Pruning Sets of Modularity Partitions in Single and Multi-layer Networks Via an Equivalence to Stochastic Block Model Inference

Ryan A. Gibson

An undergraduate honors thesis submitted to the faculty of the University of North Carolina at Chapel Hill in fulfillment of the requirements for the Honors Carolina Senior Honors Thesis in the Department of Mathematics.

Chapel Hill
2019

Approved by:

Peter J. Mucha

Katherine Newhall

Marc Niethammer

ABSTRACT

RYAN A. GIBSON: Pruning Sets of Modularity Partitions in Single and Multi-layer Networks Via
an Equivalence to Stochastic Block Model Inference
(Under the direction of Peter J. Mucha, Katherine Newhall, and Marc Niethammer)

Partitioning a network into communitites of densely connected nodes is an important problem across a wide range of disciplines. We demonstrate a method for pruning sets of such partitions to identify small subsets that are statistically significant from the perspective of stochastic block model inference. Crucially, our method works in both single and multi-layer domains and allows for restricting to a fixed number of communities when desired. On the examples considered here, our procedure identifies fewer than 10 "important" partitions, even when the number of unique input partitions exceeds 100,000. Additionally, we derive upper bounds on the resolution parameter for which modularity maximization can be equivalent to optimizing the likelihood fit to a degree-corrected stochastic block model. We demonstrate that these bounds hold in practice, providing a priori regions where modularity maximization heuristics "should" be run if a certain number of communities is desired.

To Peter Mucha and all members of the Mucha research group, past, present, and future…

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my thesis advisor, Peter Mucha, for his endless enthusiam and guidance on all things related to network science, navigating the complexities of higher education, and so much more. He fosters an amazing amount of collaboration in his research group and in doing so, has provided the most interesting experiences of my undergraduate career at UNC. Our weekly meetings have been and continue to be the highlight of my week.

A special thank you goes to Saray Shai, who introduced me to the world of network science and encouraged me to get involved in research when I was just a first-year student in her discrete mathematics course. Without her, I would never have met the great people of Dr. Mucha's group and would likely not have delved into the intricacies of networks and their widespread applications.

I extend my thanks to William Weir and his invaluable work on the CHAMP software package. It is greatly appreciated and this project would not have been possible without it.

I am also extremely grateful of every single person in the research group, of which there are too many to name. Your comments and suggestions in our weekly meetings have guided the progress of this thesis through the last few months and all of you have helped this come to fruition in one way or another.

Finally, I'd like to thank my parents, Richard and Sharon, and my brothers, Christian and Robert, for supporting me and promoting my love of computer science and mathematics from a young age. I would not be the person I am today without you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF COMMON ABBREVIATIONS AND SYMBOLS

AMI	Adjusted Mutual Information
NMI	Normalized Mutual Information
CHAMP	Convex Hull of Admissible Modularity Partitions algorithm [26]
SBM	Stochastic Block Model
A_{ij}	Elements of (intralayer) adjacency matrix
C_{ij}	Elements of interlayer adjacency matrix (“interlayer coupling”)
ω_{rs}	Edge propensities in a single-layer, degree-corrected SBM
θ_{rs}	Intralayer edge propensities in a multi-layer, degree-corrected SBM
K	Number of communities in a partition or number of blocks in an SBM
n or N	Number of nodes in a network
m	Number of edges in a network
T	Number of layers in a network
k_i	Degree of (or number of edges connected to) node i
κ_r	Strength of group r, $\kappa_r = \sum_i k_i \delta(g_i, r)$
c_i or g_i	Community or group membership of node i
Q	Modularity
γ	Resolution parameter of single-layer modularity
ω	Interlayer resolution parameter of multi-layer modularity

CHAPTER 1

Introduction to Community Detection

Many real-world data sets can be naturally encoded as a “network” in which entities and their relationships are represented by the nodes and edges in an underlying graph structure, respectively. The use of graph-based analysis has proved to be a powerful tool in biology, computer science, sociology, neuroscience, and many related fields.

One of the most popular techniques for the analysis of networks is community detection (also known as graph or network clustering in some fields). A precise definition of “community” has never been widely accepted, but informally the problem is to partition a network into clusters of nodes that are more densely connected internally than to the rest of the network.

While the interpretation of such community structure is domain-specific, its existence is often of significant interest. For example, in social networks, communities may demarcate the limits of social cliques or groups. In biological networks encoding the relations between genes or proteins, clusters can reveal information about biological pathways and processes. In technological networks, the hierarchical structure of communities (that is, the recursive structure obtained by repeatedly dividing a graph into communities within communities) can be used to compress datasets and detect abnormalities. In computer science, many standard problems such as scheduling work on computing clusters can be naturally reduced to community detection or graph partitioning. For general reviews of community detection’s applications, see [1–4].

Needless to say, interest in community detection is vast and interdisciplinary. Unfortunately, most useful formulations of the problem are NP-Hard and thus intractable to solve exactly. In this chapter, we discuss two popular methods for detecting communities in networks and an important equivalence between them that the rest of this thesis will focus on.

1.1 Modularity-Based Methods

One of the most popular methods for community detection is to heuristically maximize a quantity known as modularity, which for unweighted networks and the standard Newman-Girvan [5] null model is given by

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \gamma \cdot \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)\,, \tag{1.1}$$

where A is the adjacency matrix of the network ($A_{ij} = 1$ when nodes i and j are connected and $A_{ij} = 0$ otherwise), m is the number of edges in the network, k_i is the degree of node i (the number of edges connected to node i), c_i is the community label of node i, and δ is the Kronecker delta function so that $\delta(c_i, c_j) = 1$ when nodes i and j are in the same community (i.e. $c_i = c_j$) and is 0 otherwise.

In Newman and Girvan's original definition, $\gamma = 1$ so that Q gives a measure of how many more edges are observed in the network's communities than would be expected in a random rewiring of its edges. That is, if the communities in a partition are much more densely connected than one would expect by random chance alone, the partition has a "high" value of Q.

The "resolution parameter" γ was added by Reichardt and Bornholdt [6] (similar to an approach proposed by Arenas et al. [7]) to overcome detectability issues in large networks. In particular, when the network is very large in comparison to its communities, modularity with $\gamma = 1$ can fail to detect some community structure (in such cases, merging two communities may increase Q even when the connections between them are very weak [8]).

This addition of the γ prefactor helps resolves this issue by allowing us to detect communities at many different scales – when γ is small, we tend to find a few large communities and as γ increases, we tend to find many smaller communities. In this way, γ can serve as a parameter that chooses the "importance" of the network topology (the A_{ij} term) versus the null model (here, the $\frac{k_i k_j}{2m}$ term).

Exact optimization of modularity is NP-Hard[1] [11] and many limitations are known when the null model does not describe the network well (e.g. when the community sizes vary drastically). Regardless, fast heuristics exist for its maximization (perhaps most notably the Louvain [12] and

[1] In fact, for any constant $\rho > 0$, it is NP-Hard to find a partition of a network with modularity at least $\rho \cdot Q_{\text{opt}}$ where Q_{opt} is the optimal modularity over all possible partitions [9, 10].

Leiden [13] algorithms) and this strategy remains one of the most popular methods for detecting communities in real-world networks.

1.2 Stochastic Block Model Inference

Another popular method for detecting communities is to fit a particular generative model known as a "stochastic block model" to the network of interest. Importantly, this method is statistically principled rather than being ad hoc or motivated through heuristics alone.

In general, one divides a set of n nodes into K groups and denotes the group membership of node i by g_i. Additionally, a matrix Ω is specified whose elements determine the connection strengths between the various communities; the higher the value for $\left(\Omega\right)_{rs} = \omega_{rs}$, the greater the number of edges between groups r and s.

The simplest of these models considers every possible edge between nodes in groups r and s to exist with probability ω_{rs} (and thus, no edge to exist with probability $1 - \omega_{rs}$). In this way, the diagonal of the Ω matrix determines the internal connection densities of the K communities and the off-diagonal terms specify the density of connections between communities.

Unsurprisingly, this simple model does not fit real-world networks well[2] and as with most community detection methods, many different variants exist. For the remainder of this section, we will direct our focus towards the "degree-corrected stochastic block model", which can fit networks with arbitrary degree distributions. Much of the discussion here is adapted from [14, 15], but we've altered the notation to more closely match later parts of the thesis.

In addition to the group assignments and Ω matrix, we assign an expected degree to the nodes of the network such that node i will on average have k_i neighbors. Then, the number of edges between nodes i and j are independently Poisson distributed with mean $\frac{k_i k_j}{2m} \cdot \omega_{g_i g_j}$ or half this value when $i = j$.

Naturally, when fitting such a model to a network, one chooses the observed degree sequence for the k_i's and the observed number of edges for m. With this information, we can determine the probability that a partition of a network with adjacency matrix $\mathbf{A}$ and group assignments $\mathbf{g}$ was

[2] As Newman notes in [14], "there are no good fits when the model you are fitting is simply wrong".

drawn from this stochastic block model by simply considering each possible edge one-by-one:

$$
\begin{aligned}
P(\mathbf{A} \mid \Omega, \mathbf{g}) &= \left[\prod_i \frac{\left(\frac{1}{4m}k_i^2\omega_{g_ig_i}\right)^{A_{ii}/2}}{(A_{ii}/2)!} e^{-\frac{1}{4m}k_i^2\omega_{g_ig_i}}\right] \cdot \left[\prod_{i<j} \frac{\left(\frac{1}{2m}k_ik_j\omega_{g_ig_j}\right)^{A_{ij}}}{A_{ij}!} e^{-\frac{1}{2m}k_ik_j\omega_{g_ig_j}}\right] \\
\ln P(\mathbf{A} \mid \Omega, \mathbf{g}) &= \sum_i \left\{\frac{1}{2}A_{ii}\ln\left(\frac{1}{4m}k_i^2\omega_{g_ig_i}\right) - \ln\left[\,(A_{ii}/2)!\right] - \frac{1}{4m}k_i^2\omega_{g_ig_i}\right\} \\
&\quad + \sum_{i<j}\left[A_{ij}\ln\left(\frac{1}{2m}k_ik_j\omega_{g_ig_j}\right) - \ln(A_{ij}!) - \frac{1}{2m}k_ik_j\omega_{g_ig_j}\right].
\end{aligned}
$$

For the purposes of optimization, we may neglect constants that do not alter the argmax of this expression and simplify this log-likelihood to

$$
\begin{aligned}
\ln P(\mathbf{A} \mid \Omega, \mathbf{g}) &= \sum_i \left\{\frac{1}{2}A_{ii}\ln\left(\frac{1}{4m}k_i^2\omega_{g_ig_i}\right) - \cancel{\ln\left[\,(A_{ii}/2)!\right]} - \frac{1}{4m}k_i^2\omega_{g_ig_i}\right\} \\
&\quad + \sum_{i<j}\left[A_{ij}\ln\left(\frac{1}{2m}k_ik_j\omega_{g_ig_j}\right) - \cancel{\ln(A_{ij}!)} - \frac{1}{2m}k_ik_j\omega_{g_ig_j}\right] \\
&= \sum_{i,j}\left[A_{ij}\ln\omega_{g_ig_j} - \frac{1}{2m}k_ik_j\omega_{g_ig_j}\right] + \cancel{\sum_{i,j}\left[A_{ij}\ln\left(\frac{1}{2m}k_ik_j\right)\right]} \\
\ln P(\mathbf{A} \mid \Omega, \mathbf{g}) &= \sum_{i,j}\left(A_{ij}\ln\omega_{g_ig_j} - \frac{k_ik_j}{2m}\omega_{g_ig_j}\right). \qquad (1.2)
\end{aligned}
$$

Essentially, partitions of a network whose groups exhibit strong community structure consistent with Ω will have “large” log-likelihoods in this equation. Hence, it is possible to heuristically maximize this quantity with respect to Ω and $\mathbf{g}$ to find the most likely set of group assignments under this generative model.

1.3 Newman’s Equivalence Between Modularity Maximization and Maximum Likelihood of a Stochastic Block Model

Newman demonstrated that these two schemes of community detection, modularity maximization and statistical inference based on stochastic block models, become equivalent under certain conditions. In this section, we briefly review his primary results from [14].

Consider a restricted version of the stochastic block model discussed in section 1.2 where the Ω matrix only takes on two values: one shared by all diagonal entries and one shared by all off-diagonal entries. That is,

$$\omega_{rs} = \begin{cases} \omega_{\text{in}}, & \text{if } r = s, \\ \omega_{\text{out}}, & \text{if } r \neq s \end{cases}$$

so that all communities have the same in-group and between-group connection propensities. This particular case is called a "planted partition model".[3] Following Newman, note that this allows us to write

$$\begin{aligned} \omega_{rs} &= (\omega_{\text{in}} - \omega_{\text{out}})\delta(r,s) + \omega_{\text{out}} \\ \ln \omega_{rs} &= \ln\left(\frac{\omega_{\text{in}}}{\omega_{\text{out}}}\right)\delta(r,s) + \ln \omega_{\text{out}} \\ &= \left(\ln \omega_{\text{in}} - \ln \omega_{\text{out}}\right)\delta(r,s) + \ln \omega_{\text{out}}\,. \end{aligned}$$

We can use these equations to rewrite our objective function from Equation 1.2 for optimizing the log-likelihood that an observed graph fits a degree-corrected SBM,

$$\begin{aligned} \ln P(\mathbf{A} \mid \Omega, \mathbf{g}) &= \sum_{i,j}\left(A_{ij} \ln \omega_{g_i g_j} - \frac{k_i k_j}{2m}\omega_{g_i g_j}\right) \\ &= \sum_{i,j}\left\{A_{ij}\left[\ln\left(\frac{\omega_{\text{in}}}{\omega_{\text{out}}}\right)\delta(g_i,g_j) + \ln \omega_{\text{out}}\right] - \frac{k_i k_j}{2m}\left[(\omega_{\text{in}} - \omega_{\text{out}})\delta(g_i,g_j) + \omega_{\text{out}}\right]\right\} \\ &= \ln\left(\frac{\omega_{\text{in}}}{\omega_{\text{out}}}\right)\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\cdot\frac{\omega_{\text{in}} - \omega_{\text{out}}}{\ln \omega_{\text{in}} - \ln \omega_{\text{out}}}\right]\delta(g_i,g_j) \\ &\quad + \sum_{i,j}\left[A_{ij} \ln \omega_{\text{out}} - \frac{k_i k_j}{2m}\omega_{\text{out}}\right]\,. \end{aligned}$$

Once again, we may ignore constants that do not affect optimization to obtain

$$\begin{aligned} \ln P(\mathbf{A} \mid \Omega, \mathbf{g}) &= \cancel{\ln\left(\frac{\omega_{\text{in}}}{\omega_{\text{out}}}\right)}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\cdot\frac{\omega_{\text{in}} - \omega_{\text{out}}}{\ln \omega_{\text{in}} - \ln \omega_{\text{out}}}\right]\delta(g_i,g_j) \\ &\quad + \sum_{i,j}\left[\cancel{A_{ij} \ln \omega_{\text{out}}} - \cancel{\frac{k_i k_j}{2m}\omega_{\text{out}}}\right] \\ &= \sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\cdot\frac{\omega_{\text{in}} - \omega_{\text{out}}}{\ln \omega_{\text{in}} - \ln \omega_{\text{out}}}\right]\delta(g_i,g_j)\,, \end{aligned}$$

[3] More commonly, "planted partition" refers to the case in which nodes are connected with probabilities p_{in} and p_{out} (depending on community membership), but we are considering a degree-corrected version here.

