

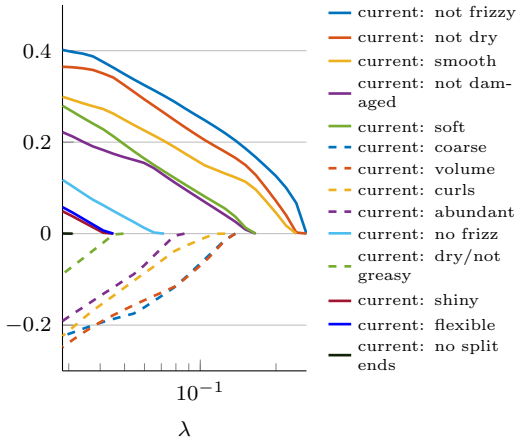
To assess how closely the multinomial regression model explains the community structure we find, note that once we have fitted the model using our observed community structure, we can in turn use the model to predict community assignments for each node, by assigning a node to the community which has the highest probability in Eq. (5.20). To assess how well the model recovers a particular community, we use the Jaccard coefficient J of the community c in the observed partition $\boldsymbol{Y}$ and the predicted partition $\tilde{\boldsymbol{Y}}$, where

$$J(c) = \frac{\sum_{i=1}^{n} \delta(\boldsymbol{Y}_i, c)\delta(\tilde{\boldsymbol{Y}}_i, c)}{\sum_{i=1}^{n} \max\left(\delta(\boldsymbol{Y}_i, c), \delta(\tilde{\boldsymbol{Y}}_i, c)\right)}. \tag{5.22}$$

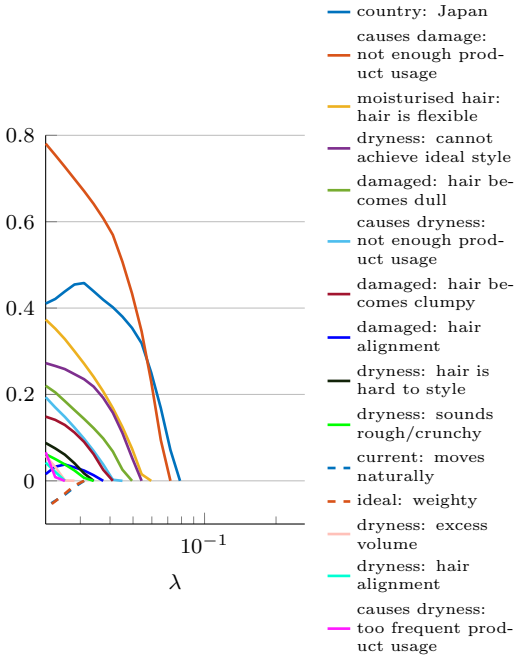
For the larger communities, 15 explanatory variables tend to be enough to be able to recover them reasonably closely, however for smaller communities we seem to need more variables (see Fig. 5.15). For very small communities, we do not have enough



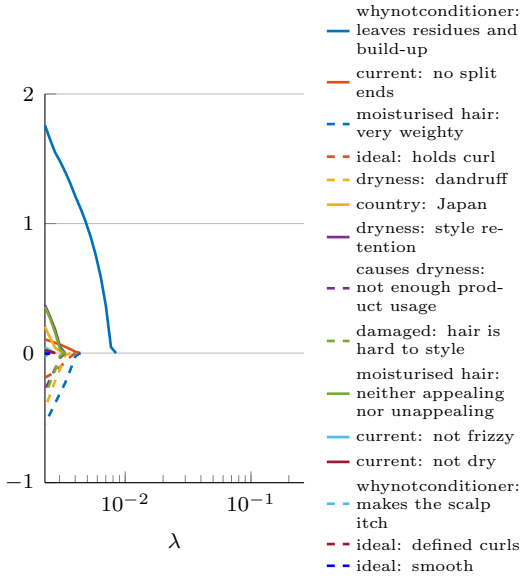
(a) Coefficients for Community 1 (size: 1146, recovered: 0.79 (0.92))



(b) Coefficients for Community 2 (size: 1213, recovered: 0.87 (0.97))

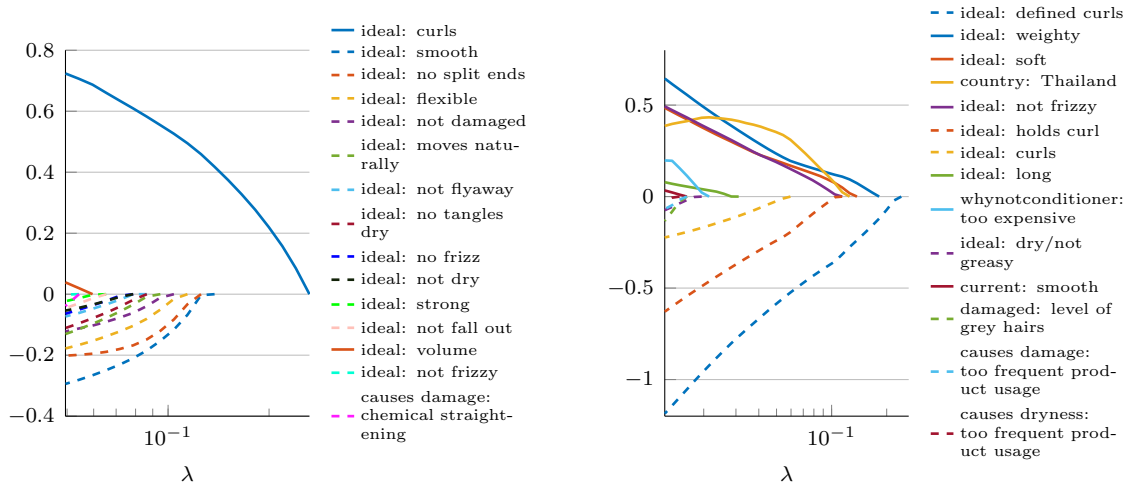


(c) Coefficients for Community 3 (size: 304, recovered: 0.35 (0.72))



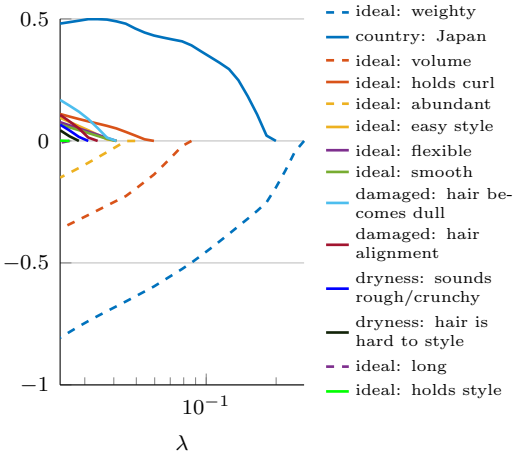
(d) Coefficients for Community 4 (size: 20, recovered: 0.10 (1.00))

Figure 5.15: Multinomial LASSO coefficients for current hair attributes. For each community we indicate its size and how well the model recovers the community using the Jaccard coefficient between the actual community and the predicted community based on the model for the smallest value of λ shown in the plot and the full model (in brackets).



(a) Coefficients for Community 1 (size: 818, recovered: 0.72 (0.93))

(b) Coefficients for Community 2 (size: 1164, recovered: 0.89 (0.96))

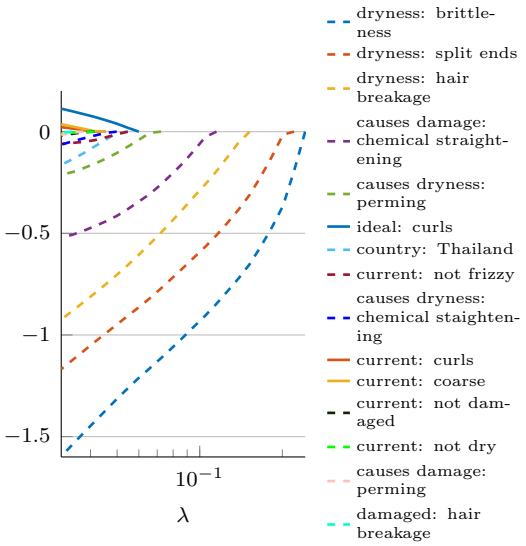


(c) Coefficients for Community 3 (size: 702, recovered: 0.81 (0.97))

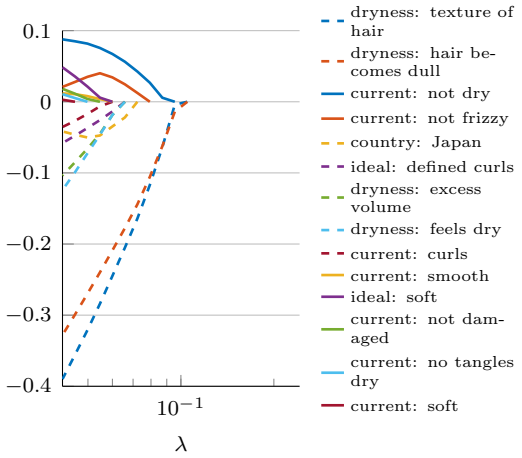
Figure 5.16: Multinomial LASSO coefficients for ideal hair attributes. For each community we indicate its size and how well the model recovers the community using the Jaccard coefficient between the actual community and the predicted community based on the model for the smallest value of λ shown in the plot and the full model (in brackets).

samples to properly estimate the coefficients and to avoid numerical issues with the optimisation algorithm we have excluded any community with less than 20 participants from the analysis. Note that we mostly find large communities with on the order of 1000 nodes which agrees with the observations based on NCPs for current and ideal hair attributes in Section 5.3.1.

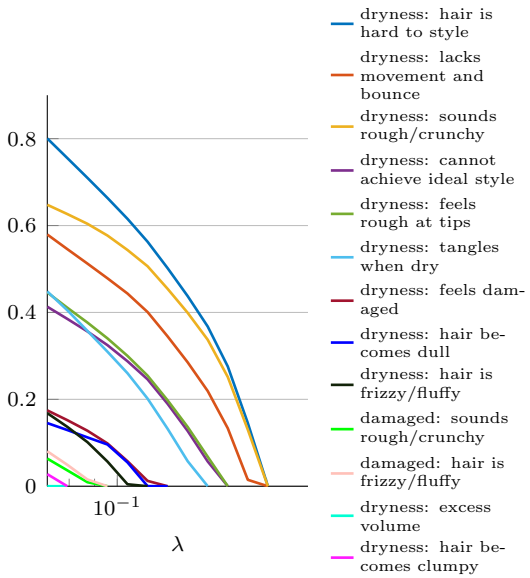
To interpret the values of the coefficients, it is perhaps most straight forward to



(a) Coefficients for Community 1 (size: 924, recovered: 0.70 (0.78))



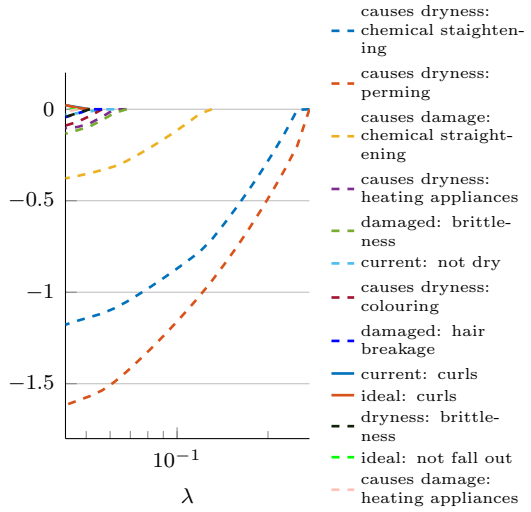
(b) Coefficients for Community 2 (size: 1075, recovered: 0.68 (0.81))



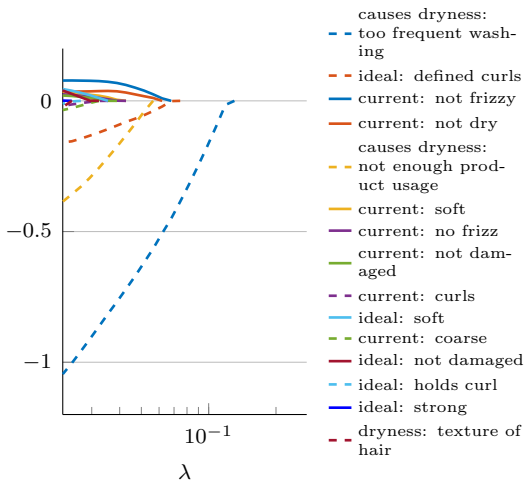
(c) Coefficients for Community 3 (size: 680, recovered: 0.68 (0.92))

Figure 5.17: Multinomial LASSO coefficients for symptoms of dryness. For each community we indicate its size and how well the model recovers the community using the Jaccard coefficient between the actual community and the predicted community based on the model for the smallest value of λ shown in the plot and the full model (in brackets).

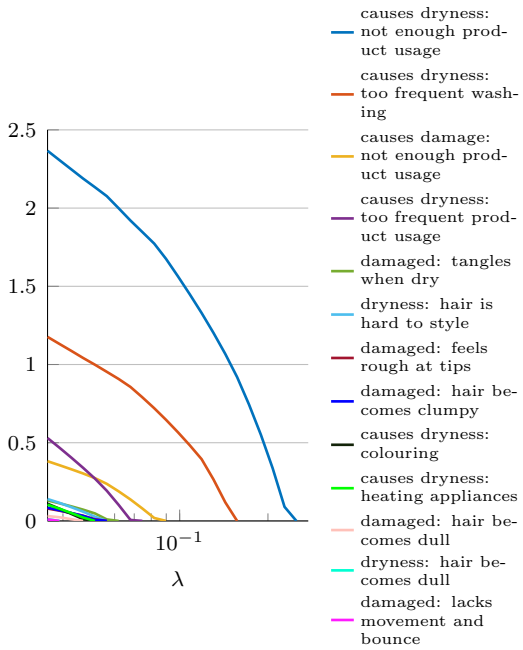
consider the way they influence the latent variables (see Eq. (5.18)) underlying the logistic model. The latent variables are composed of a linear function of the data



(a) Coefficients for Community 1 (size: 1132, recovered: 0.66 (0.77))

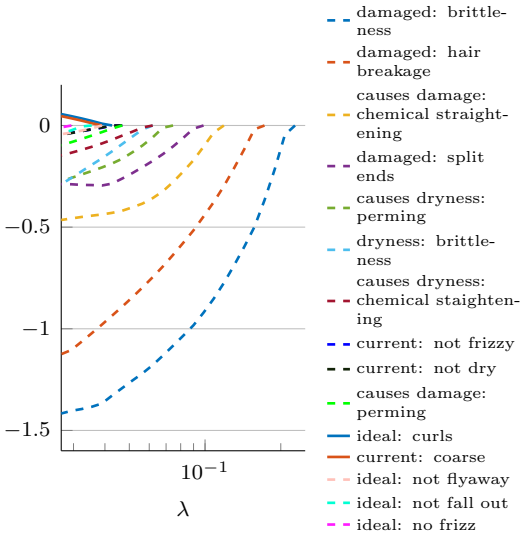


(b) Coefficients for Community 2 (size: 872, recovered: 0.64 (0.70))

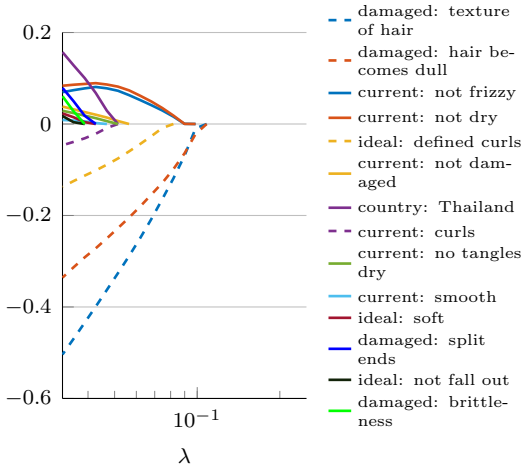


(c) Coefficients for Community 3 (size: 670, recovered: 0.71 (0.79))

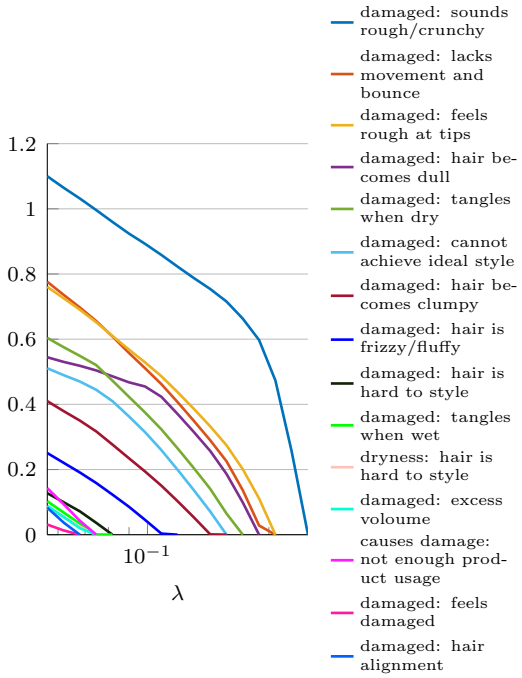
Figure 5.18: Multinomial LASSO coefficients for causes of dryness. For each community we indicate its size and how well the model recovers the community using the Jaccard coefficient between the actual community and the predicted community based on the model for the smallest value of λ shown in the plot and the full model (in brackets).



(a) Coefficients for Community 1 (size: 703, recovered: 0.62 (0.74))

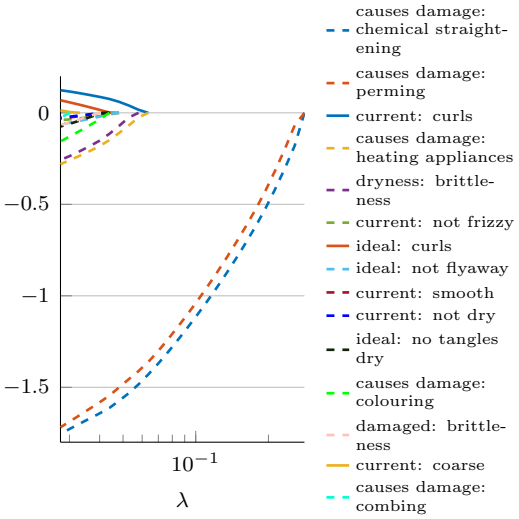


(b) Coefficients for Community 2 (size: 1208, recovered: 0.75 (0.84))

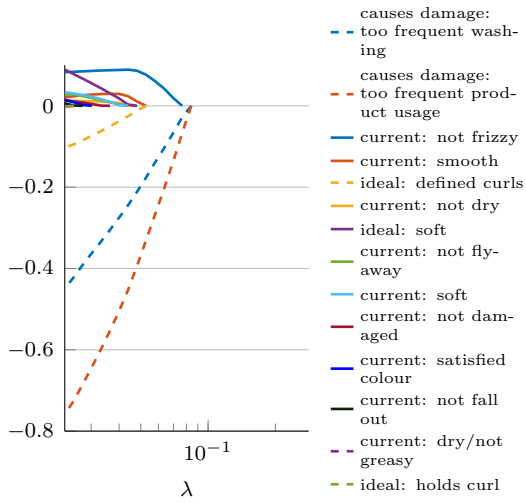


(c) Coefficients for Community 3 (size: 757, recovered: 0.76 (0.94))

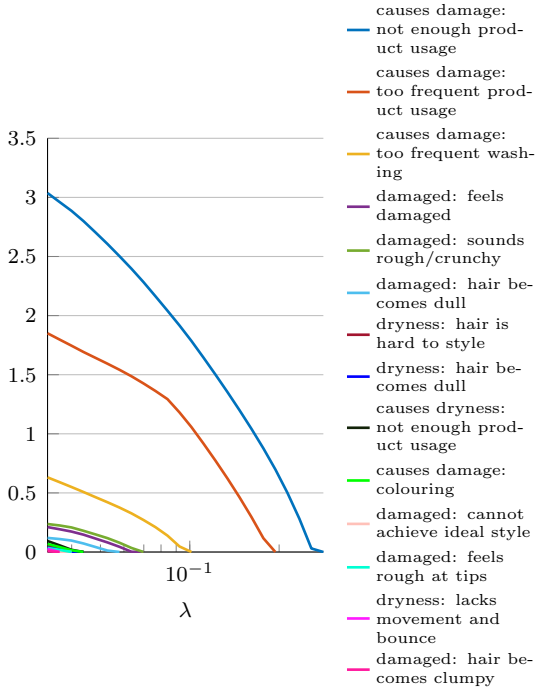
Figure 5.19: Multinomial LASSO coefficients for symptoms of damage. For each community we indicate its size and how well the model recovers the community using the Jaccard coefficient between the actual community and the predicted community based on the model for the smallest value of λ shown in the plot and the full model (in brackets).



(a) Coefficients for Community 1 (size: 1001, recovered: 0.69 (0.77))



(b) Coefficients for Community 2 (size: 905, recovered: 0.64 (0.73))



(c) Coefficients for Community 3 (size: 772, recovered: 0.78 (0.87))

Figure 5.20: Multinomial LASSO coefficients for causes of damage. For each community we indicate its size and how well the model recovers the community using the Jaccard coefficient between the actual community and the predicted community based on the model for the smallest value of λ shown in the plot and the full model (in brackets).

(determined by the coefficients) and an error term. A large positive coefficient for a particular explanatory variable and community means that, everything else held constant, a participant with a larger value for this explanatory variable is more likely to be assigned to this community than a second participant with a smaller value, as the value for the latent variable for this community will on average be larger for the first participant and a participant is assigned to the community with largest associated latent variable (see Eq. (5.19)).

5.3.4 Conclusions

The behaviour of the coefficients for the different communities in the different layers suggest some intuitive aspects of peoples perception of their hair. Curly versus straight hair appears to be a main factor for both current (Fig. 5.15) and ideal (Fig. 5.16) hair attributes. Interestingly, it appears that curly hair is perceived as being more damaged, and at least some of the participants that want curly hair accept this as an unavoidable trade-off. There are also hints of differences in culture between the different countries in the data set. In particular Thailand and Japan stand out as being different from the other countries (Brazil, UK, US) but also from each other. This stands out in particular for the ideal hair attributes (Fig. 5.16), where it appears that participants from Japan have a strong dislike for weighty hair, compared to the other countries, whereas participants from Thailand appear to have a preference for weighty, straight hair. It also appears that Japanese participants put particular importance on having hair that is easy to style. There also appears to be a difference in perception of hair dryness and hair damage between participants with curly and participants with straight hair. One of the differences appears to be that participants that have or want curly hair are less likely to think that methods to straighten hair cause damage, though there appear to be other factors involved as well. The third communities in the network layers related to dryness and damage (Figs. 5.17–5.20) seems to be mostly participants that agree with a large fraction of the provided answers.

Overall, this analysis suggests some interesting hypothesis about peoples perception of their hair that might be worth investigating in more detail. There are also other interesting features of the multinomial logistic regression model that we have not explored. The model not only gives a discrete community assignment for each participant, but rather a probability distribution over community assignments for the participant. Thus, for each participant we have a notion as to how certain we are about their community assignment based on the model and it would be interesting to see how this compares to the variability we see between different runs of the community detection algorithm.