Note the striking resemblance to modularity! In particular, maximizing this expression is exactly the same as maximizing

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \gamma \cdot \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

when

$$\gamma = \frac{\omega_{\text{in}} - \omega_{\text{out}}}{\ln \omega_{\text{in}} - \ln \omega_{\text{out}}} \,. \tag{1.3}$$

In this way, this choice of γ is the "correct value" of the resolution parameter if we wish to make modularity maximization equivalent to the maximum likelihood fit of a planted partition, degree-corrected stochastic block model. We will often call this the "γ estimate" or "resolution parameter estimate" of a partition.

Additionally, Newman gives an iterative procedure to find this correct choice of γ. First, note that the expected number of within-community edges in this model is

$$m_{\text{in}} = \frac{1}{2} \sum_{i,j} \left[\frac{k_i k_j}{2m} \cdot \omega_{\text{in}} \cdot \delta(g_i, g_j) \right] = \frac{\omega_{\text{in}}}{4m} \sum_r \kappa_r^2 \,,$$

where $\kappa_r = \sum_i k_i \delta(g_i, r)$ is the sum of the degrees of all nodes in group r. Then, we can estimate

$$\omega_{\text{in}} = \frac{2m_{\text{in}}}{\sum_r \kappa_r^2/(2m)}, \qquad \omega_{\text{out}} = \frac{2m_{\text{out}}}{\sum_{r \neq s} \kappa_r \kappa_s/(2m)} = \frac{2m - 2m_{\text{in}}}{2m - \sum_r \kappa_r^2/(2m)} \,. \tag{1.4}$$

Thus, with an initial guess for γ, we can repeatedly maximize modularity *with the number of communities fixed* and compute new estimates for ω_{in} and ω_{out}. This gives a new value for γ and we repeat until convergence. Moreover, when considering networks drawn from a planted partition model, this scheme is guaranteed to converge in the limit of large node degrees.

For real-world networks (which are obviously not drawn from such planted partitions), no such guarantee can be made, but the procedure still appears to be efficient in practice.

We emphasize that this equivalence only holds if the number of communities is fixed during the maximization of modularity. Otherwise, maximizing modularity is akin to the simultaneous maximum likelihood fit between many different SBMs, each with a different number of blocks K and potentially different parameters ω_{in} and ω_{out}.[4] Unsurprisingly, this method of statistical inference

[4] In this case, it is not exactly the same as the simultaneous maximum likelihood fit, but we defer a more in-depth discussion to section A.5.

among many different SBMs is not common in the literature (though some recent work has focused on choosing priors that allow for comparison between different models [16]) and is one of the reasons that modularity maximization is not used to infer the number of communities in a network. Indeed, in discussing the equivalence, Newman notes that

> "Maximization of modularity with [the number of communities] allowed to vary does not, in general, give good estimates of the number of communities in a network, and it is certainly possible that we get different and incorrect numbers of communities were [the number of communities] allowed to vary." [14]

Unfortunately, in practice the number of communities in a network is not known a priori. Indeed, one of the goals of community detection is to find statistically significant clusters of nodes without considering how many exist in the system that the network is drawn from.

As such, the most widely used heuristics (e.g. the Louvain algorithm [12]) do not keep the number of communities fixed. We will return to this issue in section 2.5 and section A.5.

CHAPTER 2

The Use of Newman's Parameter Estimation in Pamfil et al.

Pamfil et al. [17] generalized Newman's [14] original equivalence to several variants of "multi-layer networks" in which a collection of interrelated networks are treated as individual "layers" in a larger, connected network. In this chapter, we briefly describe this extension of Newman's duality to multi-layer settings.

As usual, they are many different formulations of multi-layer networks, but Pamfil et al.'s extension focuses on three particular types: "temporal", "multilevel", and "multiplex" networks. Under certain choices of resolution parameters, an extension of modularity to multi-layer settings can be shown to be equivalent to statistical inference using multi-layer stochastic block models.

2.1 Multi-layer Modularity

Consider a set of T layers of $n \times n$ adjacency matrices $\mathbf{A}^t$, $1 \leq t \leq T$, each representing the same set of n nodes. Further introduce a set of interlayer couplings C^{st}, one for each pair of distinct layers $1 \leq s, t \leq T$ such that node j in layer s is connected to itself in layer r with weight C_j^{sr}.

Then, the goal of community detection is to determine group membership per node and per layer, so e.g. we must find the group assignment g_i^s of node i in layer s.[1] Under this framework, Mucha et al. [18] derived an extension of modularity to multi-layer settings:

$$Q = \frac{1}{2\mu} \sum_{ijsr} \left[\left(A_{ij}^s - \gamma_s \frac{k_i^s k_j^s}{2m_s} \right) \delta(s, r) + C_j^{sr} \delta(i, j) \right] \delta\left(g_i^s, g_j^r\right) \ , \tag{2.1}$$

[1] In the original paper, all variables are indexed with subscripts, so the group assignment of node i in layer s is denoted g_{is}, but we choose to use the notation g_i^s of Pamfil et al. [17] here to simplify the discussion in later sections.

where k_i^s is the degree of node i in layer s, m_s is the number of edges in layer s, and $2\mu = \sum_{is}\left(k_i^s + \sum_r C_i^{sr}\right)$ is twice the sum of all the network's edge weights. Note that in this formulation, each layer has a different "intralayer resolution parameter" with the weighting of the null model in layer s being controlled by γ_s.

Mucha et al. [18] also introduced an interlayer resolution parameter ω to control the importance of the interslice couplings in Equation 2.1. Similar to the usage of γ_s, one could include this as a multiplicative factor on the C_j^{st} term, but the original formulation takes the equivalent approach of absorbing ω into the definition of the weights of C^{st}. For simplicity, the interslice couplings were taken to be binary so that this resolution parameter ω appears as

$$C_j^{st} = \begin{cases} \omega, & \text{if node } j \text{ is connected between layers } s \text{ and } t \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

This particular choice is known as uniform (interslice) coupling [19] since the weights of the interslice couplings are identical across all layers.

2.2 Temporal Networks

Temporal networks are those in which each layer encodes interactions during some period or instance of time. Hence, the layers can be placed in chronological order and a larger graph can be created by connecting each node to its copy in the preceding and subsequent layers ("ordinal coupling" between layers). Other interlayer connection strategies exist, but when the literature refers to "temporal networks", it is almost always the case that interlayer connections are strictly ordinal [20].

2.2.1 Uniform Coupling

Consider a network with T layers where layer t, $1 \leq t \leq T$ has adjacency matrix $\mathbf{A}^t$. Then, where the interlayer edges between layers $t-1$ and t all have weight ω_t, this network has a "supra-adjacency matrix” given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^1 & \omega_2\mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^2 & \omega_3\mathbf{I} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \omega_t\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}^t \end{bmatrix}$$

so that each block on the diagonal represents intralayer connections and each nonzero off-diagonal block represents connections between adjacent layers (directed so as to represent the flow of information forward in time, though this is equivalent to having half the weight in both directions). In this way, node i in layer t is represented in the supra-adjacency matrix by node $n(t-1)+i$.

We will initially restrict our focus to the uniform interslice coupling case where $\omega_t = \omega$ for all layers t. Then, the setup of the underlying multi-layer SBM is a fairly straightforward extension of Newman’s strategy [14] for the duality in the monolayer case.

As before, consider the intralayer connections in layer t given by $\mathbf{A}^t$ to be drawn from a degree-corrected, planted partition stochastic block model with K blocks. In this SBM, the within-community edge propensities are given by θ_{in} and between-community edge propensities are given by θ_{out}.

Note that these edge propensities θ_{in} and θ_{out} are directly analogous to the values of ω_{in} and ω_{out} from our discussion of the monolayer duality in section 1.3. However, ω is already used to represent the interlayer resolution parameter in Equation 2.1 and Equation 2.2, so to avoid confusion we adopt this $\theta_{\text{in}}, \theta_{\text{out}}$ notation from Pamfil et al. [17].

The underlying SBM model further assumes that labels are copied between layers with “copying probability” p. That is, the ground truth group assignment g_i^t of node i in layer t is copied from layer $t-1$ with probability p and is assigned randomly according to a null distribution $\mathbb{P}_0$ with probability $1-p$. For the purposes of the equivalence here, this null distribution $\mathbb{P}_0$ is taken to be uniform across all possible community labels $1, \dots, K$.

Under this model, consider a partition $\mathbf{g}$ of the multi-layer network where g_i^t is the group membership of node i in layer t. Then, neglecting constants that do not affect optimization, we can write out the posterior probability of $\mathbf{g}$ in our model[2]

$$\begin{aligned}\ln P(\mathbf{g} \mid \mathbf{A}, \theta_{\text{in}}, \theta_{\text{out}}, p, K) = &\sum_{t=1}^{T} \sum_{i,j=1}^{N} \left(A_{ij}^t - \frac{\theta_{\text{in}} - \theta_{\text{out}}}{\ln \theta_{\text{in}} - \ln \theta_{\text{out}}} \cdot \frac{k_i^t k_j^t}{2m_t} \right) \delta\left(g_i^t, g_j^t\right) \\ &+ \sum_{t=2}^{T} \sum_{i=1}^{N} \frac{\ln\left(1 + \frac{p}{1-p} K\right)}{\ln \theta_{\text{in}} - \ln \theta_{\text{out}}} \cdot \delta\left(g_i^{t-1}, g_i^t\right) \ ,\end{aligned}$$

where as before, k_i^t is the degree of node i in layer t and m_t is the number of edges in layer t. Note the resemblance to multi-layer modularity – maximizing this expression is exactly the same as maximizing multi-layer modularity

$$Q = \frac{1}{2\mu} \sum_{t=1}^{T} \sum_{i,j=1}^{N} \left(A_{ij}^t - \gamma \cdot \frac{k_i^t k_j^t}{2m_t} \right) \delta\left(g_i^t, g_j^t\right) + \sum_{t=2}^{T} \sum_{i=1}^{N} \omega \cdot \delta\left(g_i^{t-1}, g_i^t\right) \tag{2.3}$$

when

$$\gamma = \frac{\theta_{\text{in}} - \theta_{\text{out}}}{\ln \theta_{\text{in}} - \ln \theta_{\text{out}}} \qquad \text{and} \qquad \omega = \frac{\ln\left(1 + \frac{p}{1-p} K\right)}{\ln \theta_{\text{in}} - \ln \theta_{\text{out}}} \ , \tag{2.4}$$

where we have rewritten Q from Equation 2.1 with uniform interlayer and intralayer resolution parameters to make the equivalence more obvious.

Hence, these γ and ω are the "correct values" of the intralayer and interlayer resolution parameters if we want multi-layer modularity maximization to be equivalent to the maximum likelihood fit of the SBM considered here. As before, we will call the values in Equation 2.4 the "γ estimate" and "ω estimate" (together, "resolution parameter estimates") of a partition.

θ_{in} and θ_{out} are estimated in much the same way as we estimated ω_{in} and ω_{out} in the monolayer case, with the added restriction that group memberships are considered per-layer rather than in aggregate.[3]

The copying probability p of labels from one layer to the next is simply estimated using the observed frequency with which the group membership of node i persists across layers – i.e. we can estimate p by calculating the probability that $g_i^{t-1} = g_i^t$ over all layers $t = 2, \ldots, T$ and all nodes $i = 1, \ldots, N$.

[2] Some of the simplifications required here are particularly nontrivial, so we direct the interested reader to the original description in [17] for the complete derivation.

[3] I.e. the community strengths in layer t are computed as $\kappa_r^t = \sum_i k_i^t \delta\left(g_i^t, r\right)$ and then normalized using the number of edges per layer in Equation 1.4 with $\sum_r \kappa_r^2/(2m_t)$.

Then as in the monolayer case in section 1.3, we can iteratively find "correct" values for (γ, ω) by maximizing modularity with the number of communities fixed, computing new estimates for γ and ω, and then repeating until convergence.