The general approach we use in this section of using a classification algorithm to interpret the results of a clustering algorithm should be applicable to any data set where we have a reasonably large number of potential explanatory variables but no clear dependent variable. There are a large number of possible algorithms for both the clustering and classification part of this procedure [85]. One of the key advantages of using a multilayer networks based approach for the clustering algorithm is that it allows us to investigate the data at a finer level of aggregation (the layers of the multilayer network) while still exploiting correlations between the layers. With most traditional clustering algorithms we would either have to aggregate the layers or analyse each layer independently. Also, modularity maximisation has been shown to be effective at identifying clusters that pose difficulties for many traditional clustering algorithms [47, 82, 89]. Multinomial logistic regression is reasonably straight forward to fit and produces results that are easily interpretable, which is particularly important in this application. However it has some key limitations to keep in mind. Notably it is a linear classifier (see [85]), whereas the clusters one identifies using modularity maximisation are in general not linearly separable. For the hair survey data we discuss in this chapter, multinomial logistic regression is able to recover the input partition reasonably well. However, this might not be the case in other applications, in which case one would need to use a non-linear classifier to obtain meaningful results.

Chapter 6

Conclusions and Future Work

A key topic of this thesis is the relationship between community structure in networks and the behaviour of dynamical processes on networks. In Chapter 3 we showed, using random walks and geodesic spreading dynamics as examples, that identifying bottlenecks of locally biased dynamical processes provides detailed insights into the mesoscale properties of large networks. We used the network community profile (NCP) introduced in [127, 128] as our main tool for summarising the behaviour of dynamical processes starting from different initial conditions (i.e., seed nodes). We showed that the qualitative features of an NCP for a given network are very robust to changes in the way one samples communities. Even using rough proxies for communities, such as k-neighbourhoods of different seed nodes (see Section 3.4.4), allows one to recover the qualitative trends of an NCP for a network. By analysing three pairs of networks of different types, we also showed that the qualitative features of an NCP reveal detailed information about the mesoscale organisation of different types of networks. In particular, the way local communities for different seed nodes combine to form larger scale structures is very different in different types of networks, whereas it is similar in networks of the same type. From the shape of their NCP and based on analysing the way local communities for different seed nodes combine more directly using association matrices, we found core-periphery structure with embedded small communities for co-authorship networks, mostly expander-like structure with some fuzzy large communities for Facebook networks and low-dimensional structure

for temporally-embedded multilayer voting networks.

The qualitative shape of an NCP of a network is intimately connected to the type of metric space a network embeds well in. This observation is directly related to the analogy between conductance and a surface area to volume ratio (see Section 3.2.1). Note that minimising conductance is analogous to minimising surface area while maximising volume and hence good communities are analogous to spheres in Euclidean space. For example networks with a downward sloping NCP tend to embed well in a low-dimensional Euclidean space. This can easily be made precise in the case of low dimensional lattices [127] and still approximately holds for other, less regular networks. Similarly, networks with a flat NCP should resemble high dimensional spaces, whereas networks with upward-sloping NCPs should resemble hyperbolic spaces [2, 29, 33, 96]. We hope to explore this connection between the shape of a networks NCP and metric embedding in more detail in the future as it could potentially help with designing benchmark networks with community structure that have NCPs that reproduce some of the properties of NCPs in real networks. This is a shortcoming of existing popular benchmark networks (see Section 3.5) which have expander-like communities with expander-like connections between communities which results in roughly flat NCPs that do not reflect the more intricate structure we observe in real networks.

Our local perspective of communities as bottlenecks to locally-biased dynamical processes also extents naturally to multilayer networks (see Section 3.6) which provide a framework for representing many types of data that cannot be adequately represented using a single-layer network. Community detection has recently received increasing attention and two of the most popular methods for community detection in single-layer networks, modularity optimisation and InfoMap have been generalised for multilayer networks [48, 149]. An interesting aspect of multilayer networks is that there are many different ways to generalise the unbiased random walk (which has been used as a parsimonious model for information diffusion motivating many community detection methods on single-layer networks including modularity optimisation [109–111] and InfoMap [176, 177]) and multilayer modularity and multilayer

InfoMap used different random walks to motivate the community quality function. As we show in Section 3.6 on the example of two small multiplex networks, these different definitions of a random walk result in very different notions of what considers a good multilayer community. This illustrates one of the key advantages of the local perspective we adopt in this thesis, as it allows us to compare the behaviour of different random walk dynamics directly and in a size-resolved manner.

Community detection is inherently an ill-defined problem [61, 167] and our results in Section 3.6 suggests that there is even more potential for disagreement as to what kind of structure should constitute a good community in multilayer networks. As a result, benchmark networks with known community structure are an invaluable tool for assessing whether a given community detection method can identify the type of structure one is interested in. For single-layer networks different types of benchmark networks have been developed (see Section 2.5.4) but for general multilayer networks such tools do not yet exist. The key difficulty when developing benchmark networks for multilayer networks is in sampling random partitions that reflect the desired interlayer correlations. We propose a method to sample such partitions in Chapter 4 and used it to generalise the popular LFR benchmarks [114, 118] to a multilayer setting. The modular nature of our approach allows us to easily generalise other planted partition models for single-layer networks to a multilayer setting as well. We used the LFR-like networks to compare the performance of multilayer modularity optimisation and multilayer InfoMap. Our results suggest that multilayer modularity optimisation is often able to exploit correlations between layers to identify the planted partition in situations where one cannot easily identify it using only information from individual layers. Multilayer InfoMap however does not seem to be able to exploit correlations in this way in our experiments. In the future, we plan to use this way of generating correlated partitions to develop benchmark networks with more complex structure for the intralayer connections. We also plan a more comprehensive study of different methods for identifying multilayer communities.

In Chapter 5 we used the local community detection methods from Chapter 3 and multilayer modularity optimisation to analyse multilayer similarity networks we

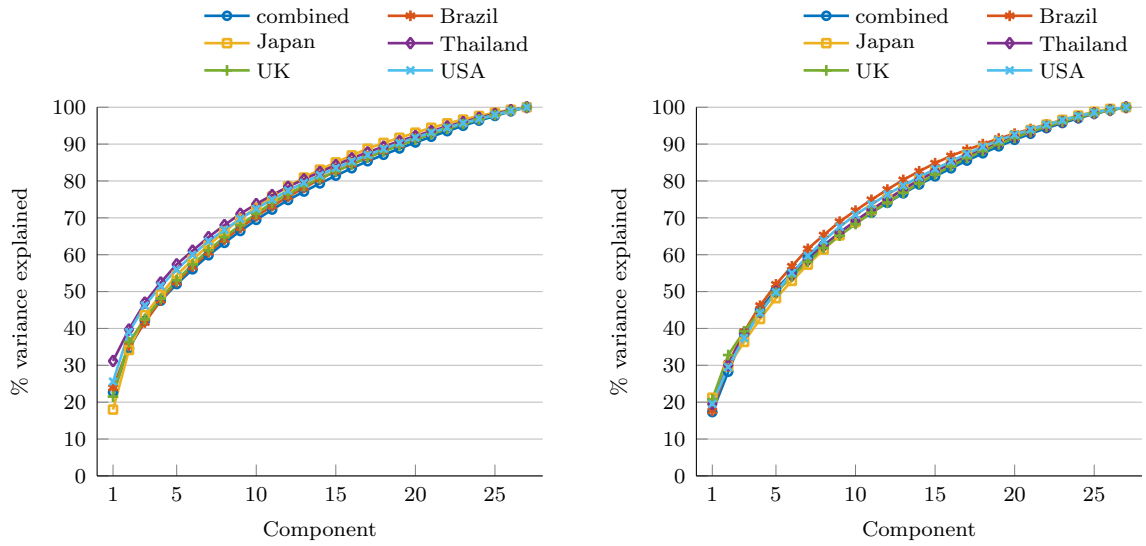
construct from survey data about people's perception of their hair. While at first glance using multilayer networks to represent this type of data is not the most natural choice, it allows us to represent the data at a relatively fine level of aggregation while still exploiting correlations between different aspects of the data. A key part of the survey data take the form of ordinal response scales and there is an ongoing, heated debate as to what kind of methods are appropriate for this type of data (see Section 5.2.1). We compare networks generated using different similarity measures in Section 5.3.1, though we conclude that, from the perspective of community structure in the resulting networks, the different similarity measures give reasonably similar results. Instead of relying on results based on a single similarity measure, we used a consensus clustering approach to identify common features of the results based on different similarity measures. One key advantage of this data set compared to most traditional network data is the large number of potential explanatory variables we have available to interpret our community detection results. However, this also means that we need a good method to select only the most relevant variables to obtain interpretable results. The approach we propose in Chapter 5 relies on the observation that one can treat the problem of community assignment as a classification task, where we want to predict the community assignment of a node based on the available explanatory variables. This is a very common type of problem and there are many possible ways to approach it [85]. Here we used multinomial logistic regression with l_1 regularisation to select only the most relevant variables as its output is particularly easy to interpret. This analysis reveals some interesting aspects of the data set. Some are not very surprising, but others are maybe less obvious. For example there appears to be a difference in perception of what constitutes damaged hair between people with curly and people with straight hair. There are also clear hints of cultural differences between the different countries in the data set. Overall, this approach suggests several interesting hypothesis about people perception of their hair (see Section 5.3.4), highlighting its potential use as a tool for exploratory analysis. In general, this type of approach should be particularly useful in situations where one has relatively many variables but no clear dependent variable. There are

some key aspects of the logistic model which we have not explored so far. First, it does not just return a community assignment for each node, but rather a probability distribution over communities for each node. This effectively defines an overlapping partition and it would be interesting to see whether nodes where the logistic model indicates a high uncertainty about their community assignment are also the nodes that show the most variability in their assignment between different runs of the community detection algorithm. Furthermore, it should be possible to fit a similar model directly to a sample of partitions, rather than just a single partition, which would be a different way to obtain a consensus structure than the co-classification approach we used in Section 5.3.3.

There are still a large number of open questions about mesoscale structure in large networks and the influence it has on the behaviour of dynamical processes that take place on these networks. In this thesis we have mostly considered random walks and related dynamics as a proxy for information diffusion on real networks. However, actual information diffusion on a social network for example is potentially much more intricate. Combining a local approach to analysing mesoscopic structure with more realistic dynamical processes should provide valuable insights into how people actually experience a network. Furthermore, real networks evolve over time and often comprise different types of connections. While we can conveniently represent these types of networks by using the framework of multilayer networks, the way these different aspects should interact and influence dynamical processes taking place on a multilayer network is not currently understood. As we have seen based on the example of two definitions of a random walk on a multilayer network, differences in the way a dynamical process traverses the interlayer structure of a multilayer network can lead to vastly different conclusions about its mesoscale structure and a local perspective is ideally suited to tease out these differences.

Appendix A

Country-by-Country PCA for Survey Data



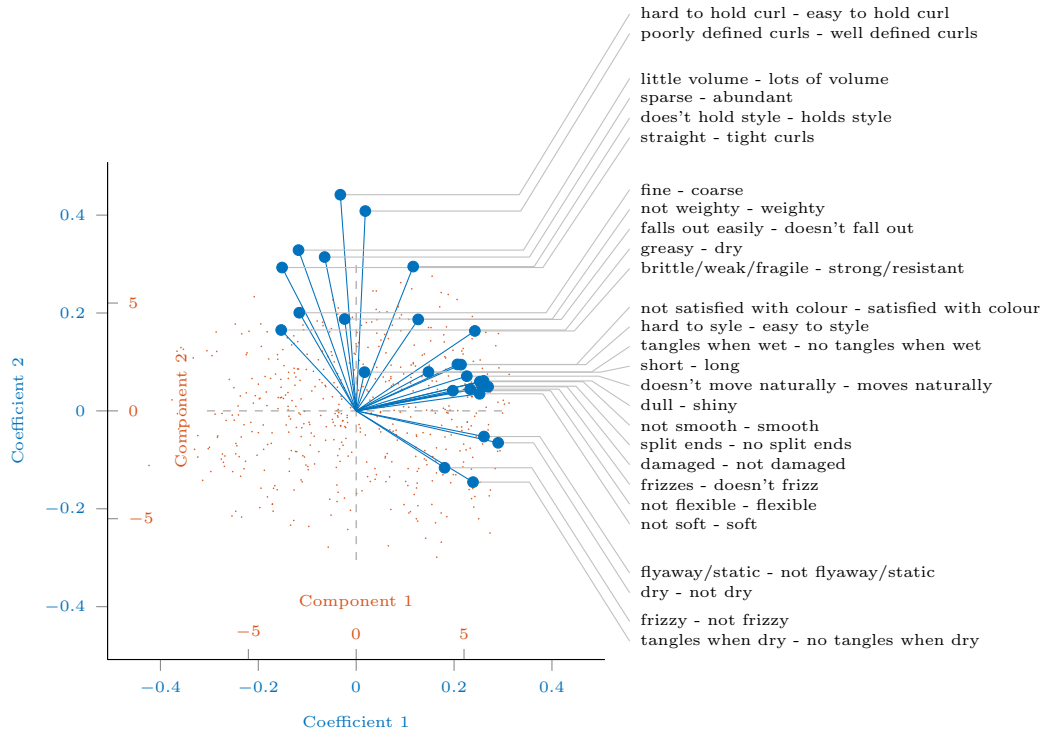
(a) Current hair

(b) Ideal hair

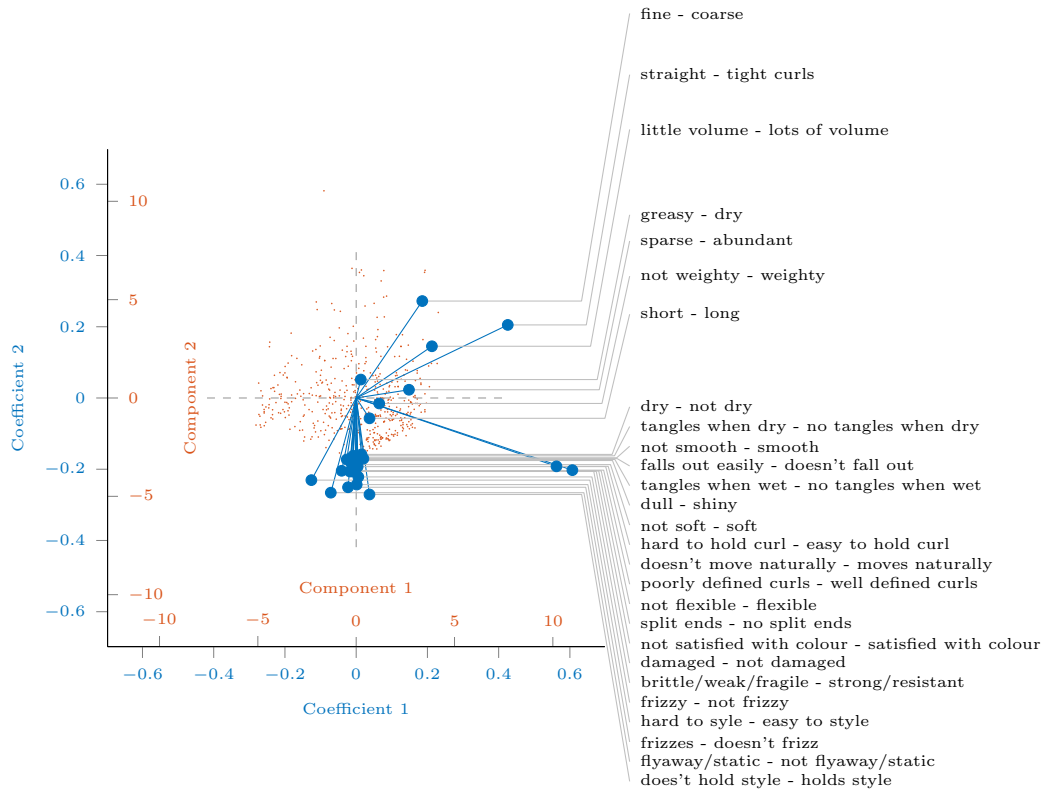
Figure A.1: Fraction of the total variance explained by successive principle components when analysing the data for each country separately.

In this appendix we show the results of applying the principle component analysis from Section 5.1.1 to the data for each country separately instead of using the combined data set. As we can see from Fig. A.1, the fraction of the total variance explained by the first k principle components is very similar for the different countries. Figures A.2–A.6 show the biplots of the first versus second principle components for

current and ideal hair attributes for each of the countries. The relationship between the first and second principle components is mostly similar for the different countries, where the first component is mostly related to hair dryness and damage and the second component to curliness. Notably the role of first and second component are swapped for the ideal hair attributes for Brazil and curliness does not play as strong a role for Japan.

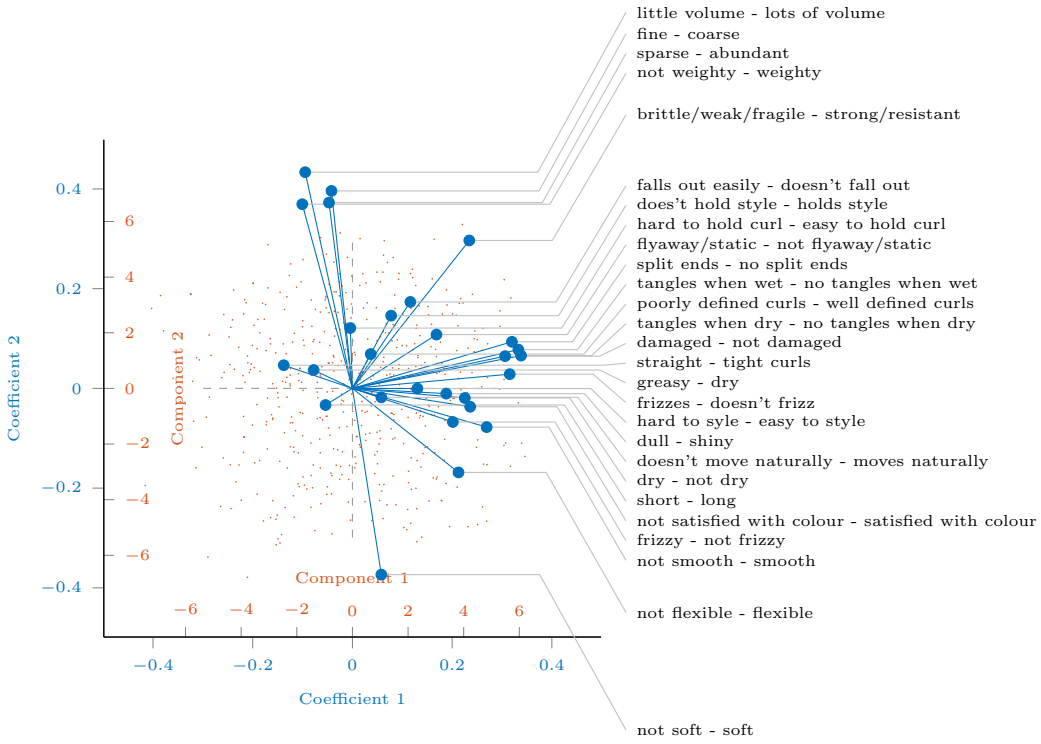


(a) Current hair

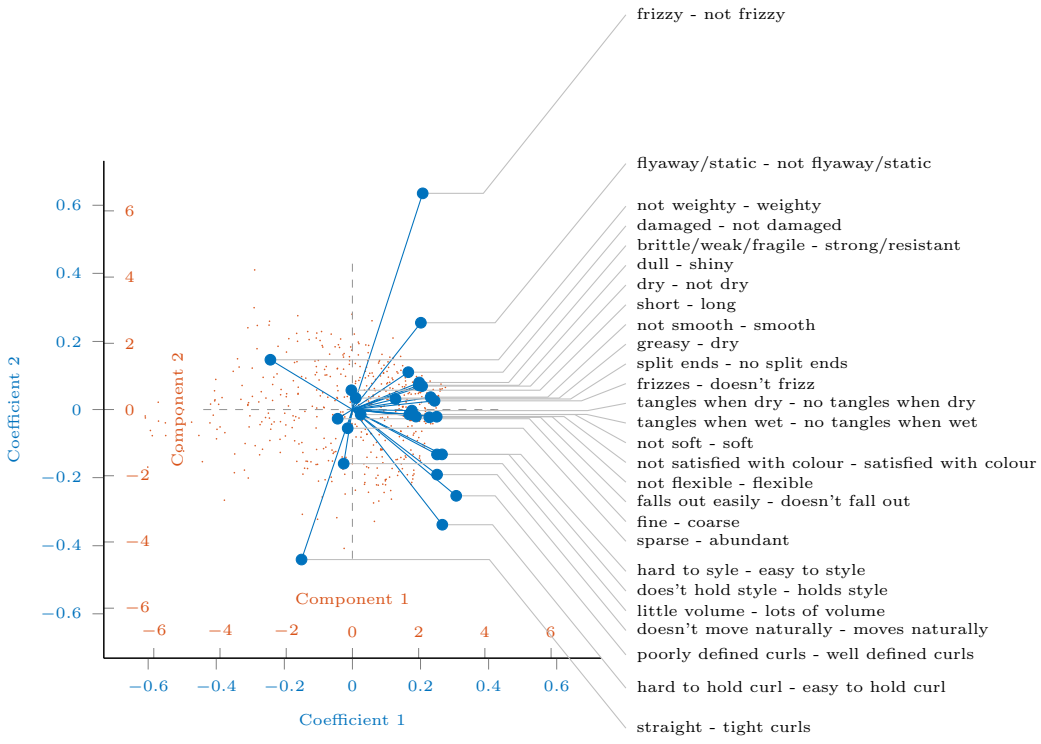


(b) Ideal hair

Figure A.2: Brazil: Biplot of first and second principle components for current and ideal hair attributes.