2.2.2 Non-Uniform Coupling

A similar result can be derived when considering non-uniform coupling where the intralayer SBM parameters θ^t_{in}, θ^t_{out}, number of communities per layer K_t, intralayer resolution parameters γ_t, interlayer resolution parameters ω_t, and copying probabilities p_t vary over the layers t. We defer that discussion to the original paper in [17], but for the sake of completeness, the equivalence in this case is to multi-layer modularity with added intralayer weights β_t (cf. Equation 2.3)

$$Q = \frac{1}{2\mu} \sum_{t=1}^{T} \beta_t \sum_{i,j=1}^{N} \left(A^t_{ij} - \gamma_t \cdot \frac{k^t_i k^t_j}{2m_t} \right) \delta\left(g^t_i, g^t_j\right) + \sum_{t=2}^{T} \sum_{i=1}^{N} \omega_t \cdot \delta\left(g^{t-1}_i, g^t_i\right)$$

and the correct choices of γ_t, ω_t, and β_t are given by

$$\gamma_t = \frac{\theta^t_{\text{in}} - \theta^t_{\text{out}}}{\ln \theta^t_{\text{in}} - \ln \theta^t_{\text{out}}}, \qquad \omega_t = \frac{\ln\left(1 + \frac{p_t}{1-p_t} K_t\right)}{\langle \ln \theta^t_{\text{in}} - \ln \theta^t_{\text{out}} \rangle_t}, \qquad \beta_t = \frac{\ln \theta^t_{\text{in}} - \ln \theta^t_{\text{out}}}{\langle \ln \theta^t_{\text{in}} - \ln \theta^t_{\text{out}} \rangle_t} \tag{2.5}$$

where $\langle \cdot \rangle_t$ denotes a mean across all layers.

2.3 Multilevel Networks

A multilevel network is an extension of temporal networks in which the interactions between layers encode a hierarchy of relationships [21]. For example, individuals may work in a department of a company in a certain sector the economy and this inclusion could be modeled as interlayer edges between four different layers (individual, department, company, and sector of economy). In contrast to temporal networks, these interlayer relationships need not represent an ordinal flow (e.g. across time) and nodes connected across layers do not necessarily represent the same entity.

Importantly, the hierarchy of relationships admits a natural ordering of layers and the copying probability model from temporal networks in section 2.2 can be adapted for use here. In particular, the group memberships need to be copied from parent to child across layers to ensure that at most a single connection exists to each node as the labels are copied from layer to layer.

Other than this tweak to copying probabilities, the derivation is very similar to temporal networks. Indeed, the correct choices for resolution parameters match exactly with the temporal case! For uniform coupling, the "correct γ and ω estimates" are given in Equation 2.4 and for non-uniform coupling, the "correct parameters" are as in Equation 2.5.

2.4 Multiplex Networks

Multiplex networks are those in which the edges between nodes are categorized by type. Then, the edges of each type are embedded into layers of a larger network with exactly one type of edge per layer [20]. Here, the interactions between layers are taken to be fully connected.

The precise details of the derivations for the multiplex case are complicated and we will only briefly summarize the methods and results here. Once again, we direct the interested reader to the full description in the original paper [17].

Unfortunately, these networks do not admit any natural ordering of layers and so the copying probability approach from the temporal and multilevel network models does not directly apply here. In fact, the approach taken in Pamfil et al. [17] is essentially to consider label copying over all possible orderings of the layers.

Indeed when the network has T layers, the derived expression for the optimal interlayer resolution parameter ω_{st} between layers s and t requires a sum over the symmetric group S_{T-1} in which each summand individually involves a product over all layers $t = 2, \ldots, T$. Needless to say, this sum has $(T-1)!$ terms and is only tractable for networks with an extremely small number of layers,[4] though it is not uncommon to have only a few layers in some real-world settings.

However, simplifications can be made when the copying probabilities p_{st} between all pairs of layers $s, t \in \{1, \ldots, T\}$ are taken to have the same value $p_{st} = p$ and all possible orderings of the layers are sampled with equal probability $1/(T!)$. In this case, the problematic sum over S_{T-1} can

[4] The rate of growth of the factorial function is often not fully appreciated – over the course of a few days, current high end desktop CPUs can perform about 19! operations, high end GPUs or computational accelerators can perform about 21! operations, and the global computing capacity can perform about 26! operations.

be approximated to obtain

$$\omega = \frac{\ln\left(1 + \frac{p}{1-p}K\right)}{T\langle \ln\theta_{\text{in}} - \ln\theta_{\text{out}}\rangle_t}$$

as the optimal value for the interlayer resolution parameter in uniform multiplex networks. Note that this estimate is equivalent to dividing the ω estimate from temporal and multilevel models by the number of layers T (cf. Equation 2.5). Informally, this scaling by a factor of T corrects for the T-fold increase in number of interlayer edges when compared to temporal networks.[5]

Moreover, it is not straightforward to estimate p in the multiplex case since one must consider all permutations of the layers individually. The estimate of persistence of group labels must be considered across all pairs of layers

$$P(g_i^s = g_i^t) \approx \frac{1}{NT(T-1)} \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{s \neq t} \delta\left(g_i^s, g_i^t\right) \tag{2.6}$$

and it can be shown that the expected probability of $g_i^s = g_i^t$ under all possible permutations of layers is given by

$$P(g_i^s = g_i^t) = \frac{2\left(1 - \frac{1}{K}\right)}{T(T-1)} \sum_{n=1}^{T-1} p^n (T-n) + \frac{1}{K}\,. \tag{2.7}$$

Hence, we can estimate p here by equating the right-hand sides of Equation 2.6 and Equation 2.7 and solving the resulting polynomial root-finding problem.

This calculation only works when all copying probabilities take the same value $p_{st} = p$ as before. A tractable method for accurately approximating p_{st} when this is not the case remains an open problem.

[5] Due to the ordinal coupling in temporal networks with T layers and N nodes per layer, such networks have $N(T-1)$ interlayer edges. However, multiplex networks here have fully connected interlayer interactions and thus have $N \cdot T(T-1)$ interlayer edges.

2.5 Differences with Newman’s Scheme for Determining the “Correct Values” of Resolution Parameters

Importantly in Newman’s equivalence [14] and the extensions described in this chapter, the duality between modularity maximization and maximum likelihood to a stochastic block model only holds when the number of communities, K, is fixed.

In iterating until the resolution parameter estimates converge, Pamfil et al. [17] maximizes modularity using the GenLouvain algorithm [22]. Notably, this algorithm is based on the Louvain procedure of [12] and thus does not keep the number of communities fixed. This poses three primary issues in practice.

1. Due to a significantly larger search space, modularity maximization algorithms that allow K to vary tend to be much more stochastic than those that fix K. This can cause problems with the convergence of the iterative resolution parameter estimation.

2. When not fixing K, we will discover multiple “correct” values for the resolution parameters far more frequently than when K is fixed, depending on the initialization values (e.g. when two fixed points of the iterative scheme correspond to different K). Indeed, one of the nice guarantees of Newman’s original procedure was that it would eventually “converge to the correct value of γ (and the correct community structure) for networks that are actually generated from a planted partition model (in the limit of large node degrees)” [14].

3. In regions where modularity maximization heuristics do not consistently return partitions with the same number of communities, the resulting resolution parameter estimates vary wildly and can lead to convergence problems. In fact, this can “hide” significant partitions when heuristics do not effectively estimate K.

We now demonstrate these issues on the karate club network of Zachary [23] which is one of the simplest and most popular examples of community structure in the network science literature. This network describes the social relationships between individuals in a university karate club shortly before a disagreement between an administrator and the instructor split the group in two.

First, we analyze the results of the iterative scheme to find γ estimates when using two different modularity maximization algorithms: the Louvain algorithm [12] as implemented by Vincent Traag in [24] (which does not restrict the number of communities K) and the spin glass algorithm of Reichardt and Bornholdt [6] as implemented in igraph [25] (where we restrict $K = 2$ as in Newman's paper [14]). The behavior of γ estimation under these two algorithms is shown in Figure 2.1.

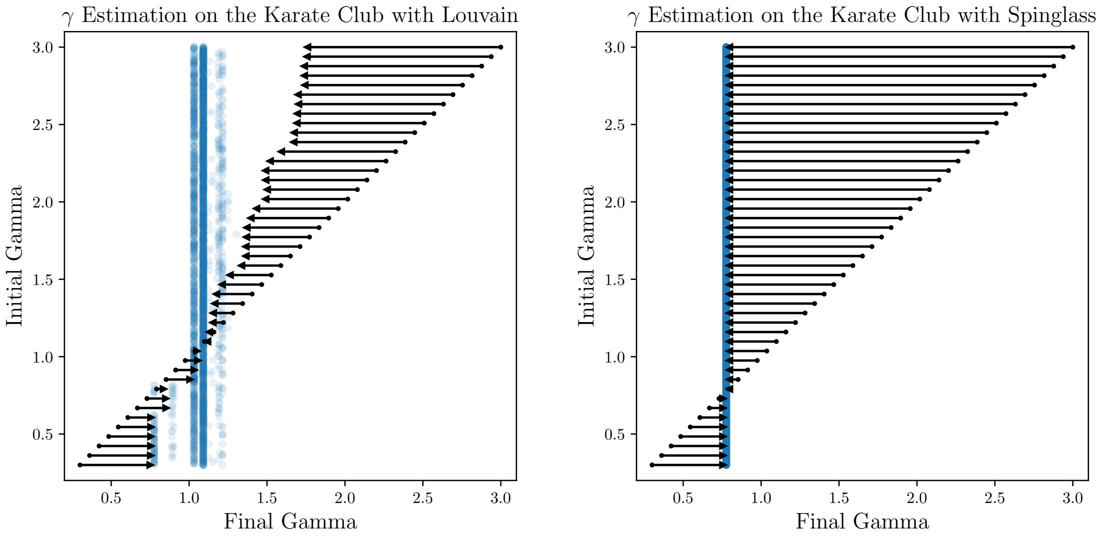


Figure 2.1: Behavior of the iteration to determine "correct" values for the resolution parameter γ. Arrows show the average movement (over 100 trials) induced in γ space where the base of the tail lies at the initialization γ and the head of the tail lies at the resulting γ estimate after one iteration. Pairs of (initialization γ, final converged γ) obtained across many runs as shown in blue.
Left: Behavior of the scheme when using the Louvain algorithm [12], which does not fix K. Right: Behavior of the scheme when using the spin glass algorithm [6] restricted to finding $K = 2$ communities.

First, note that the stochasticity is greatly increased when using Louvain to maximize modularity. Newman [14] found that his iterative scheme consistently converged to finding an optimal estimate $\gamma \approx 0.78$ where a 2-community partition has highest modularity. This matches the behavior we see when using the spin glass algorithm where convergence to $\gamma \approx 0.78$ occurs after the first iteration of the scheme, regardless of initialization γ.

However, when using Louvain, the scheme more frequently converges to an estimate of $1.0 \leq \gamma \leq 1.1$, where a 4-community partition has highest modularity. Indeed, convergence to $\gamma \approx 0.78$ only occurs when the iterative procedure is initialized with a very small γ value. Even then, the algorithm still usually returns an estimate of $1.0 \leq \gamma \leq 1.1$.

Second, while the procedure using the spin glass algorithm always returns the same final resolution parameter estimate, the procedure using Louvain finds multiple “correct values”. Most frequently, a 4-community partition with $1.0 \leq \gamma \leq 1.1$ is returned, but the 2-community partition with $\gamma \approx 0.78$ from Newman’s experiments [14] and a 3-community partition with $\gamma \approx 0.9$ are occasionally found as well.

Indeed, there is a strong dependence of a partition’s γ estimate and its number of communities, which causes much of the inconsistency when using Louvain here. We show this dependency on the karate club in Figure 2.2.

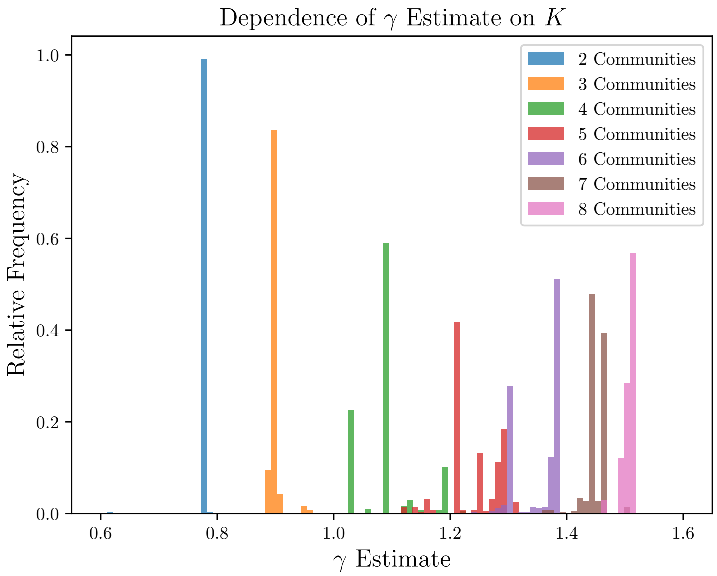


Figure 2.2: Frequency of γ estimates on the Karate Club from 1,000,000 runs of Louvain across a uniform grid of $\gamma \in [0.0, 2.0]$. Relative frequencies are given in terms of all observed partitions with the same number of communities K.

For example, if Louvain “chooses” at random between a 2-community or 3-community partition, the iterative scheme will randomly continue with $\gamma \approx 0.78$ or $\gamma \approx 0.9$.

In hindsight, this is unsurprising – the duality between modularity optimization and stochastic block model inference depends on estimates of the SBM parameters, which may differ greatly when the number of blocks is changed.[6] Hence, in general, the results of resolution parameter estimation will strongly depend on the number of communities returned by the modularity maximization heuristic of choice.

[6] We will discuss the duality more in appendix A and will analyze models in which multiple meaningful ground-truth partitions exist with different number of blocks K in appendix B.

CHAPTER 3

Introduction to the CHAMP Algorithm and Partition Post-Processing

Weir et al. described a method in [26] for pruning a set of partitions of a network to a subset in which each partition has higher quality than all others for some choice of resolution parameter values. The motivation and “CHAMP” (Convex Hull of Admissible Modularity Partitions) algorithm are briefly summarized here.

3.1 Linearity of Modularity

Initially, we’ll restrict our focus to the single-layer notion of modularity. This is given by

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \gamma P_{ij} \right] \delta(c_i, c_j) \,,$$

where the “null model” is typically taken to be the configuration null model so that $P_{ij} = \frac{k_i k_j}{2m}$.

We often run modularity maximization heuristics at many different values of γ to detect communities at various scales, especially when a ground truth number of communities is unknown. In these cases, runs of the heuristics at one value of γ are treated completely independently from those at a different value $\gamma' \neq \gamma$.

However, we can rewrite the equation for modularity to gain information about each partition across many values of the resolution parameter, regardless of the original value for which the modularity maximization heuristic was run. Ignoring leading multiplicative constants (which do not affect optimization), the modularity of a fixed partition σ is given by

$$\begin{aligned} Q_\sigma(\gamma) &= \sum_{i,j} \left[A_{ij} - \gamma P_{ij}\right] \delta(c_{i,\sigma}, c_{j,\sigma}) \\ &= \sum_{i,j} A_{ij} \cdot \delta(c_{i,\sigma}, c_{j,\sigma}) - \gamma \sum_{i,j} P_{i,j} \cdot \delta(c_{i,\sigma}, c_{j,\sigma}) \\ &= \hat{A}_\sigma - \gamma \hat{P}_\sigma \, , \end{aligned}$$

where $\hat{A}_\sigma$ and $\hat{P}_\sigma$ are the within-community sums of A_{ij} and P_{ij} respectively. Note that for a fixed partition σ, these values are constant, so Q_σ is linear in γ.

A similar result holds for multi-layer networks where we introduce an additional interlayer connections matrix C and interlayer resolution parameter ω. In this case, (ignoring multiplicative constants), modularity is given by

$$\begin{aligned} Q_\sigma(\gamma, \omega) &= \sum_{i,j} \left[A_{ij} - \gamma P_{ij} + \omega C_{ij}\right] \delta(c_{i,\sigma}, c_{j,\sigma}) \\ &= \hat{A}_\sigma - \gamma \hat{P}_\sigma + \omega \hat{C}_\sigma \, , \end{aligned}$$

where $\hat{A}_\sigma$, $\hat{P}_\sigma$, and $\hat{C}_\sigma$ are the within-community sums of A_{ij}, P_{ij} and C_{ij} respectively. Importantly, this is linear in both resolution parameters γ and ω.

In other words, each partition of a single-layer network is represented by a line in (γ, Q) space and each partition of a multi-layer network is represented by a plane in (γ, ω, Q) space. In this way, we can consider the quality of a partition across many different resolution parameter values.

This generalizes in a straightforward way to the case in which we have an arbitrary number of resolution parameters. If we have a form of modularity written as

$$\begin{aligned} Q_\sigma(\gamma_1, \dots, \gamma_k) &= \sum_{i,j} \left[A_{ij} - \gamma_1 P_{1,i,j} - \gamma_2 P_{2,i,j} - \cdots - \gamma_k P_{k,i,j}\right] \delta(c_{i,\sigma}, c_{j,\sigma}) \\ &= \hat{A}_\sigma - \gamma_1 \hat{P}_{1,\sigma} - \gamma_2 \hat{P}_{2,\sigma} - \cdots - \gamma_k \hat{P}_{k,\sigma} \, , \end{aligned}$$

then each partition is represented by a hyperplane in $(\gamma_1, \dots, \gamma_k, Q)$ space.

3.2 Definitions

Consider a network and a set of partitions $\Sigma = \{\sigma_1, \ldots, \sigma_s\}$ of this network. When modularity has resolution parameters $\gamma_1, \ldots, \gamma_k$, we introduce the following terminology:

- A partition $\sigma \in \Sigma$ is "dominant" or "optimal" at $(\gamma_1, \ldots, \gamma_k)$ if $Q_\sigma(\gamma_1, \ldots, \gamma_k) \geq Q_{\sigma'}(\gamma_1, \ldots, \gamma_k)$ for all $\sigma' \in \Sigma$.

- A partition $\sigma \in \Sigma$ is "somewhere optimal" if it is optimal for some choice of $(\gamma_1, \ldots, \gamma_k)$. Otherwise, we say σ is "nowhere optimal".[1]

- The "domain of optimality" or "domain of dominance" of $\sigma \in \Sigma$ is the convex polytope of values $(\gamma_1, \ldots, \gamma_k)$ for which σ is dominant. This domain is nonempty if and only if σ is somewhere optimal.

For example, the domain of optimality of a single-layer partition is a (potentially empty) contiguous range of γ values. Similarly, the domain of optimality of a multi-layer partition is a convex polygon in the (γ, ω) plane.

3.3 The CHAMP Algorithm

Consider a set of partitions $\Sigma = \{\sigma_1, \ldots, \sigma_s\}$ of a network. For any particular choice of resolution parameter, one might be interested in the partition $\sigma_i \in \Sigma$ that has maximal modularity.

The CHAMP algorithm provides a method for determining the set of optimal partitions at all values for the resolution parameters at once while simultaneously determining each partition's domain of optimality.

This is done by noting that the optimal partition in Σ for any given choice of resolution parameters $(\gamma_1, \ldots, \gamma_k)$ is the one with greatest modularity value $Q(\gamma_1, \ldots, \gamma_k)$. Hence, the set of all partitions that are somewhere optimal can be described by the piecewise hyperplanar upper envelope of the hyperplanes in $(\gamma_1, \ldots, \gamma_k, Q)$ space that represent the partitions in Σ.

In this way, finding the set of all partitions that are somewhere dominant is equivalent to finding the region in $(\gamma_1, \ldots, \gamma_k, Q)$ space that is above all the partitions' hyperplanes. That is, if we consider

[1] In the original paper [26], "admissible" is used in place of "somewhere optimal".

for each partition σ_i, the halfspace defined by the region "above" its hyperplane in $(\gamma_1, \ldots, \gamma_k, Q)$ space (i.e. the region of larger Q values for all choices of resolution parameters), then our problem reduces to "halfspace intersection".

This is precisely the method that CHAMP uses to construct the subset of Σ consisting of somewhere dominant partitions and their associated domains of optimality. This procedure is visually summarized in Figure 3.1.

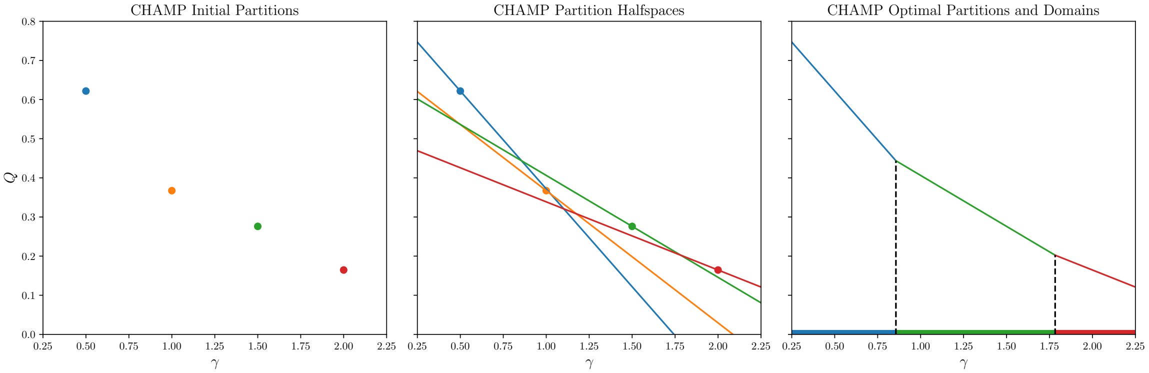


Figure 3.1: Summary of the CHAMP algorithm for single-layer networks with one resolution parameter γ. Left: Partitions are sampled in (γ, Q) space, using a modularity maximization heuristic. Center: The partitions are considered as halfspaces in (γ, Q) space. Right: The intersection of the halfspaces is obtained and the resulting facets are projected into the space of resolution parameters to obtain the set of somewhere dominant partitions and their domains of optimality.

Halfspace intersection is a well studied computational geometry problem and efficient algorithms exist for solving it (provided the dimensionality of the space is not too large). In particular, CHAMP's implementation at [27] uses the software package Qhull which "may be used for 2-d up to 8-d", according to the authors on qhull.org [28]. Crucially, for single-layer and multi-layer network analysis (where only 2 and 3 dimensions are needed, respectively), the worst case running time of Qhull is $O(n \lg n)$ for n input partitions.

3.4 Summary of CHAMP's Benefits

Primarily, CHAMP provides a method for pruning a large number of partitions of a network (potentially taken from a vast range of the resolution parameter space and from various computational heuristics) into the small subset of partitions that are somewhere dominant.

Importantly, each partition is treated as a hyperplane and not just as a single point in the resolution parameter space to take full advantage of the input partitions. Indeed, if a partition were returned by a computational heuristic running at resolution parameter γ, it is possible for that partition to be dominant for many values $\gamma' \neq \gamma$. Moreover, we have found that the most common partitions returned by the Louvain algorithm at γ are generally less optimal than the highest quality partition obtained by sweeping over nearby $\gamma' \approx \gamma$.

In the original CHAMP paper, the pruned subsets were often observed to be several orders of magnitude smaller than the full set of partitions,[2] though this obviously depends on the size and variance in quality of the set of partitions to be pruned.

Moreover, since this pruning problem reduces to halfspace intersection, the method is very quick for practical resolution parameter spaces where the dimensionality is small. In fact, taking the intersection of millions of random halfspaces in $\mathbb{R}^3$ takes only a few seconds – compared to running modularity maximization heuristics, this amounts to a vanishingly small fraction of the execution time of community detection pipelines.

[2] The original paper focused on examples where "pruned subsets of admissible partitions [were] 20-to-1785 times smaller than the sets of unique partitions [...] input into CHAMP" [26].

CHAPTER 4

Using CHAMP with the Parameter Estimation Map

4.1 Stability Under the Resolution Parameter Estimation Map with CHAMP

If we use CHAMP [26] to prune a set of partitions, we can consider the behavior of the resolution parameter estimation procedures on each of the somewhere dominant partitions. Here, by passing through Newman's duality from [14], note that restricting focus to somewhere dominant partitions is equivalent to only comparing those partitions that have a maximum likelihood fit to a (planted partition, degree corrected) stochastic block model as the parameters vary.[1]

Crucially, the resolution parameter estimation on this pruned subset of partitions is completely deterministic and discrete.

For instance, consider a set of somewhere dominant partitions of a multi-layer network so that the convex 2D domains of optimality tile the (γ, ω) plane. Then, since each partition admits an estimate for its "correct" (γ, ω) parameters, we can consider the domain of optimality in which these parameters lie. This defines a discrete map between all domains given by the movements through the resolution parameter space induced by our estimates.

Then, a partition whose resolution parameter estimate lies within its domain of optimality is analogous to a fixed point of the iterative estimation procedures from Newman [14] and Pamfil et al. [17]. In this case, we will say that the partition is "stable under the resolution parameter estimation map" with respect to the pruned subset of partitions or (for the purposes of brevity) that the partition is "stable".

[1] In general, there may be some concerns about potentially removing important partitions through pruning with CHAMP. However, the quality of partitions considered for the iterative resolution parameter estimation are directly related to the likelihood of the underlying fit to an SBM and thus, the quality of the resulting γ estimate. Hence, we believe that it is reasonable to ignore the nowhere dominant partitions here.

4.2 Benefits Gained from Using CHAMP

First, recall precisely what it means for a partition to be stable under our map here. Such a partition has greater quality than all other partitions of interest at the value of the resolution parameter where modularity maximization becomes equivalent to (planted partition, degree corrected) stochastic block model inference. In short, stability of a partition under the resolution parameter estimation map signifies that it is "statistically significant" from the perspective of SBM inference.

The benefit of using CHAMP here is many fold.

4.2.1 Suppression of Stochasticity in Parameter Estimation

First, by suppressing the stochasticity of modularity maximization heuristics during iterative resolution parameter estimation, we no longer have to worry about issues of randomness while determining stability or fixed points. Of course, we have to pay for this by actually finding a set of partitions to prune in the first place, but testing a partition for stability now becomes as easy as finding self loops in a graph.

4.2.2 Statistically Principled When Restricting the Number of Communities

Second, we may regain the statistical grounding of Newman's [14] scheme that Pamfil et al.'s [17] lacks by restricting our focus to partitions with a fixed number of communities through CHAMP. Then, the duality between modularity maximization and stochastic block model inference is well principled once again.

We obviously cannot guarantee that the results will be exactly the same as with heuristics that keep the number of communities fixed since we're ultimately at the mercy of our community detection algorithm of choice. However, if the input set of partitions contains some reasonably high quality K-community partitions, then the behavior of our scheme should not differ greatly from Newman's original proposal. Indeed, in practice when networks have strong K-block structure, we have found that the Louvain algorithm detects partitions with K communities for fairly large ranges of the resolution parameters.

4.2.3 Handling Networks with Multiple “Correct” Resolution Parameter Values

Third, the reduction to a deterministic map allows us to more easily handle networks in which there are multiple “correct” values for the resolution parameter. We have seen this occur in practice and have explicitly constructed models (see appendix B) in which multiple partitions of a network are simultaneously stable under the parameter estimation map.

4.2.4 When Communities are Strong Enough, K Does Not Need to be Fixed

Fourth, particularly strong community structure may be stable under the map even when the number of communities is not fixed (as in Pamfil et al.’s proposal [17]). As previously stated, this notion of stability is not as well founded as convergence in Newman’s original scheme, but it is strictly stronger.

By fixing the number of communities being considered and thus decreasing the number of partitions of interest, we only increase the sizes of the domains of optimality. Hence, if a partition is stable when considering any number of communities, it will also be stable when only considering those partitions that share its number of communities. Indeed, stability of a partition σ with respect to a full set of partitions Σ necessarily implies stability of σ with respect to any subset of Σ that includes σ.

CHAPTER 5

Results

In this chapter, we discuss the results of our pruning strategy from chapter 4 on a synthetic multi-layer network and two real-world networks. In all three cases, we will make direct comparisons to the behavior of Pamfil's et al.'s procedure in [17].