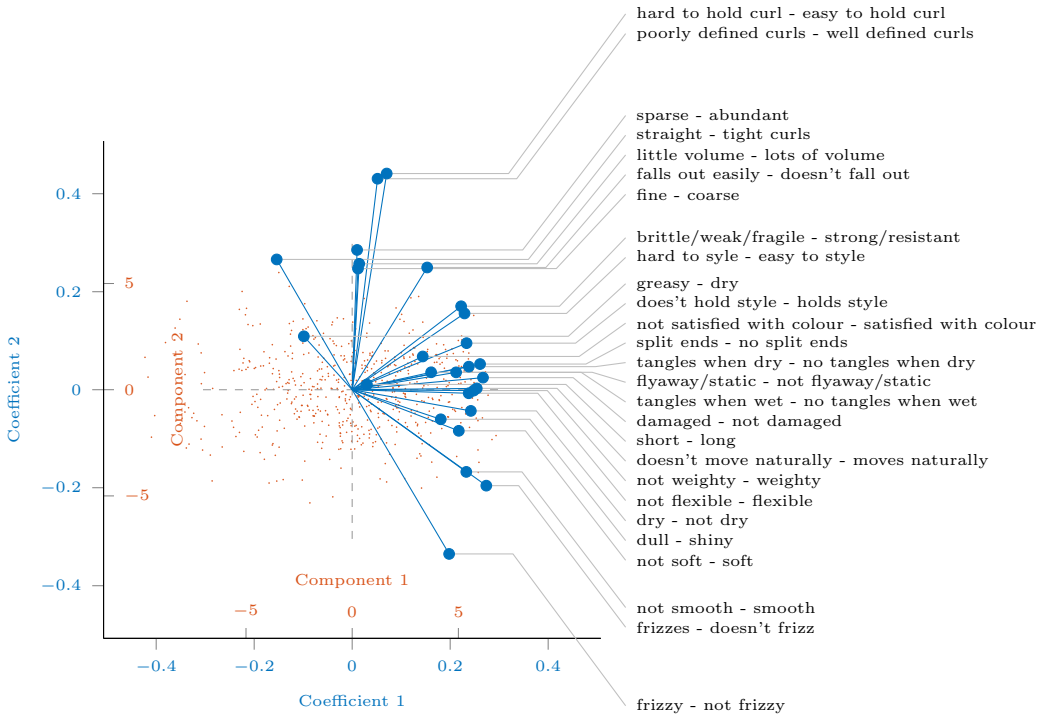


(a) Current hair

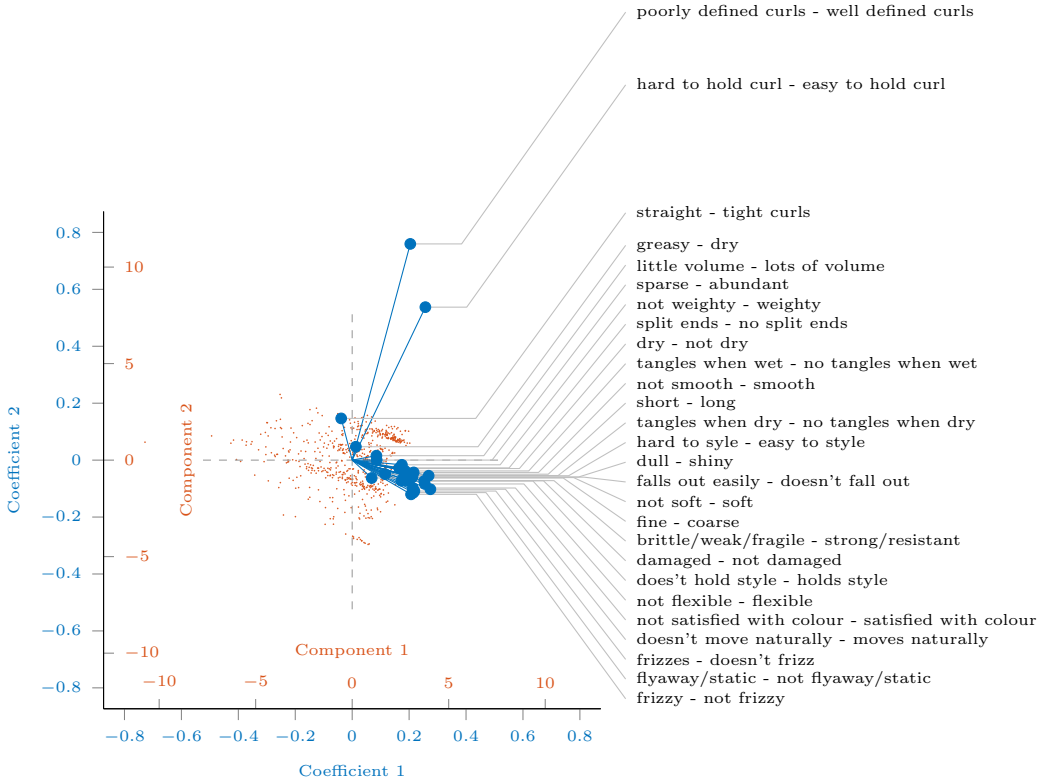


(b) Ideal hair

Figure A.3: Japan: Biplot of first and second principle components for current and ideal hair attributes.

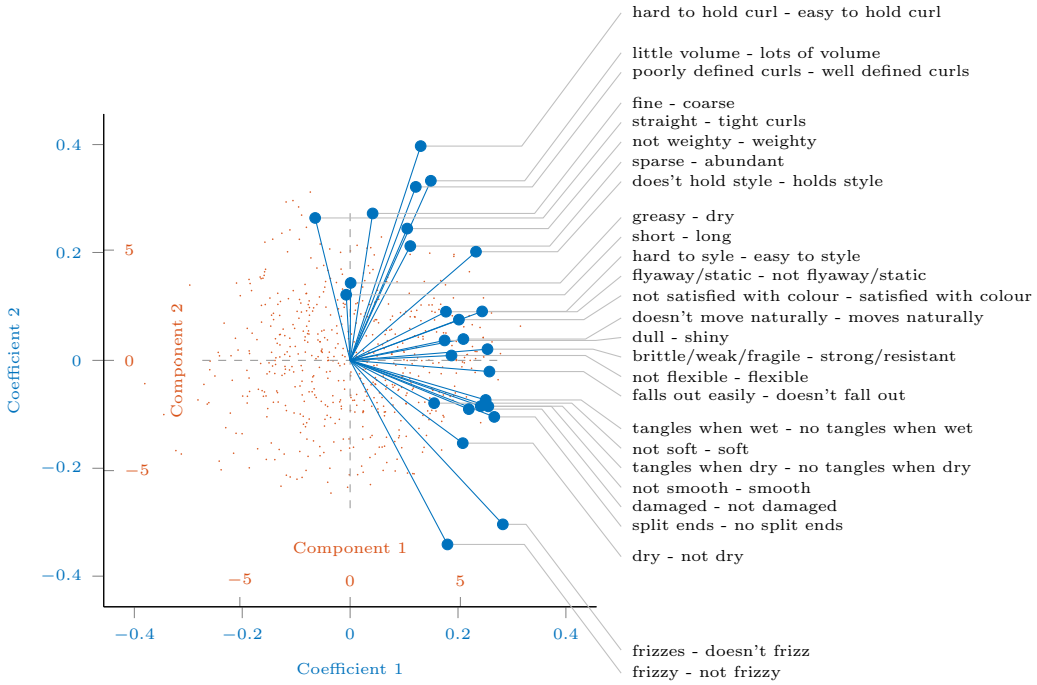


(a) Current hair

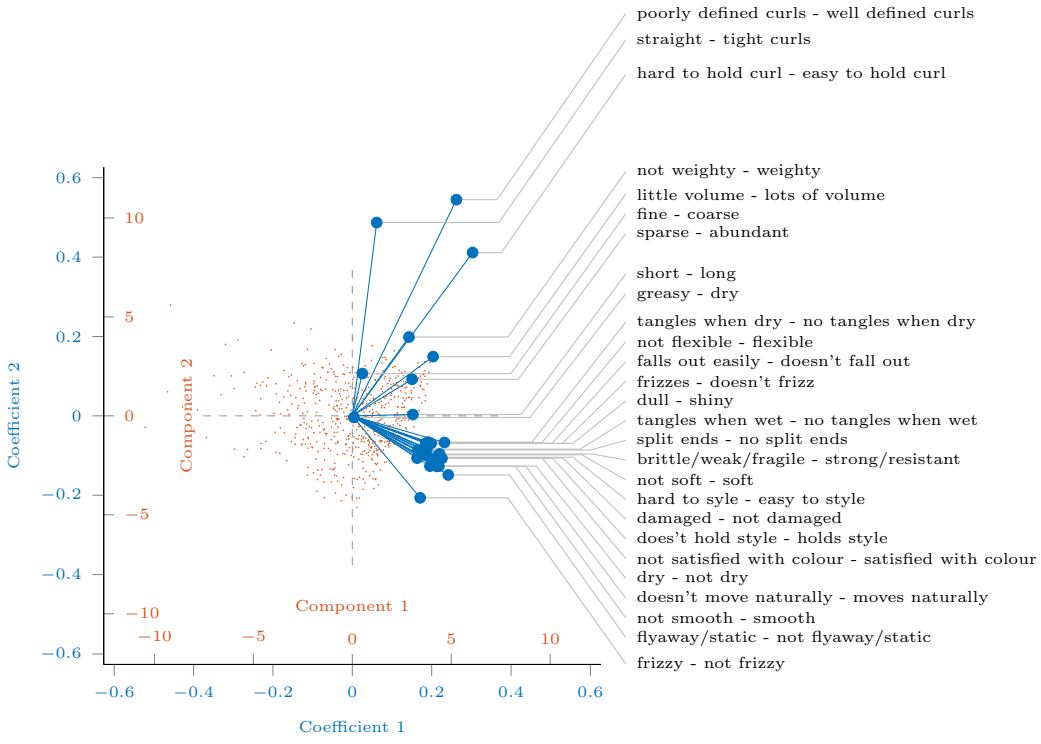


(b) Ideal hair

Figure A.4: Thailand: Biplot of first and second principle components for current and ideal hair attributes.

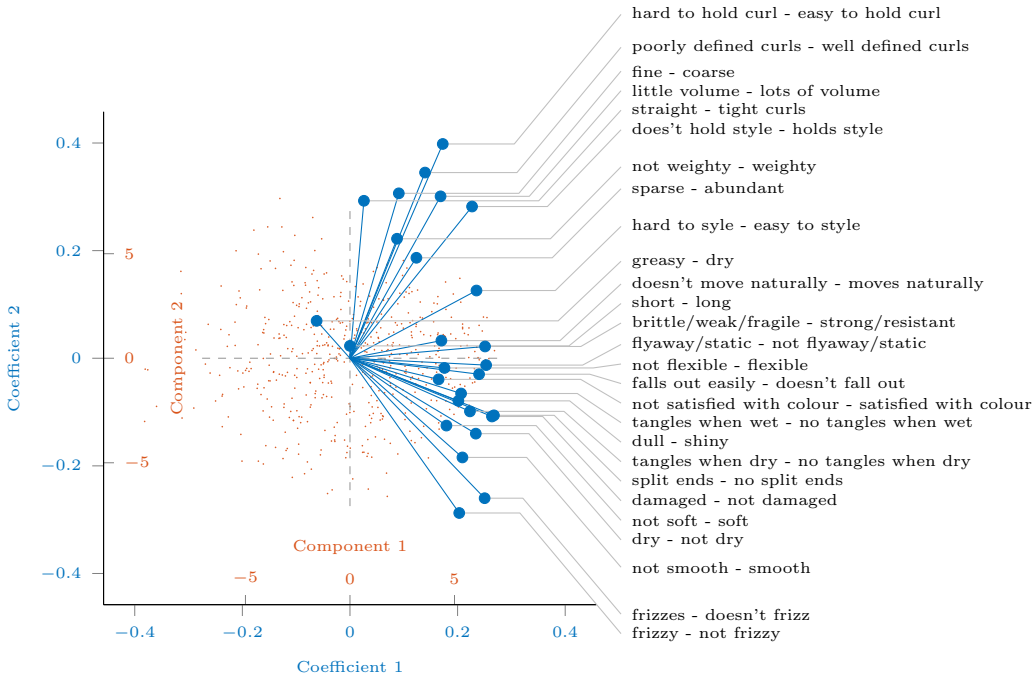


(a) Current hair

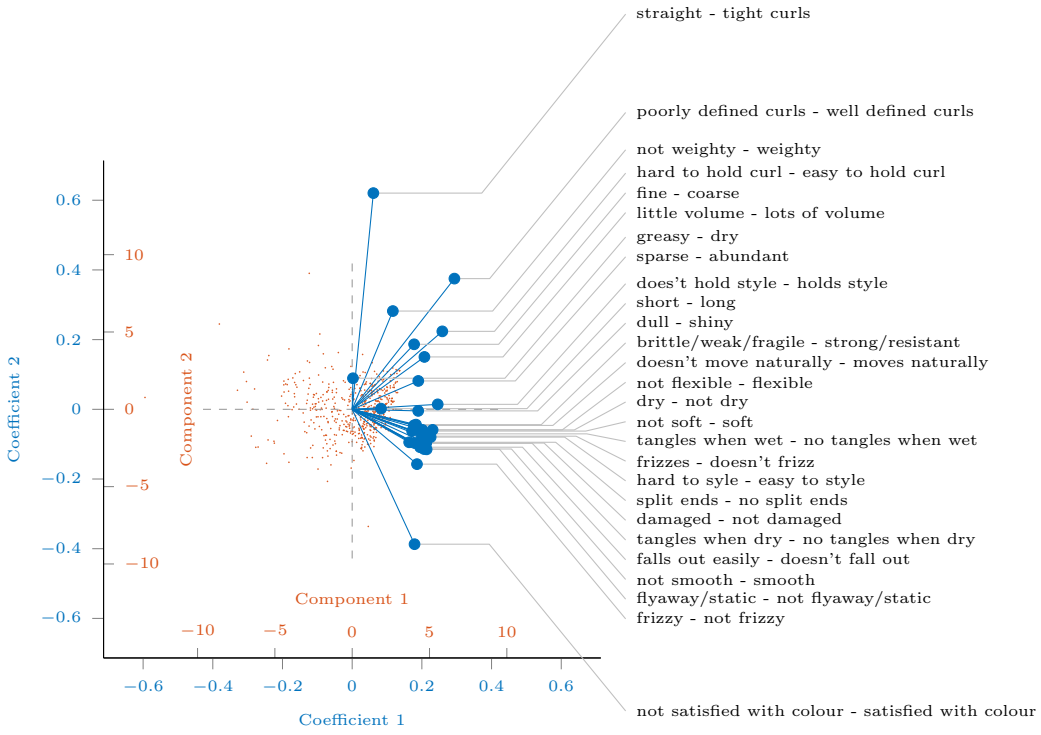


(b) Ideal hair

Figure A.5: UK: Biplot of first and second principle components for current and ideal hair attributes.