5.1 The Karate Club

We first show the results of our procedure on the karate club network of Zachary [23] that we used in section 2.5. Recall that this network describes the social relationships in a university karate club shortly before a disagreement split the club in half.

We obtained 10,000,000 partitions[1] of this network by running the Louvain algorithm (as implemented by Vincent Traag in [24]) across a uniformly spaced grid for $\gamma \in [0.0, 2.0]$ and then used CHAMP [26] to prune this to the subset of partitions that are somewhere dominant.

Among these partitions were 539 unique partitions with more than one community (the single-community partition is optimal for very small values of γ) and the pruned subset from CHAMP has only 9 partitions. Some details of these partitions are shown in Table 5.1 and their domains of optimality and γ estimates are shown in Figure 5.1.

[1] It's important to note that we are running the Louvain algorithm an extremely large number of times only for the purposes of being exhaustive (though 10 million runs on this small of a network takes fewer than 5 minutes on my desktop computer) – the results for this network are qualitatively similar when the number of Louvain runs are as low as 100 to 1000. In fact, the stable partitions of interest in Figure 5.2 generally appear with as few as a dozen input partitions.

In practice, you would run community detection heuristics for only as long as your situation allows (of course, the computational abilities differ greatly between those using a laptop and those with access to large compute clusters or AWS instances). Fortunately, the number of partitions in CHAMP's pruned subset appears to converge very rapidly, so while the quality of our method's results may improve as the number of input partitions increases, it is in no way mandatory to run modularity maximization heuristics for an inordinate amount of time. Indeed, CHAMP will prune a set of partitions, regardless of how small or large the set may be.

Number of communities K	2	3	4	5	6	7	8
Number of unique partitions	10	23	109	184	156	49	7
Number of unique partitions in CHAMP's pruned subset	1	1	2	1	2	1	0
Number of unique partitions in CHAMP's pruned subset when only considering partitions with K communities	2	3	3	4	4	3	2

Table 5.1: A description of the unique partitions returned by running the Louvain heuristic 10 million times on Zachary's karate club network and the pruned subsets from running CHAMP on these partitions.

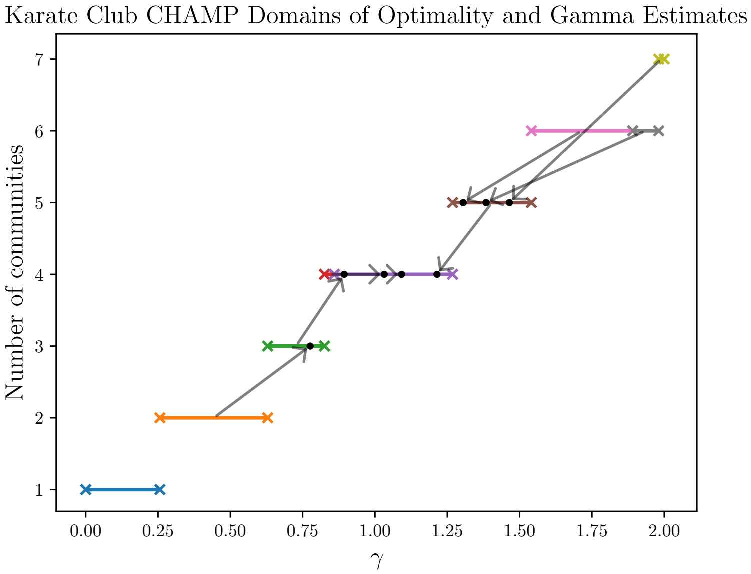


Figure 5.1: The domains of optimality and associated γ estimates for the 9 partitions of the karate club in the pruned subset from CHAMP.

Note that when the number of communities is left unrestricted, there is exactly one stable partition in CHAMP's pruned subset (namely, the 4-community partition in Figure 5.1 whose γ estimate lies within its domain of optimality). This corresponds to the partition that Pamfil et al.'s iterative procedure most frequently converged to in Figure 2.1.

However, when the number of communities K is restricted prior to post-processing with CHAMP, we find exactly one stable partition per choice of $K = 2, 3, \ldots, 8$. The stable 2, 3, and 4 community partitions are shown in Figure 5.2.

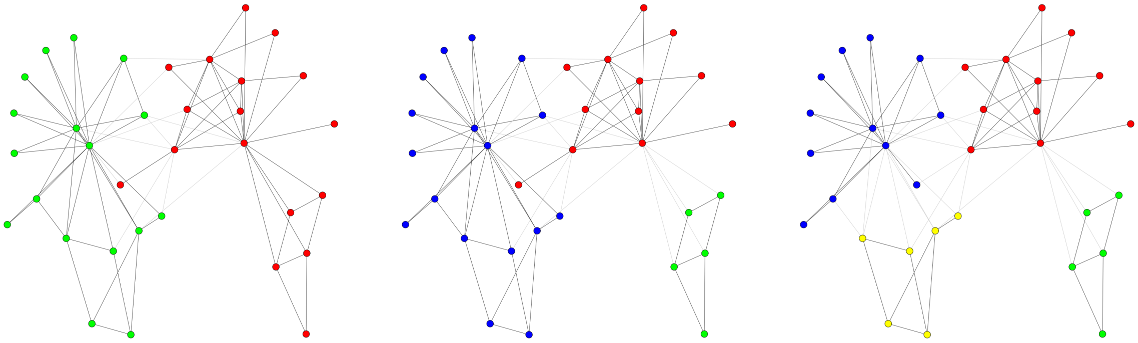


Figure 5.2: Force-directed layouts of the 3 stable partitions of the network from CHAMP's pruned subsets when only considering partitions of 2, 3, and 4 communities, respectively (left to right).

Importantly, our 2-community stable partition closely matches the true splitting of the karate club and our 4-community stable partition is the same as the one obtained from Pamfil et al.'s iterative procedure.[2]

5.2 Synthetic Multi-layer Temporal Network

Recall from section 2.2 that a temporal network represents interactions that occurred at several different instances of time. In such a network, each period of time is represented by a layer and the nodes in each layer are connected to the instances of the same node in the previous and subsequent layers (hence encoding the chronological sequence of the layers' interactions).

In this section, we focus on a synthetic test used in Pamfil et al. [17] to generate temporal networks. The generative model, first used by Ghasemian et al. [29], is as follows.

First, we generate a ground-truth community membership in the first layer by splitting evenly between K available community labels. Then, for each subsequent layer, the community label is

[2] Recall from subsection 4.2.4 that a stable partition σ with respect to a set of partitions Σ will also be stable in any subset of Σ that includes σ. Hence, since this 4-community partition was stable in Figure 5.1, it is also stable when we restrict consideration to partitions with exactly 4 communities.

copied from the previous layer with probability η and randomly assigned from all K possible labels with probability $1 - \eta$.

Using this ground-truth community assignment, edges are independently placed between pairs of nodes in each layer with probability p_{in} if the nodes are in the same ground-truth community and with probability p_{out} otherwise. A model parameter $\varepsilon = p_{\text{out}}/p_{\text{in}}$ is used to control the strength of the community structure in these layers (smaller values of ε place more edges within communities than between communities).

In our tests here, we generated multi-layer networks with "copying probability" $\eta = 0.7$, edge probability ratio $\varepsilon = 0.4$, number of layers $T = 15$, number of communities $K = 2$ and 150 nodes per layer.[3] The behavior of Pamfil et al.'s iterative procedure on this network is shown in Figure 5.3.

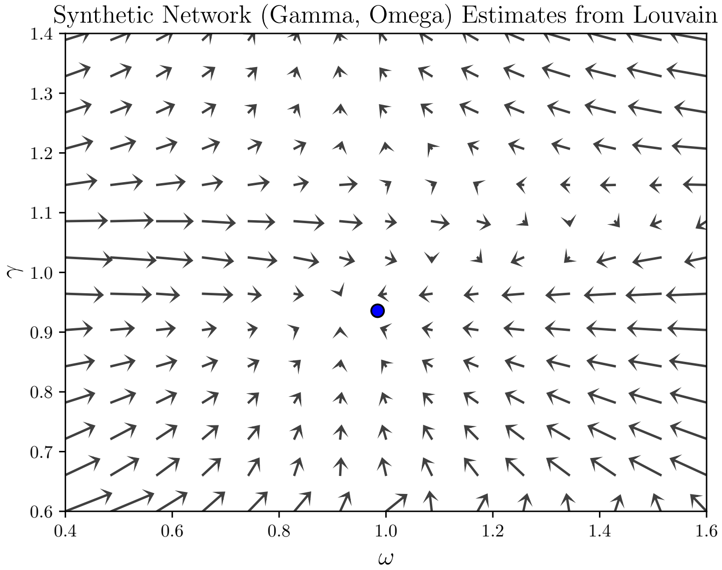


Figure 5.3: The behavior of the iterative procedure introduced in Pamfil et al. [17] on our synthetic network. The parameter values for the ground truth community are shown as a blue point near $(\omega, \gamma) \approx (0.98, 0.94)$. Over a grid of the (ω, γ) plane, arrows indicate the direction of the updated resolution parameter estimates after maximizing modularity with the Louvain algorithm for this choice of (ω, γ), averaged over five trials. As in [17], arrow sizes are scaled down for clarity (here, shown as 10% their actual update movement).

[3] Note that this choice of $K = 2$ means the copying of labels from one layer to the next actually occurs with probability $\eta + \frac{1}{2}(1 - \eta) = 0.85$. This choice is used in both [17] and [29], so we have also chosen to use only two ground truth communities.

This iterative scheme converges close to the ground truth resolution parameter estimates for much of the (ω, γ) plane, but notably there are regions for which the scheme diverges away from the ground truth values (e.g. most initializations with $\gamma > 1.1$ fail to converge to the ground truth).

We now compare these results to our method, which in part will explain why the above scheme can diverge in this way.

After generating a network using the above model, we obtained 50,625 partitions by running the Louvain algorithm in a 225 × 225 uniform grid of $\gamma \in [0.0, 2.0]$, $\omega \in [0.0, 2.0]$. Of these partitions were 33,410 unique partitions with more than one community. We start by pruning with CHAMP *with the number of communities left unconstrained.*

Across the entire resolution parameter area $\gamma \in [0.0, 2.0]$, $\omega \in [0.0, 2.0]$, our pruned subset from CHAMP has 77 partitions with more than one community. The domains of optimality and associated resolution parameter estimates are shown in Figure 5.4.

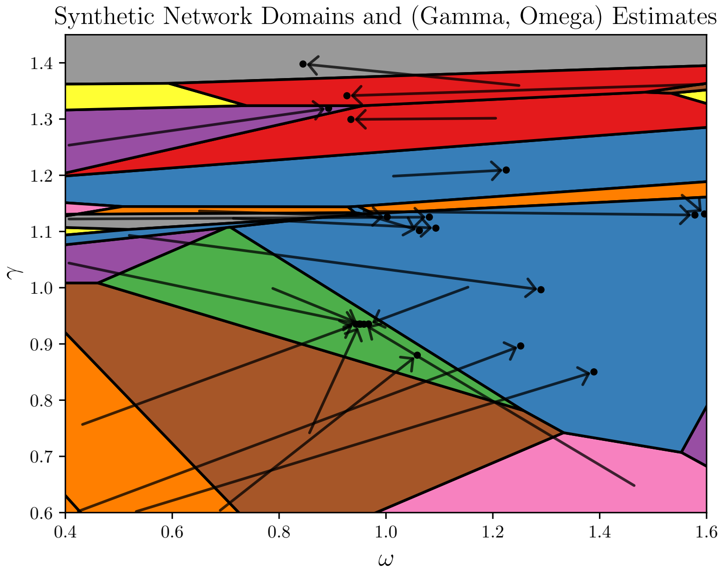


Figure 5.4: Domains of optimality for the partitions in CHAMP's pruned subset (approximately 25 partitions are somewhere dominant in the region of the (ω, γ) plane shown). For each partition, an arrow is drawn from the centroid of the partition's domain of optimality to its resolution parameter estimate (ω, γ).

Qualitatively, these domains exhibit the same behavior as the iterative scheme shown in Figure 5.3 with many partitions' estimates lying close to the ground truth resolution parameter values of

$(\omega, \gamma) \approx (0.98, 0.94)$. Once again, the partitions dominant in the approximate region $\gamma > 1.1$ do not converge near this ground truth. To see why this is the case, we plot the number of communities in these partitions and their AMIs[4] with the ground truth 2-community partition in Figure 5.5.

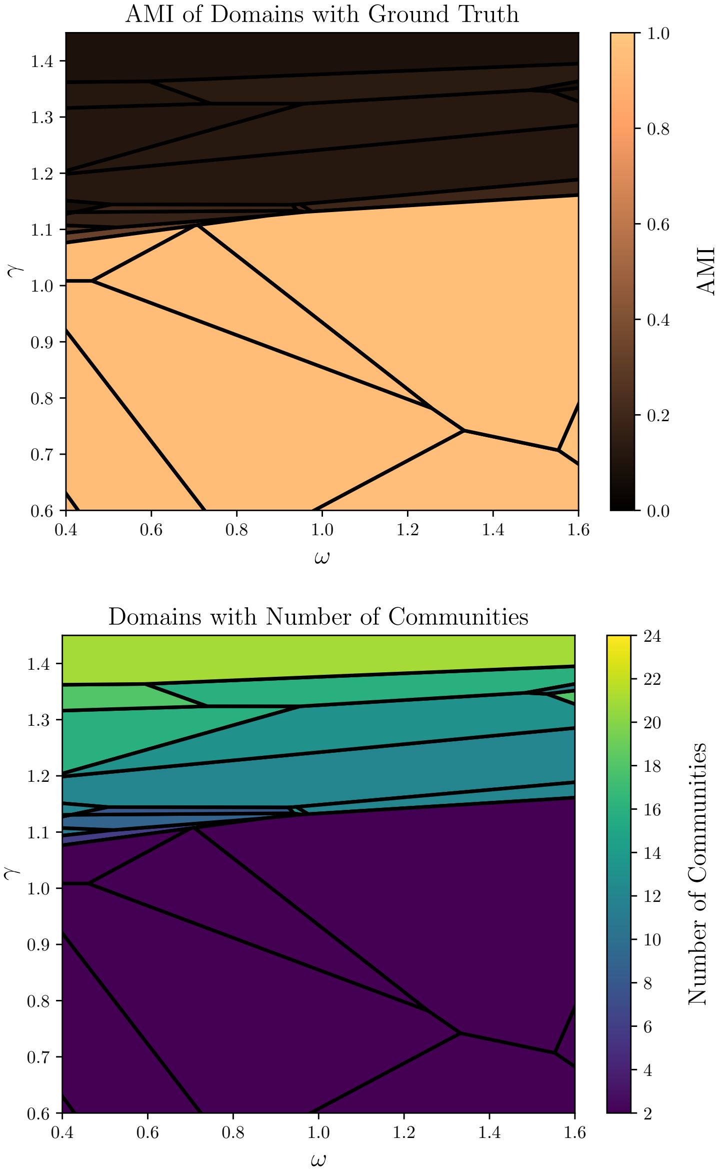


Figure 5.5: Top: Domains of optimality from CHAMP's pruned subset, colored by AMI with the ground truth partition. Bottom: Domains of optimality from CHAMP's pruned subset, colored by number of communities.

[4] In short, the adjusted mutual information (AMI) is a measure of how closely two partitions agree where values closer to 1 indicate stronger alignment and 0 indicates "no alignment". In contrast to normalized mutual information (NMI), AMI is adjusted for chance so that random clustering have an expected AMI of 0.

Here, we see the issue is precisely the one discussed in section 2.5 where the resolution parameter estimates depend heavily on the number of communities in the partition. In fact, the transition around $\gamma \approx 1.1$ is marked by an incredibly rapid increase in the number of communities from $K = 2$ to $K \approx 15$! Unsurprisingly, this transition also coincides with a sharp decrease in alignment with the ground truth partition.

Now, we analyze the stable partitions from CHAMP's pruned subset (once again, with the number of communities unconstrained). 5 out of the 77 partitions with more than one community are stable and the details of these partitions are given in Table 5.2.

Number of communities K	AMI with ground truth	(ω, γ) estimate
2	0.95	(0.94, 0.95)
12	0.12	(1.21, 1.23)
13	0.13	(1.30, 0.93)
16	0.11	(1.32, 0.89)
21	0.09	(1.50, 0.85)
28	0.07	(1.62, 0.84)

Table 5.2: Details of the stable partitions from CHAMP's pruned subset on this synthetic network. Recall that the ground truth resolution parameter estimates here are $(\omega, \gamma) \approx (0.98, 0.94)$.

Notably, the 2-community stable partition has very strong alignment with the ground truth (agreeing for $\sim$99.5% of the network's nodes). In fact, it has the highest AMI value with the ground truth group membership among any of the partitions in CHAMP's pruned subset.

The domains of all other stable partitions lie beyond the transition where the number of communities increases at $\gamma \approx 1.1$ and do not closely match the ground truth. Indeed, in a network with two ground truth communities and only 15 layers, the relatively large number of communities in the other stable partitions ($K = 12, 13, 16, 21, 28$) suggest that they are only dominant for unreasonable choices of the intralayer resolution parameter γ.

Finally, we reproduce Figure 5.4 when CHAMP's pruned subset only considers partitions with $K = 2$, constituting 6,794 partitions from the original runs of the Louvain algorithm. We show the domains of optimality and (ω, γ) estimates in Figure 5.6.

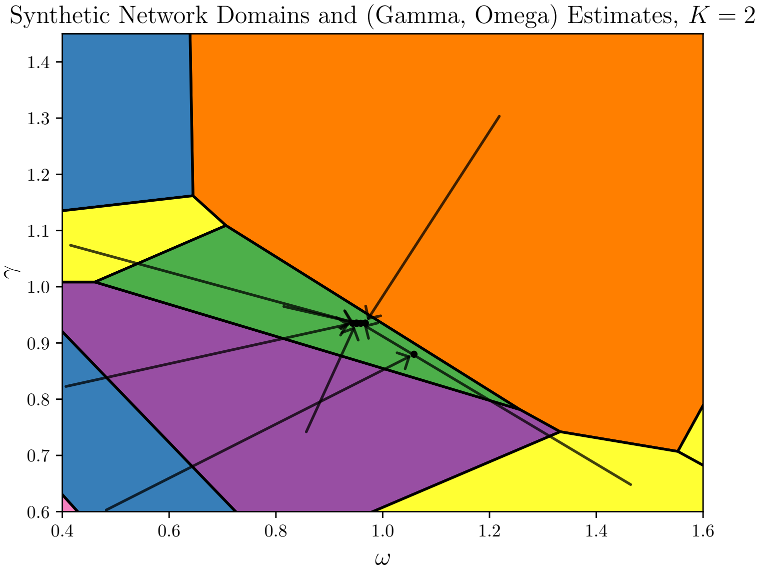


Figure 5.6: Domains of optimality for the partitions in CHAMP's pruned subset when we restrict $K = 2$. For each partition, an arrow is drawn from the centroid of the partition's domain of optimality to its resolution parameter estimate (ω, γ).

In this case, CHAMP's pruned subset has 23 partitions that are somewhere dominant and exactly one of these is stable. This is the same high-AMI, stable 2-community partition that was discovered when K was left unconstrained (since as we've seen before, stability of a partition persists when restricting focus to a smaller subset of partitions).

5.3 Lazega Law Network

Following Pamfil et al. [17], we now demonstrate our approach on the "Lazega Law Firm" network [30]. This is a 3-layer multiplex network that describes the relationships between 71 attorneys. In particular, the individuals were asked to list

1. The members of the firm that they go to for basic professional advice.

2. The members of the firm that they closely work with.

3. The members of the firm that they socialize with outside of work.

These directed associations were then used to form three layers of a network, referred to as the "Advice", "Coworker", and "Friend" layers, respectively. Each node (representing an individual) is then connected to its copy in all other layers to form the complete multiplex network with 213 nodes (71 in each of the 3 layers).

This network is also annotated with various pieces of metadata, which we will use to analyze our results. Here, in order to compare with [17], we will use the following pieces of metadata: status ("partner" or "associate"), gender, office (3 possibilities), seniority (years with the firm, grouped into 5-year bins), age (grouped into 5-year bins), practice ("litigation" or "corporate"), and law school (4 options, one of which is "other").

We ran 1,000,000 instances[5] of the Louvain algorithm on a uniform 2000×500 grid of $\gamma \in [0, 2]$, $\omega \in [0, 3]$, which identified 211,219 unique partitions with more than one community. The pruned subset from CHAMP (where we do not yet restrict the number of communities K) has 152 unique partitions with more than one community. Details of these partitions are given in Table 5.3 and their domains of optimality are shown in and Figure 5.7 and Figure 5.8.

Number of communities K	2	3	4	5	6	7	≥ 8
Number of unique partitions	2.6K	30.4K	34.1K	22.4K	19.5K	18.2K	83.9K
Number of unique partitions in CHAMP's pruned subset	8	19	23	22	15	13	52
Number of unique partitions in CHAMP's pruned subset when only considering partitions with K communities	38	45	52	71	55	63	-
Number of stable partitions in CHAMP's pruned subset when only considering partitions with K communities	3	2	3	3	2	2	-

Table 5.3: A breakdown of the unique partitions returned by the running the Louvain algorithm 1 million times on the Lazega Law Firm network and the resulting pruned subsets from CHAMP.

[5] We note that this large number of iterations is excessive, but we choose to err on the side of being exhaustive here. We will return to the performance of our scheme on this network at the end of this section.

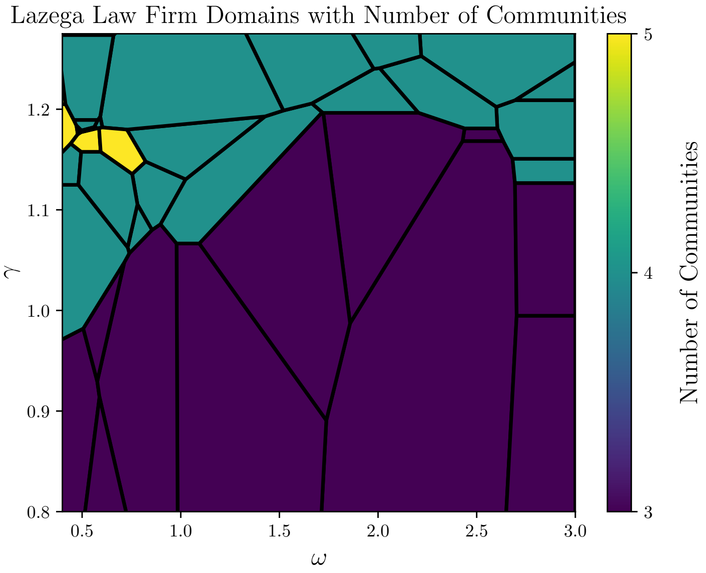


Figure 5.7: Domains of optimality for the partitions in CHAMP's pruned subset, colored by number of communities.

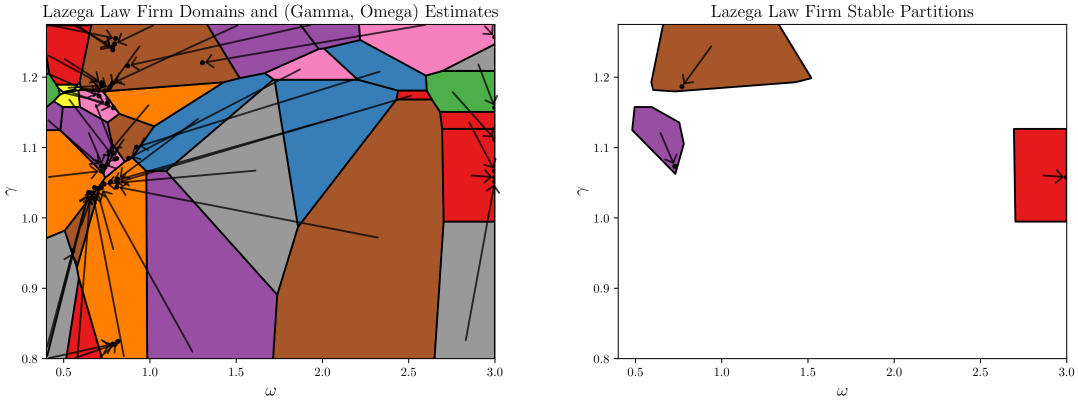
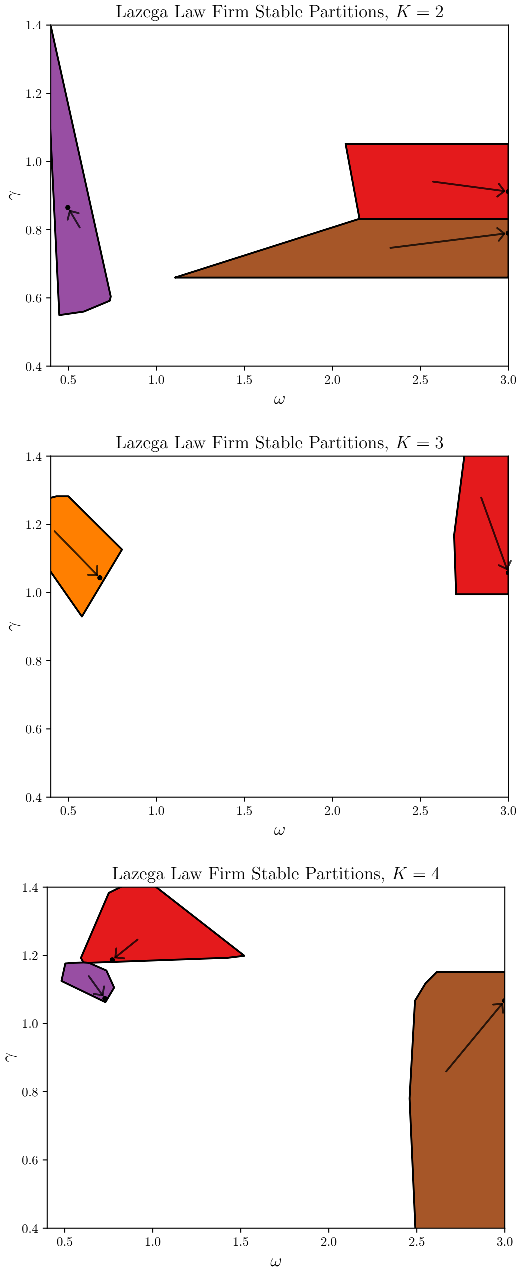


Figure 5.8: Domains of optimality for the partitions in CHAMP's pruned subset. Left: Domains are annotated with arrows that indicate their partition's resolution parameter estimates (ω, γ). Partitions with communities that are the same across all layers have an estimate of $\omega = \infty$, so we have truncated to $\omega = 3$ for plotting purposes. Right: Domains and resolution parameter estimates for the three stable partitions.

The three stable partitions here (one with $K = 3$ and two with $K = 4$) agree with three of the common convergence points (ω, γ) of Pamfil et al.'s [17] iterative procedure on this network.

We additionally consider restricting $K = 2$, 3, and 4 prior to post-processing with CHAMP, which reveals five more stable partitions, four of which correspond to (ω, γ) estimates that are missed by the iterative procedure in [17]. These domains are shown in Figure 5.9 and the associated community memberships are visualized in Figure 5.10.


Lazega Law Firm Stable Partitions, $K = 2$

Lazega Law Firm Stable Partitions, $K = 3$

Lazega Law Firm Stable Partitions, $K = 4$

Figure 5.9: Domains of optimality and resolutions parameter estimates for the stable partitions when we separately fix $K = 2, 3, 4$ (shown top-to-bottom) prior to pruning with CHAMP.