(a) Current hair



(b) Ideal hair

Figure A.6: USA: Biplot of first and second principle components for current and ideal hair attributes.

Appendix B

Louvain-Like Algorithms for Multilayer Community Detection

This appendix discusses our implementation of a Louvain-like [23] locally greedy algorithm for optimising a general modularity quality function of the type

$$Q(\mathcal{S}) = \sum_{i,j} \boldsymbol{B}_{ij} \delta(\boldsymbol{s}_i, \boldsymbol{s}_j), \tag{B.1}$$

which includes all the variants of modularity (see Section 2.5.1) we discuss in this thesis. In particular one can also write multilayer modularity in this form by flattening the tensors.

The Louvain algorithm for optimising a quality function of this type proceeds in two phases. Starting from an initial partition, during the first phase we iterate over nodes in randomised order repeatedly, where at each iteration we move the node under consideration to the group that results in the largest increase in modularity (if there is no move that improves modularity, a node keeps its assignment), until we cannot move any more nodes. At this point we enter the second phase of the algorithm where we aggregate the communities found so far to obtain a new modularity matrix

$$\tilde{\boldsymbol{B}}_{cd} = \sum_{ij} \boldsymbol{B}_{ij} \delta(\boldsymbol{s}_i, c) \delta(\boldsymbol{s}_j, d), \tag{B.2}$$

where $\boldsymbol{s}$ is the partition vector of the current partition of the nodes into communi-

ties. We proceed by applying phase one to the new modularity matrix and iterating phase one and two until convergence, keeping track of the community assignments of the original nodes along the way. This implementation of the Louvain algorithm deliberately sacrifices speed for convenience in experimenting with different null models, as it always performs computations using the full modularity matrix rather than taking advantage of the low-rank nature of many null-models. However it can still handle relatively large networks with hundreds of thousands of nodes in the case of multilayer networks with many layers as it takes advantage of the sparsity of the modularity matrix in these situations.

This is the simplest variant of the algorithm which we refer to as GenLouvain. However it has significant limitations when dealing with multilayer networks [21], as it often gets stuck in local optima where communities are identified sub-optimally across layers and exhibits abrupt transitions in behaviour as one changes the interlayer coupling ω. The second problem can be easily mitigated by relaxing the greedy nature of the algorithm, where instead of choosing the best possible move at each step, we choose a random move that improves modularity, where the probability of a particular move is proportional to its increase in modularity. This can be implemented at little extra computational cost, as we need to compute the change in modularity for all potential moves even for GenLouvain. We call this variant GenLouvainRand.

B.1 Post-Processing Multilayer Partitions

The problem of the algorithm getting stuck in local optima where communities are identified sub-optimally across layers has an interesting solution. Note that if we fix the restriction of the multilayer partition on each layer, maximising multilayer modularity simply amounts to maximising the sum of the persistence between restricted partitions (see Section 2.5.3 in adjacent layers (all pairs of layers for categorical coupling or consecutive layers for ordinal coupling). In the case of ordinal coupling we can solve this problem exactly fairly efficiently if we do not have too many communities. Note that starting from the first layer, the optimal community assignment for the

second layer is that which minimises the classification error between the two community assignments (see the discussion in Section 2.5.3), which is a classical assignment problem which one can solve using the Hungarian algorithm [107, 150]. Given the assignment for the second layer, we can then proceed to in the same way to find the optimal assignment for the third layer and so on. When we have identified the optimal assignment for the last layer, we have found the optimal multilayer partition that has the given restricted partitions for each layer.

For categorical coupling the situation is a bit more complex and we cannot necessarily identify the true optimal partition in the same way due to the circular dependencies between layers. Instead we proceed in an iterative manner. Note that if we fix the community labels for all layers except one, we can identify the optimal label for that layer in the same way as for the ordinal coupling. We can then iterate over layers repeatedly, where at each iteration we relabel the partition of the layer under consideration in an optimal way. If there is more than one optimal labelling for a layer, we pick one at random and we only update the labelling when the current labelling is not already optimal.

One can apply the post-processing procedure at different points during the execution of GenLouvain or GenLouvainRand. In this thesis we have used a variant where we post-process after convergence of the first iteration through the first phase of GenLouvainRand, which we call GenLouvainRandPost. This seems to work well for categorical coupling but not for ordinal coupling where it overemphasis persistence. Our code implementing GenLouvain and an earlier variant of GenLouvainRand is available at `http://netwiki.amath.unc.edu/GenLouvain/GenLouvain`.

Appendix C

Visualising Networks with Community Structure

Visualising networks directly can sometimes reveal interesting structural features in an intuitive way. The algorithm we use for the network visualisations in this thesis is a variant of the Kamada-Kawai spring embedding algorithm [97] which attempts to place the nodes in a way such that the geodesic distance between pairs of nodes is approximately proportional to the distance between them in the visualisation. In the case where we have identified communities in a network, we would often like to visualise the network in a way that communities are clearly separated, which allows one to see connection patterns between communities. To represent communities, we modify the spring embedding objective function proposed in [97] by replacing nodes with charged particles. Each community is represented by a charge, where nodes in the same community interact neutrally and nodes in different communities repulse each other. The modified objective function we optimise is

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in\mathbb{R}^{n\times 2}}{\operatorname{argmin}} \sum_{i,j=1}^{n} \frac{1-\delta(i,j)}{4\Delta_{ij}^2}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 - \Delta_{ij})^2 - \frac{1}{2}F\frac{\boldsymbol{C}_{ij}}{\|\boldsymbol{x}_i - \boldsymbol{x_j}\|_2}, \tag{C.1}$$

where $\boldsymbol{x}_i$ are node i's coordinates in the layout space, $\boldsymbol{C}_{ij}$ is a charge matrix representing community assignment, with $\boldsymbol{C}_{ij} = 0$ if nodes i and j are in the same community and $\boldsymbol{C}_{ij} = 1$ otherwise, and F is a parameter controlling the strength of the repulsive force between nodes in different communities. One can optimise this

objective function using a block-coordinate descent method [171] where we only move a single node at each iteration. Our code implementing this visualisation is available at `https://github.com/LJeub/SpringVisCom`.

Bibliography

[1] Adams, E. W., Fagot, R. F., and Robinson, R. E. (1965). A theory of appropriate statistics. *Psychometrika*, 30(2):99–127.

[2] Adcock, A. B., Sullivan, B. D., and Mahoney, M. W. (2013). Tree-like structure in large social and information networks. In *Proceedings of the 13th IEEE International Conference on Data Mining*, ICDM, pages 1–10. IEEE.

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

[4] Andersen, R., Chung, F. R. K., and Lang, K. J. (2006). Local graph partitioning using pagerank vectors. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '06, pages 475–486. IEEE.

[5] Andersen, R. and Lang, K. J. (2006). Communities from seed sets. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 223–232. ACM.

[6] Anderson, D., Tenzer, A., Barlev, G., Girvan, M., Antonsen, T. M., and Ott, E. (2012). Multiscale dynamics in communities of phase oscillators. *Chaos*, 22(1):013102.

[7] Arenas, A., Díaz-Guilera, A., and Pérez-Vicente, C. J. (2006). Synchronization reveals topological scales in complex networks. *Physical Review Letters*, 96(11):114102.

[8] Arenas, A., Fernández, A., and Gómez, S. (2008). Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039.

[9] Arora, S., Rao, S., and Vazirani, U. (2009). Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM*, 56(2):1–37.

[10] Backstrom, L., Boldi, P., Rosa, M., Ugander, J., and Vigna, S. (2012). Four degrees of separation. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 33–42. ACM.

[11] Bagrow, J. P. and Bollt, E. (2005). Local method for detecting communities. *Physical Review E*, 72(4):046108.

[12] Ball, B., Karrer, B., and Newman, M. E. J. (2011). Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3):036103.

[13] Bar-Joseph, Z., Gifford, D. K., and Jaakkola, T. S. (2001). Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(Suppl 1):S22–S29.

[14] Barabási, A.-L. (2005). Taming complexity. *Nature Physics*, 1(2):68–70.

[15] Barber, M. (2007). Modularity and community detection in bipartite networks. *Physical Review E*, 76(6):066102.

[16] Bassett, D. S., Owens, E. T., Daniels, K. E., and Porter, M. A. (2012). Influence of network topology on sound propagation in granular materials. *Physical Review E*, 86(4):041306.

[17] Bassett, D. S., Porter, M. A., Wymbs, N. F., Grafton, S. T., Carlson, J. M., and Mucha, P. J. (2013a). Robust detection of dynamic community structure in networks. *Chaos*, 23(1):013142.

[18] Bassett, D. S., Wymbs, N. F., Porter, M. A., Mucha, P. J., Carlson, J. M., and Grafton, S. T. (2011). Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences*, 108(18):7641–7646.

[19] Bassett, D. S., Wymbs, N. F., Rombach, M. P., Porter, M. A., Mucha, P. J., and Grafton, S. T. (2013b). Task-based core-periphery organization of human brain dynamics. *PLoS Computational Biology*, 9(9):e1003171.

[20] Battiston, F., Nicosia, V., and Latora, V. (2014). Structural measures for multiplex networks. *Physical Review E*, 89(3):032804.

[21] Bazzi, M., Porter, M. A., Williams, S., McDonald, M., Fenn, D. J., and Howison, S. D. (2014). Community detection in temporal multilayer networks, and its application to correlation networks. arXiv:1501.00040 [physics.soc-ph].

[22] Beguerisse-Díaz, M., Garduño-Hernández, G., Vangelov, B., Yaliraki, S. N., and Barahona, M. (2014). Interest communities and flow roles in directed networks: The Twitter network of the UK riots. *Journal of The Royal Society Interface*, 11(101):20140940.

[23] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

[24] Boccaletti, S., Bianconi, G., Criado, R., del Genio, C. I., Gómez-Gardeñes, J., Romance, M., Sendiña-Nadal, I., Wang, Z., and Zanin, M. (2014). The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122.

[25] Borgatti, S. P. and Everett, M. G. (2000). Models of core/periphery structures. *Social Networks*, 21(4):375–395.

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

[27] Brauer, F. and Castillo-Chavez, C. (2012). *Mathematical Models in Population Biology and Epidemiology*, volume 40 of *Texts in Applied Mathematics*. Springer New York, New York, NY, 2nd edition.

[28] Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52. Morgan Kaufmann Publishers Inc.

[29] Bridson, M. R. and Häfliger, A. (1999). *Metric Spaces of Non-Positive Curvature*. Springer, Berlin.

[30] Buskirk, T. D., Willoughby, L. M., and Tomazic, T. T. (2013). *Nonparametric Statistical Techniques*. Oxford University Press, Oxford, UK.