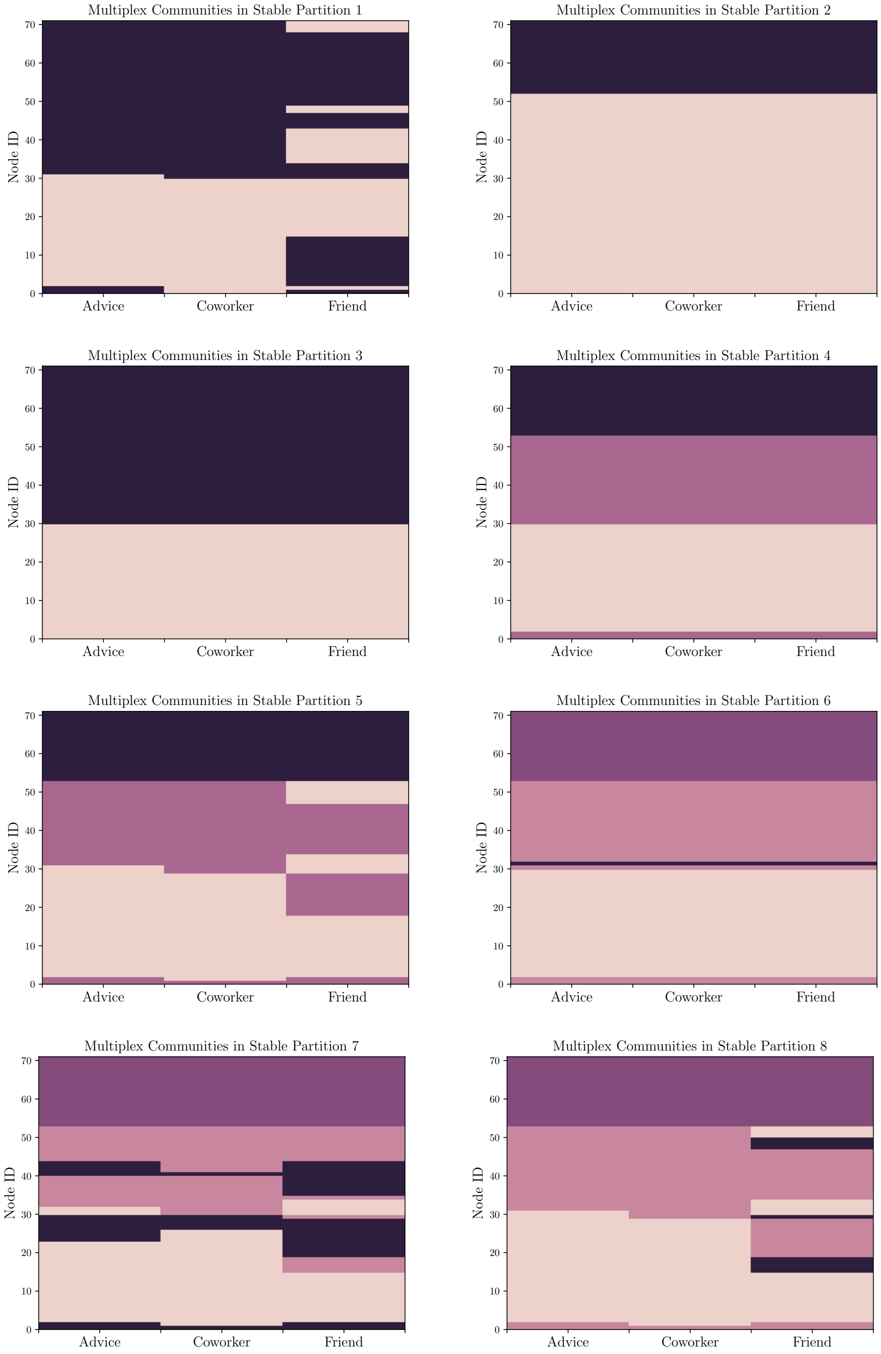


Figure 5.10: Visualizations of the stable partitions in the Lazega Law Firm network with $K = 2, 3, 4$. In each plot, the nodes are colored based on their community label and all plots show the same ordering of nodes. Note that stable partition 5 is very similar to stable partition 8. Also, stable partitions 4 and 6 are virtually identical.

As one might expect, the community labels per individual match closely between the layers that represent their close coworkers and the members of the firm that they go to for advice. However, we often find individuals that are placed in a community in the "Friend" layer that differs from their community in the "Advice" or "Coworker" layers.

We now compute the alignment of our stable partitions with the metadata and compare our results to those of Pamfil et al. [17]. In short, Pamfil et al. ran their iterative procedure 100 times, identified three primary clusters of common convergence points (ω, γ), and used consensus clustering to convert these clusters into three partitions, each with 3 communities. As in [17], we will refer to these as "cluster 1, 2, 3" and note that the partitions for clusters 2 and 3 are very similar (in fact, they differ in only about 5 of the 271 community labels). These results are shown in Table 5.4.

	Approximate (ω, γ)	Office	Practice	Age	Seniority	Status	Gender	Law School
Stable Partition 1, $K = 2$	$(0.5, 0.9)$	0.192	0.163	0.033	0.024	0.002	0.000	0.006
Stable Partition 2, $K = 2$	$(\infty, 0.8)$	0.854	0.000	0.072	0.031	0.037	0.016	0.007
Stable Partition 3, $K = 2$	$(\infty, 0.9)$	0.268	0.507	0.041	0.027	0.025	0.001	0.000
Stable Partition 4, $K = 3$	$(\infty, 1.1)$	0.610	0.469	0.105	0.052	0.040	0.026	0.007
Stable Partition 5, $K = 3$	$(0.7, 1.0)$	0.599	0.193	0.114	0.071	0.079	0.027	0.017
Stable Partition 6, $K = 4$[†]	$(\infty, 1.1)$	0.595	0.455	0.118	0.061	0.049	0.031	0.022
Stable Partition 7, $K = 4$	$(0.8, 1.2)$	0.525	0.212	0.159	0.117	0.150	0.091	0.026
Stable Partition 8, $K = 4$[†]	$(0.7, 1.1)$	0.563	0.196	0.116	0.090	0.077	0.035	0.020
Cluster 1, $K = 3$	$(0.5, 1.0)$	0.575	0.351	0.136	0.148	0.161	0.041	0.026
Cluster 2, $K = 3$	$(1.3, 1.0)$	0.613	0.462	0.097	0.054	0.040	0.026	0.008
Cluster 3, $K = 3$	$(\infty, 1.0)$	0.614	0.401	0.097	0.054	0.043	0.027	0.012
Average over 100 runs	-	0.588	0.431	0.114	0.086	0.080	0.032	0.016

[†] Here, the fourth communities are extremely small (only one node in the case of Stable Partition 6), so one could also group these with $K = 3$.

Table 5.4: Normalized Mutual Information (NMI) scores of our five stable partitions with $K = 2, 3, 4$ and the three consensus clusterings from Pamfil et al. [17].

Our stable partitions 5 and 8 roughly match cluster 1 and our stable partitions 4 and 6 roughly match clusters 2 and 3. However, we recover significant communities that are overlooked by Pamfil et al.'s iterative scheme [17]. Indeed, stable partition 2 is very strongly aligned with the office metadata and stable partition 3 is more strongly aligned with the practice metadata than any of the three clusters from the iterative procedure.

One might assume that we recover more information about the network simply because we have run modularity maximization heuristics an extremely large number of times (one million!). However, when we use only 25 runs of Louvain on a 5×5 uniform grid of the more reasonable range $\gamma \in [0.5, 1.5]$, $\omega \in [0, 3]$, we almost always find the same convergence points (ω, γ) from Table 5.4.

Indeed, Pamfil et al. uses 100 runs of their iterative procedure to analyze this network, each of which requires optimizing modularity multiple times. We have found that we are able to perform $\sim$2000 runs of the Louvain algorithm in the same amount of time, especially since the computations required for finding (ω, γ) estimates on multiplex networks are particularly complicated (see section 2.4).

We in no way suggest that the Pamfil et al.'s strategy *needs* to be run 100 times, nor that the implementation is particularly optimized, but believe this indicates our procedure can be performant in practice.

CHAPTER 6

Maximum γ Estimates

In this chapter, we derive maximum possible γ estimates for ground truth partitions of stochastic block models.

We first derive the result for simple, non-degree-corrected SBMs and then show that the same maximums hold in the general, degree-corrected case. Finally, we demonstrate that real-world networks have community structure consistent with these derivations, yielding bounds on which values of γ "should" be used in modularity when searching for partitions of K communities.

6.1 Equal Block Size, Non-Degree-Corrected, Planted Partition SBMs

Consider a simple stochastic block model that follows some of the "assumptions of modularity" derived from Newman's equivalence [14] (discussed in section A.2). In particular, consider a planted partition stochastic block model in which all blocks have equal size. Let pairs of nodes within communities be connected with probability p_{in} and pairs of nodes between communities be connected with probability p_{out}. Here, we require that $p_{\text{in}} > p_{\text{out}}$ in order to have assortative SBMs that exhibit community structure.[1]

For simplicity, this SBM is not degree-corrected – we are defining the SBM in terms of p_{in} and p_{out} rather than ω_{in} and ω_{out}. We further make the simplifying assumption that nodes cannot be connected to themselves; i.e. self loops cannot exist.

[1] This makes the logic easier, though for this simple SBM, the results here hold for arbitrary p_{in} and p_{out}.

This allows us to generate networks with N nodes, divided into K blocks of size $B = N/K$ nodes each. These K communities are connected according to the preference matrix

$$P_{ij} = \begin{bmatrix} p_{\text{in}} & p_{\text{out}} & \cdots & p_{\text{out}} \\ p_{\text{out}} & p_{\text{in}} & \cdots & p_{\text{out}} \\ \vdots & \vdots & \ddots & \vdots \\ p_{\text{out}} & p_{\text{out}} & \cdots & p_{\text{in}} \end{bmatrix}, \qquad i, j = 1, \ldots, K\,.$$

Then, we have the following expected values

$$\begin{aligned} \text{mean degree } \langle k \rangle &= (B-1)p_{\text{in}} + (K-1)B \cdot p_{\text{out}} \\ \sum_r \kappa_r^2 &= K \cdot \left[B(B-1)p_{\text{in}} + (K-1)B^2 \cdot p_{\text{out}} \right]^2 \\ m_{\text{in}} &= K \cdot \binom{B}{2} \cdot p_{\text{in}} \\ m_{\text{out}} &= \binom{K}{2} \cdot B^2 \cdot p_{\text{out}} \end{aligned}$$

and thus from Equation 1.4 we obtain that, in expectation,

$$\omega_{\text{in}} = \frac{K(B-1)p_{\text{in}}}{(B-1)p_{\text{in}} + (K-1)B \cdot p_{\text{out}}} \qquad \text{and} \qquad \omega_{\text{out}} = \frac{KB \cdot p_{\text{out}}}{(B-1)p_{\text{in}} + (K-1)B \cdot p_{\text{out}}}\,. \tag{6.1}$$

Hence, our expected "ground truth" γ estimate from Equation 1.3 is given by

$$\gamma = \frac{K(B-1)p_{\text{in}} - KB \cdot p_{\text{out}}}{(B-1)p_{\text{in}} + (K-1)B \cdot p_{\text{out}}} \cdot \frac{1}{\ln\left(\frac{B-1}{B} \cdot \frac{p_{\text{in}}}{p_{\text{out}}}\right)}$$

and in the limit of large N, this yields

$$\lim_{N\to\infty} \gamma = \frac{K \cdot p_{\text{in}} - K \cdot p_{\text{out}}}{p_{\text{in}} + (K-1) \cdot p_{\text{out}}} \cdot \frac{1}{\ln\left(\frac{p_{\text{in}}}{p_{\text{out}}}\right)}\,.$$

Note this limit only involves changing the $(B-1)$ factors to B, so the expected γ estimate above is often close to this limiting value as long as B is not too small.

Perhaps surprisingly, this value only depends on the ratio $p_{\text{in}}/p_{\text{out}}$ and is thus directly tied to the strength of the SBM's community structure,[2]

$$\lim_{N\to\infty} \gamma = \frac{K\left(\frac{p_{\text{in}}}{p_{\text{out}}} - 1\right)}{\left(\frac{p_{\text{in}}}{p_{\text{out}}} + (K-1)\right)\ln\left(\frac{p_{\text{in}}}{p_{\text{out}}}\right)}\,. \tag{6.2}$$

Even more surprisingly, this implies that for any given value of K, the possible expected γ estimates are bounded! This is shown in Figure 6.1 where we plot with respect to $p_{\text{out}}/p_{\text{in}}$ (to keep the ratio bounded within 0 and 1), so smaller values of $p_{\text{out}}/p_{\text{in}}$ actually represent stronger community structure.

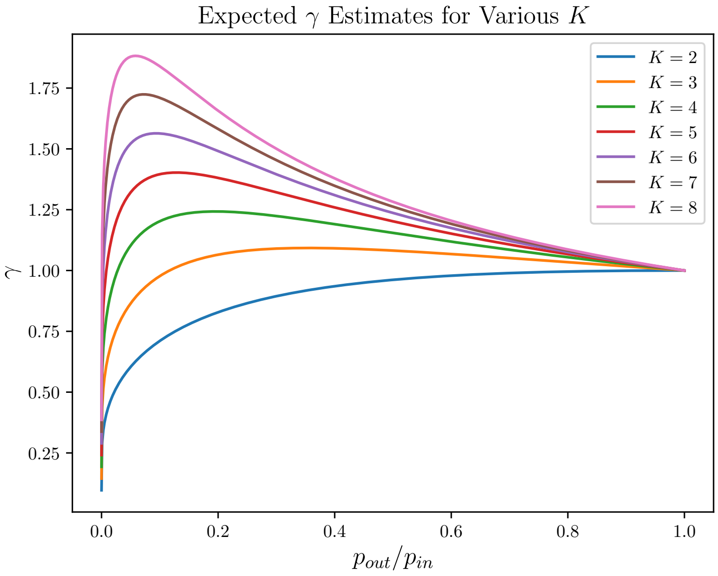


Figure 6.1: The expected γ estimates for various choices of K in our SBM as $p_{\text{out}}/p_{\text{in}}$ varies.