[31] Cardillo, A., Gómez-Gardeñes, J., Zanin, M., Romance, M., Papo, D., Pozo, F. d., and Boccaletti, S. (2013). Emergence of network features from multiplexity. *Scientific Reports*, 3.

[32] Carifio, J. and Perla, R. J. (2007). Ten common misunderstandings, misconceptions, persistent myths and urban legends about Likert scales and Likert response formats and their antidotes. *Journal of Social Sciences*, 3(3):106–116.

[33] Chen, W., Fang, W., Hu, G., and Mahoney, M. W. (2012). On the hyperbolicity of small-world and tree-like random graphs. In *Algorithms and Computation*, pages 278–288. Springer Berlin Heidelberg, Berlin, Heidelberg.

[34] Chen, W., Fang, W., Hu, G., and Mahoney, M. W. (2013). On the hyperbolicity of small-world and tree-like random graphs. *Internet Mathematics*, 9(4):434–491.

[35] Christakis, N. A. and Fowler, J. H. (2007). The spread of obesity in a large social network over 32 years. *New England Journal of Medicine*, 357(4):370–379.

[36] Christakis, N. A. and Fowler, J. H. (2013). Social contagion theory: Examining dynamic social networks and human behavior. *Statistics in Medicine*, 32(4):556–577.

[37] Chung, F. R. K. (1997). *Spectral graph theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. AMS, Providence.

[38] Chung, F. R. K. and Lu, L. (2006). *Complex graphs and networks*, volume 107 of *CBMS Regional Conference Series in Mathematics*. AMS, Providence.

[39] Clauset, A. (2005). Finding local community structure in networks. *Physical Review E*, 72(2):026132.

[40] Condon, A. and Karp, R. M. (2001). Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140.

[41] Cormack, R. M. (1971). A review of classification. *Journal of the Royal Statistical Society. Series A*, 134(3):321–367.

[42] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience, New York, NY.

[43] Cozzo, E., Kivelä, M., De Domenico, M., Solè-Ribalta, A., Arenas, A., Gómez, S., Porter, M. A., and Moreno, Y. (2015). Structure of triadic relations in multiplex networks. arXiv:1307.6780v3 [physics.soc-ph].

[44] Csermely, P., London, A., Wu, L. Y., and Uzzi, B. (2013). Structure and dynamics of core/periphery networks. *Journal of Complex Networks*, 1(2):93–123.

[45] Cuadros, O., Botelho, G., Rodrigues, F., and Neto, J. B. (2012). Segmentation of large images with complex networks. In *Proceedings of the 25th SIBGRAPI Conference on Graphics, Patterns and Images*, SIBGRAPI, pages 24–31. IEEE.

[46] Cucuringu, M. and Mahoney, M. W. (2011). Localization on low-order eigenvectors of data matrices. arXiv:1109.1355v1 [cs.DM].

[47] De Arruda, G. F., Da Fontoura Costa, L., and Rodrigues, F. A. (2012). A complex networks approach for data clustering. *Physica A: Statistical Mechanics and its Applications*, 391(23):6174–6183.

[48] De Domenico, M., Lancichinetti, A., Arenas, A., and Rosvall, M. (2015). Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Physical Review X*, 5(1):011027.

[49] De Domenico, M., Solè-Ribalta, A., Cozzo, E., Kivelä, M., Moreno, Y., Porter, M. A., Gómez, S., and Arenas, A. (2013). Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022.

[50] De Domenico, M., Solè-Ribalta, A., Gómez, S., and Arenas, A. (2014). Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences*, 111(23):8351–8356.

[51] Defays, D. (1977). An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366.

[52] Delvenne, J.-C., Yaliraki, S. N., and Barahona, M. (2010). Stability of graph communities across time scales. *Proceedings of the National Academy of Sciences*, 107(29):12755–12760.

[53] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

[54] Dongen, S. (2000). Performance criteria for graph clustering and markov cluster experiments. Technical report, CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands.

[55] Duch, J. and Arenas, A. (2005). Community detection in complex networks using extremal optimization. *Physical Review E*, 72(2):027104.

[56] Easley, D. and Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, Cambridge, UK.

[57] Ernst, M. D. (2004). Permutation methods: A basis for exact inference. *Statistical Science*, 19(4):676–685.

[58] Estrada, E. (2011). Community detection based on network communicability. *Chaos*, 21(1):016103–016117.

[59] Evans, T. S., Lambiotte, R., and Panzarasa, P. (2011). Community structure and patterns of scientific collaboration in business and management. *Scientometrics*, 89(1):381–396.

[60] Expert, P., Evans, T. S., Blondel, V. D., and Lambiotte, R. (2011). Uncovering space-independent communities in spatial networks. *Proceedings of the National Academy of Sciences*, 108(19):7663–7668.

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

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

[63] Friedman, J. H., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

[64] Gantmakher, F. R. (2000). *The Theory of Matrices*, volume 2. AMS Chelsea.

[65] Gelfand, A. E., Hills, S. E., Racine-Poon, A., and Smith, A. F. M. (1990). Illustration of bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association*, 85(412):972–985.

[66] Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

[67] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

[68] Ghosh, R., Lerman, K., Teng, S.-H., and Yan, X. (2014). The interplay between dynamics and networks: Centrality, communities, and cheeger inequality. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1406–1415. ACM.

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

[70] Gjoka, M., Smith, E., and Butts, C. (2014). Estimating clique composition and size distributions from sampled network data. In *Proceedings of the 2014 IEEE Conference on Computer Communications Workshops*, INFOCOM WKSHPS, pages 837–842. IEEE.

[71] Gleeson, J. P. (2013). Binary-state dynamics on complex networks: Pair approximation and beyond. *Physical Review X*, 3(2):021004.

[72] Gleich, D. F. (2014). PageRank beyond the web. arXiv:1407.5107v1 [cs.SI].

[73] Gleich, D. F. and Mahoney, M. W. (2014). Anti-differentiating approximation algorithms: A case study with min-cuts, spectral, and flow. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1018–1025. JMLR W&CP.

[74] Gleich, D. F. and Seshadhri, C. (2012). Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 597–605. ACM.

[75] Goebel, B., Dawy, Z., Hagenauer, J., and Mueller, J. C. (2005). An approximation to the distribution of finite sample size mutual information estimates. In *Proceedings of the 2005 IEEE International Conference on Communications*, volume 2 of *ICC*, pages 1102–1106. IEEE.

[76] Gómez, S., Díaz-Guilera, A., Gómez-Gardeñes, J., Pérez-Vicente, C. J., Moreno, Y., and Arenas, A. (2013). Diffusion dynamics on multiplex networks. *Physical Review Letters*, 110(2):028701.

[77] Gómez, S., Jensen, P., and Arenas, A. (2009). Analysis of community structure in networks of correlated data. *Physical Review E*, 80(1):016114.

[78] González, M. C., Herrmann, H. J., Kertész, J., and Vicsek, T. (2007). Community structure and ethnic preferences in school friendship networks. *Physica A: Statistical Mechanics and its Applications*, 379(1):307–316.

[79] Good, B. H., de Montjoye, Y.-A., and Clauset, A. (2010). Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106.

[80] Grady, D., Brune, R., Thiemann, C., Theis, F., and Brockmann, D. (2012). Modularity maximization and tree clustering: Novel ways to determine effective geographic borders. In Thai, M. T. and Pardalos, P. M., editors, *Handbook of Optimization in Complex Networks*, Springer Optimization and its Applications, pages 169–208. Springer, New York.

[81] Granell, C., Darst, R. K., Arenas, A., Fortunato, S., and Gómez, S. (2015). A benchmark model to assess community structure in evolving networks. arXiv:1501.05808v1 [physics.soc-ph].

[82] Granell, C., Gómez, S., and Arenas, A. (2011). Mesoscopic analysis of networks: Applications to exploratory analysis and data clustering. *Chaos*, 21(1):016102.

[83] Granovetter, M. S. (1973). The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380.

[84] Guimerà, R. and Amaral, L. A. N. (2005). Cartography of complex networks: Modules and universal roles. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(02):P02001.

[85] Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, New York, NY, 2nd edition.

[86] Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137.

[87] Holme, P. and Saramäki, J. (2012). Temporal networks. *Physics Reports*, 519(3):97–125.

[88] Hoory, S., Linial, N., and Wigderson, A. (2006). Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561 (electronic).

[89] Hu, H., Laurent, T., Porter, M. A., and Bertozzi, A. (2013). A method based on total variation for network modularity optimization using the MBO schemebo scheme. *SIAM Journal on Applied Mathematics*, 73(6):2224–2246.

[90] Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50.

[91] Jackson, M. O. (2008). *Social and Economic Networks*. Princeton University Press, Princeton, NJ.

[92] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323.

[93] Jerrum, M. and Sinclair, A. (1988). Conductance and the rapid mixing property for markov chains: the approximation of permanent resolved. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, STOC '88, pages 235–244. ACM.

[94] Jeub, L. G. S., Balachandran, P., Porter, M. A., Mucha, P. J., and Mahoney, M. W. (2015). Think locally, act locally: Detection of small, medium-sized, and large communities in large networks. *Physical Review E*, 91:012821.

[95] Jolliffe, I. T. (2013). *Principal Component Analysis*. Springer Science & Business Media.

[96] Jonckheere, E., Lou, M., Bonahon, F., and Baryshnikov, Y. (2011). Euclidean versus hyperbolic congestion in idealized versus experimental networks. *Internet Mathematics*, 7(1):1–27.

[97] Kamada, T. and Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15.

[98] Karrer, B. and Newman, M. E. J. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107.

[99] Karsai, M., Kivelä, M., Pan, R. K., Kaski, K., Kertész, J., Barabási, A.-L., and Saramäki, J. (2011). Small but slow world: How network topology and burstiness slow down spreading. *Physical Review E*, 83(2).

[100] Kawamoto, T. and Rosvall, M. (2015). Estimating the resolution limit of the map equation in community detection. *Physical Review E*, 91(1):012809.

[101] Kemeny, J. G. and Snell, J. L. (1960). *Finite Markov Chains*. The University Series in Undergraduate Mathematics. D. Van Nostrand Company, INC., Princeton, NJ.

[102] Kim, Y., Son, S.-W., and Jeong, H. (2010). Finding communities in directed networks. *Physical Review E*, 81(1):016103.

[103] Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014). Multilayer networks. *Journal of Complex Networks*, 2(3):203–271.

[104] Knapp, T. R. (1990). Treating ordinal scales as interval scales: An attempt to resolve the controversy. *Nursing Research*, 39(2):121–123.

[105] Kraskov, A., Stögbauer, H., Andrzejak, R. G., and Grassberger, P. (2005). Hierarchical clustering using mutual information. *Europhysics Letters*, 70(2):278–284.

[106] Kruskal, W. H. (1958). Ordinal measures of association. *Journal of the American Statistical Association*, 53(284):814.

[107] Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

[108] Kuperman, M. N. (2013). Invited review: Epidemics on social networks. *Papers in Physics*, 5(0):050003.

[109] Lambiotte, R. (2010). Multi-scale modularity in complex networks. In *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, WiOpt, pages 546–553. IEEE.

[110] Lambiotte, R., Delvenne, J.-C., and Barahona, M. (2009). Laplacian dynamics and multiscale modular structure in networks. arXiv:0812.1770v3 [physics.soc-ph].

[111] Lambiotte, R., Delvenne, J.-C., and Barahona, M. (2014). Random walks, markov processes and the multiscale modular organization of complex networks. *IEEE Transactions on Network Science and Engineering*, 1(2):76–90.

[112] Lambiotte, R. and Rosvall, M. (2012). Ranking and clustering of nodes in networks with smart teleportation. *Physical Review E*, 85(5):056107.

[113] Lambiotte, R., Sinatra, R., Delvenne, J.-C., Evans, T. S., Barahona, M., and Latora, V. (2011). Flow graphs: Interweaving dynamics and structure. *Physical Review E*, 84(1):017102.

[114] Lancichinetti, A. and Fortunato, S. (2009a). Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118.

[115] Lancichinetti, A. and Fortunato, S. (2009b). Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117.

[116] Lancichinetti, A. and Fortunato, S. (2011). Limits of modularity maximization in community detection. *Physical Review E*, 84(6):066122.

[117] Lancichinetti, A. and Fortunato, S. (2012). Consensus clustering in complex networks. *Scientific Reports*, 2:336.

[118] Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110.

[119] Lancichinetti, A., Radicchi, F., Ramasco, J. J., and Fortunato, S. (2011). Finding statistically significant communities in networks. *PLoS ONE*, 6(4):e18961.

[120] Larsen, B. and Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 16–22. ACM.

[121] Lazega, E. (2001). *The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership*. Oxford University Press, Oxford, UK.

[122] Lazega, E. and Pattison, P. E. (1999). Multiplexity, generalized exchange and cooperation in organizations: a case study. *Social Networks*, 21(1):67–90.

[123] Lehmann, S. (2010). Intermittent updates: Pervasive overlap. available at `http://sunelehmann.com/2010/06/29/pervasive-overlap/`.

[124] Leicht, E. A. and Newman, M. E. J. (2008). Community structure in directed networks. *Physical Review Letters*, 100(11):118703.

[125] Lerman, K. and Ghosh, R. (2012). Network structure, topology, and dynamics in generalized models of synchronization. *Physical Review E*, 86(2):026108.

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

[127] Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2008). Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 695–704. ACM.

[128] Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2009). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123.

[129] Leskovec, J., Lang, K. J., and Mahoney, M. W. (2010b). Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 631–640. ACM.

[130] Lewis, A. C. F., Jones, N. S., Porter, M. A., and Deane, C. M. (2010). The function of communities in protein interaction networks at multiple scales. *BMC Systems Biology*, 4(1):100.

[131] Li, M., Badger, J. H., Chen, X., Kwong, S., Kearney, P., and Zhang, H. (2001). An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2):149–154.

[132] Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):5–55.

[133] Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann Publishers Inc.

[134] Lovász, L. (1993). Random walks on graphs: a survey. In *Combinatorics, Paul Erdős is Eighty*, volume 2, pages 353–397. János Bolyai Mathematical Society, Budapest.

[135] Lu, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1):1–49.

[136] Luce, R. D. (1950). Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15(2):169–190.

[137] Luce, R. D. and Perry, A. D. (1949). A method of matrix analysis of group structure. *Psychometrika*, 14(1):95–116.

[138] Macon, K. T., Mucha, P. J., and Porter, M. A. (2012). Community structure in the united nations general assembly. *Physica A: Statistical Mechanics and its Applications*, 391(1-2):343–361.

[139] Maddala, G. S. (1983). *Limited-Dependent and Qualitative Variables in Econometrics*. Econometric Society Monographs. Cambridge University Press, Cambridge, UK.

[140] Mahoney, M. W. (2012). Approximate computation and implicit regularization for very large-scale data analysis. In *Proceedings of the 31st Symposium on Principles of Database Systems*, pages 143–154. ACM.

[141] Mahoney, M. W. and Orecchia, L. (2011). Implementing regularization implicitly via approximate eigenvector computation. In *Proceedings of the 28th International Conference on Machine Learning*, ICML '11, pages 121–128. ACM.

[142] Mahoney, M. W., Orecchia, L., and Vishnoi, N. K. (2012). A local spectral method for graphs: With applications to improving graph partitions and exploring data graphs locally. *Journal of Machine Learning Research*, 13:2339–2365.

[143] McGill, M. (1979). An evaluation of factors affecting document ranking by information retrieval systems. Technical Report ED1885867, Syracuse University, School of Information Studies.

[144] Meilă, M. (2005). Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05. ACM.

[145] Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

[146] Meilă, M. and Heckerman, D. (2001). An experimental comparison of model-based clustering methods. *Machine Learning*, 42(1/2):9–29.

[147] Mihail, M. (1989). Conductance and convergence of markov chains - a combinatorial treatment of expanders. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 526–531. IEEE.

[148] Mucha, P. J. and Porter, M. A. (2010). Communities in multislice voting networks. *Chaos*, 20:041108.

[149] Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878.

[150] Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.

[151] Newman, M. E. J. (2003). Mixing patterns in networks. *Physical Review E*, 67(2):026126.

[152] Newman, M. E. J. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104.

[153] Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford University Press, Oxford, UK.

[154] Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113.

[155] Newman, M. E. J. and Peixoto, T. P. (2015). Generalized communities in networks. arXiv:1505.07478 [cs.SI].

[156] Onnela, J.-P., Saramäki, J., Hyvönen, J., Szabó, G., Lazer, D., Kaski, K., Kertész, J., and Barabási, A.-L. (2007). Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 104(18):7332–7336.

[157] Onnela, J.-P., Saramäki, J., Kertész, J., and Kaski, K. (2005). Intensity and coherence of motifs in weighted complex networks. *Physical Review E*, 71(6):065103.

[158] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the Web. Technical Report 1999-66, Stanford InfoLab.

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

[160] Pan, R. K. and Sinha, S. (2009). Modularity produces small-world networks with dynamical time-scale separation. *Europhysics Letters*, 85(6):68006.

[161] Peel, L. and Clauset, A. (2014). Detecting change points in the large-scale structure of evolving networks. arXiv:1403.0989v2 [cs.SI].

[162] Peixoto, T. P. (2013). Parsimonious module inference in large networks. *Physical Review Letters*, 110(14):148701.

[163] Peixoto, T. P. (2014). Hierarchical block structures and high-resolution model selection in large networks. *Physical Review X*, 4(1):011047.

[164] Peixoto, T. P. and Bornholdt, S. (2012). Evolution of robust network topologies: Emergence of central backbones. *Physical Review Letters*, 109(11):118703.

[165] Porter, M. A., Mucha, P. J., Newman, M. E. J., and Friend, A. J. (2007). Community structure in the United States House of Representatives. *Physica A: Statistical Mechanics and its Applications*, 386(1):414–438.

[166] Porter, M. A., Mucha, P. J., Newman, M. E. J., and Warmbrand, C. M. (2005). A network analysis of committees in the US House of Representatives. *Proceedings of the National Academy of Sciences*, 102(20):7057–7062.

[167] Porter, M. A., Onnela, J.-P., and Mucha, P. J. (2009). Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 1164–1166.

[168] Psorakis, I., Roberts, S. J., Ebden, M., and Sheldon, B. (2011). Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E*, 83(6):066114.

[169] Reichardt, J. and Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, 74(1):016110.

[170] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186. ACM.

[171] Richtárik, P. and Takáč, M. (2014). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38.

[172] Rogers, D. J. and Tanimoto, T. T. (1960). A computer program for classifying plants. *Science*, 132(3434):1115–1118.

[173] Rombach, M. P. (2012). The Zachary karate cake. available at `http://people.maths.ox.ac.uk/porterm/temp/karatecake.jpg`.

[174] Rombach, M. P., Porter, M. A., Fowler, J. H., and Mucha, P. J. (2014). Core-periphery structure in networks. *SIAM Journal on Applied Mathematics*, 74(1):167–190.

[175] Ronhovde, P., Chakrabarty, S., Hu, D., Sahu, M., Sahu, K. K., Kelton, K. F., Mauro, N. A., and Nussinov, Z. (2011). Detecting hidden spatial and spatio-temporal structures in glasses and complex physical systems by multiresolution network clustering. *The European Physical Journal E*, 34(9):105.

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

[177] Rosvall, M. and Bergstrom, C. T. (2010). Mapping change in large networks. *PLoS ONE*, 5(1):e8694.

[178] Roweis, S. (1998). EM algorithms for PCA and SPCA. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, pages 626–632. MIT Press.

[179] Sales-Pardo, M., Guimerà, R., Moreira, A. A., and Amaral, L. A. N. (2007). Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39):15224–15229.

[180] Saramäki, J., Kivelä, M., Onnela, J.-P., Kaski, K., and Kertész, J. (2007). Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2):027105.

[181] Sarnak, P. (2004). What is... an expander? *Notices of the AMS*, 51(7):762–763.

[182] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295. ACM.

[183] Sarzynska, M., Leicht, E. A., Chowell, G., and Porter, M. A. (2014). Null models for community detection in spatially-embedded, temporal networks. arXiv:1407.6297v1 [physics.soc-ph].

[184] Schaub, M. T., Delvenne, J.-C., Yaliraki, S. N., and Barahona, M. (2012a). Markov dynamics as a zooming lens for multiscale community detection: Non clique-like communities and the field-of-view limit. *PLoS ONE*, 7(2):e32210.

[185] Schaub, M. T., Lambiotte, R., and Barahona, M. (2012b). Encoding dynamics for multiscale community detection: Markov time sweeping for the map equation. *Physical Review E*, 86(2):026112.

[186] Shalizi, C. R., Hagberg, A., and Clauset, A. (2013). Network scientists with karate trophies. available at `http://networkkarate.tumblr.com`.

[187] Shardanand, U. and Maes, P. (1995). Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 210–217. ACM Press/Addison-Wesley Publishing Co.

[188] Sibson, R. (1973). SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34.

[189] Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482.

[190] Snijders, T. A. B., Pattison, P. E., Robins, G. L., and Handcock, M. S. (2006). New specifications for exponential random graph models. *Sociological Methodology*, 36(1):99–153.

[191] Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

[192] Stevens, S. S. (1946). On the theory of scales of measurement. *Science*, 103(2684):677–680.

[193] Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540.

[194] Theodoridis, S. and Koutroumbas, K. (2006). *Pattern Recognition*. Academic Press, Burlington, MA, USA, 3rd edition.

[195] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288.

[196] Topchy, A., Jain, A. K., and Punch, W. (2005). Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881.

[197] Townsend, J. T. and Ashby, F. G. (1984). Measurement scales and statistics: The misconception misconceived. *Psychological Bulletin*, 96(2):394–401.

[198] Traag, V. A., Van Dooren, P., and Nesterov, Y. (2011). Narrow scope for resolution-limit-free community detection. *Physical Review E*, 84(1):016114.

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

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

[201] Ugander, J., Karrer, B., Backstrom, L., and Marlow, C. (2011). The anatomy of the Facebook social graph. arXiv:1111.4503 [cs.SI].

[202] van der Vorst, H. A. (1992). Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644.

[203] Velleman, P. F. and Wilkinson, L. (1993). Nominal, ordinal, interval, and ratio typologies are misleading. *The American Statistician*, 47(1):65–72.

[204] Wasserman, S. and Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK.

[205] Watts, D. J. (1999). Networks, dynamics, and the small-world phenomenon. *American Journal of Sociology*, 105(2):493–527.

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

[207] Waugh, A. S., Pei, L., Fowler, J. H., Mucha, P. J., and Porter, M. A. (2009). Party polarization in congress: A network science approach. arXiv:0907.3509 [physics.soc-ph].

[208] Whang, J. J., Gleich, D. F., and Dhillon, I. S. (2015). Overlapping community detection using neighborhood-inflated seed expansion. arXiv:1503.07439 [cs.SI].

[209] Wymbs, N. F., Bassett, D. S., Mucha, P. J., Porter, M. A., and Grafton, S. T. (2012). Differential recruitment of the sensorimotor putamen and frontoparietal cortex during motor chunking in humans. *Neuron*, 74(5):936–946.

[210] Yang, J. and Leskovec, J. (2012). Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, pages 3:1–3:8. ACM.

[211] Yang, J. and Leskovec, J. (2014). Structure and overlaps of ground-truth communities in networks. *ACM Transactions on Intelligent Systems and Technology*, 5(2):1–35.

[212] Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473.

[213] Zhu, Z. A., Lattanzi, S., and Mirrokni, V. (2013). A local algorithm for finding well-connected clusters. In *Proceedings of the 30th International Conference on Machine Learning*, pages 396–404. JMLR W&CP.

[214] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B*, 67:301–320.