We can numerically maximize[3] this function and have gathered maximum γ estimates for various K in Table 6.1.

[2] We are implicitly assuming that $p_{\text{out}} \neq 0$ here, but this is required to keep the network connected.

[3] For $K > 2$, the maximums seem to be transcendental and thus cannot be written in any particularly "nice" way.

K	2	3	4	5	6	7	8	9	10
γ_{max}	1.0000	1.0926	1.2427	1.4027	1.5640	1.7241	1.8824	2.0388	2.1931

Table 6.1: Maximum "expected ground truth γ estimates" for our simple SBM with $2 \leq K \leq 10$, rounded to four decimal places. We note that all the γ estimates given in Newman's original paper on the duality [14] are less than the associated γ_{max} derived here.

In this way, it is never "correct" to be searching for 5-community partitions beyond $\gamma \approx 1.4$ if we wish to make modularity maximization equivalent to the maximum likelihood fit to our SBM (which is *not* degree-corrected here).

Indeed, since partitions that exhibit strong community structure are often close to optimal near their γ estimates (e.g. stable partitions or fixed points in Pamfil et al.'s [17] iterative scheme), this also suggests that one should not be run modularity maximization heuristics above the γ_{max} values associated with the maximum K of interest.

It is important to realize that these maximum γ estimates are not directly applicable in the degree-corrected case. Indeed, we can see from Equation 6.1 that the expected ω_{in} and ω_{out} estimates are restricted to

$$\omega_{\text{in}} + (K-1)\omega_{\text{out}} = K \tag{6.3}$$

for a K block SBM here. Essentially, our simplified model restricts the degree-corrected SBM parameters to a line segment in Ω space. We visually demonstrate this phenomenon in Figure 6.2.

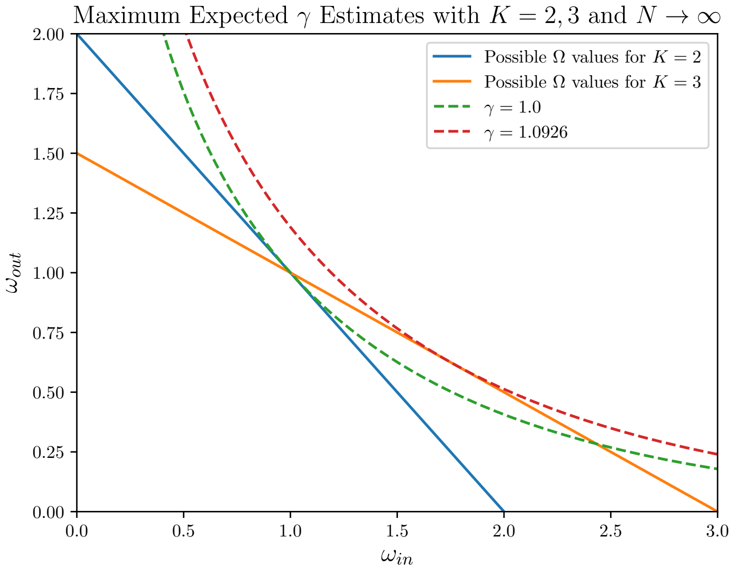


Figure 6.2: Visualization of possible ω_{in} and ω_{out} values and associated maximum γ estimates in our simple SBM with $K = 2, 3$.

6.2 Degree-Corrected, Planted Partition SBMs

We now show that the same maximum γ estimates hold in the more general, degree-corrected case. For a fixed number of communities $K > 2$, we can rewrite our ω_{in} and ω_{out} estimates from Equation 1.4 as

$$\omega_{\text{in}} = \frac{4m \cdot m_{\text{in}}}{\sum_r \kappa_r^2} \qquad \text{and} \qquad \omega_{\text{out}} = \frac{4m^2 - 4m \cdot m_{\text{in}}}{4m^2 - \sum_r \kappa_r^2}\,.$$

Then, all $\kappa_r > 0$ and $\sum_r \kappa_r = 2m$, so the Cauchy-Schwarz inequality yields

$$\left(\sum_{r=1}^{K} \kappa_r \cdot 1\right)^2 \leq \left(\sum_{r=1}^{K} \kappa_r^2\right) \cdot \left(\sum_{r=1}^{K} 1^2\right)$$
$$\frac{4m^2}{K} \leq \sum_{r=1}^{K} \kappa_r^2\,.$$

Moreover, when we have assortative community structure (i.e. $\omega_{\text{in}} > \omega_{\text{out}}$),

$$\begin{aligned}
\omega_{\text{in}} &> \omega_{\text{out}} \\
\frac{4m \cdot m_{\text{in}}}{\sum_r \kappa_r^2} &> \frac{4m^2 - 4m \cdot m_{\text{in}}}{4m^2 - \sum_r \kappa_r^2} \\
(4m \cdot m_{\text{in}}) \left(4m^2 - \sum_r \kappa_r^2 \right) &> \left(4m^2 - 4m \cdot m_{\text{in}} \right) \left(\sum_r \kappa_r^2 \right) \\
16m^3 \cdot m_{\text{in}} - \cancel{4m \cdot m_{\text{in}} \sum_r \kappa_r^2} &> 4m^2 \sum_r \kappa_r^2 - \cancel{4m \cdot m_{\text{in}} \sum_r \kappa_r^2} \\
4m \cdot m_{\text{in}} &> \sum \kappa_r^2 \, ,
\end{aligned}$$

and thus $\frac{4m^2}{K} \leq \sum_r \kappa_r^2 < 4m \cdot m_{\text{in}}$. Hence,

$$\omega_{\text{in}} + (K-1)\omega_{\text{out}} = \frac{4m \cdot m_{\text{in}}}{\sum_r \kappa_r^2} + (K-1)\frac{4m^2 - 4m \cdot m_{\text{in}}}{4m^2 - \sum_r \kappa_r^2} \, ,$$

and for these possible $\sum_r \kappa_r$ values,

$$\frac{\partial}{\partial m_{\text{in}}} \left[\omega_{\text{in}} + (K-1)\omega_{\text{out}} \right] = \frac{4m}{\sum_r \kappa_r^2} - (K-1)\frac{4m}{4m^2 - \sum_r \kappa_r^2} < 0 \iff \sum_r \kappa_r^2 > \frac{4m^2}{K} \, .$$

Finally, this means

$$\begin{aligned}
\omega_{\text{in}} + (K-1)\omega_{\text{out}} &\leq \frac{4m \cdot 0}{\sum_r \kappa_r^2} + (K-1)\frac{4m^2 - 4m \cdot 0}{4m^2 - \sum_r \kappa_r^2} \\
&\leq (K-1)\frac{4m^2}{4m^2 - 4m^2/K} \\
&\leq (K-1)\frac{1}{1 - 1/K} = (K-1)\frac{K}{K-1} = K \, ,
\end{aligned}$$

which matches the bound we obtained in Equation 6.3 for the non-degree-corrected SBM. We visualize this bound in the $(\omega_{\text{in}}, \omega_{\text{out}})$ plane in Figure 6.3.

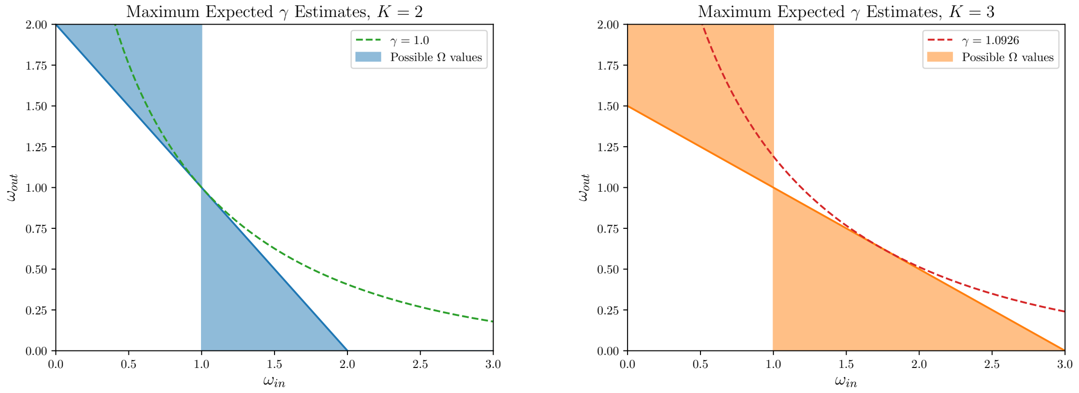


Figure 6.3: Visualization of possible ω_{in} and ω_{out} values in our degree-corrected SBM for $K = 2, 3$. Note that the maximum γ estimates from section 6.1 hold when $\omega_{\text{in}} > \omega_{\text{out}}$ ("assortative SBMs").

6.3 Observed γ Estimates in Real-World Networks

We now compute γ estimates on the giant connected component of a handful of networks from the Stanford Large Network Dataset Collection [31]. Here, we focus on 16 single-layer social networks[4], ranging from 4K to 82K nodes and 17K to 948K edges.

We ran the Louvain algorithm 1000 times on each network on a uniform grid of $\gamma \in [0, 10]$. Then, we computed γ estimates for each partition and grouped these estimates by K. These are shown in Figure 6.4, plotted alongside the average[5] and maximum γ estimates derived from Equation 6.2.

[4] In particular, we use

- The 8 networks from the Gemsec Facebook dataset (gemsec-Facebook)
- The 3 networks from the Gemsec Deezer dataset (gemsec-Deezer)
- The 2 Slashdot social networks from November 2008 and February 2009 (soc-Slashdot0811 and soc-Slashdot0922)
- The (anonymized) social circles from Facebook (ego-Facebook)
- The Who-trusts-whom network of Epinions.com (soc-Epinions1)
- The Wikipedia who-votes-on-whom network (wiki-Vote)

[5] The "average" γ estimate is the average value of $\gamma = (\omega_{\text{in}} - \omega_{\text{out}})/(\ln \omega_{\text{in}} - \ln \omega_{\text{out}})$ on the triangle in the $(\omega_{\text{in}}, \omega_{\text{out}})$ plane with vertices $(1, 0)$, $(1, 1)$, and $(K, 0)$. These triangles define the possible Ω estimates for assortative degree-corrected SBMs (as shown in Figure 6.3).

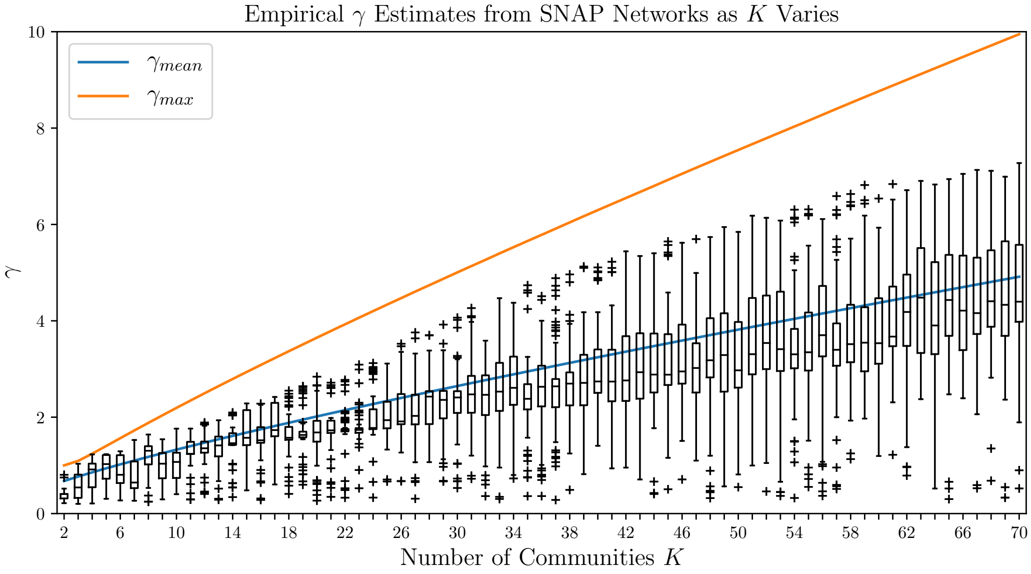


Figure 6.4: Boxplots of observed γ estimates on 16 social networks from SNAP [31], plotted alongside the average and maximum γ estimates from our degree-corrected SBM.

Indeed, all observed γ estimates on these networks lie below our γ_{max} values and our γ_{mean} values appear to match the median estimates.

This provides a priori bounds on the range of resolution parameters that should be used in modularity maximization if a maximum desired number of communities is known or can be estimated. We are hopeful that this can be used to guide modularity-based community detection strategies in a less ad-hoc way.

CHAPTER 7

Conclusion and Future Work

In this thesis, we have presented a pruning strategy that combines the equivalences of Newman [14] and Pamfil et al. [17] with the partition post-processing tool of Weir et al. [26]. This new strategy is statistically principled and reduces the number of partitions that one would need to consider by a remarkable degree. Indeed, in the examples considered here, the subsets of stable partitions (those that are "significant" from the perspective of stochastic block model inference) were often many orders of magnitude smaller than the original set of partitions. Moreover, our method addresses some of the shortcomings of Pamfil et al.'s iterative procedure and we have shown that this allows us to recover significant community structure that their algorithm misses.

This thesis also derived upper bounds on the values of the resolution parameter for which single-layer modularity maximization can be equivalent to optimizing the likelihood fit of a degree-corrected, planted partition stochastic block model. This provides a priori bounds on which values for the resolution parameter "should" be chosen if a desired number of communities is known or can be approximated.

Future work could focus on incorporating this procedure into pipelines that repeatedly maximize modularity rather than using it as a post-processing tool. In fact, Qhull's [28] implementation of halfspace intersection supports incremental adding of halfspaces, so one could create a variant of CHAMP that efficiently updates the pruned subset of partitions and their domains of optimality as each new partition is found. In fact, we have found that we tend to find the stable partitions of interest even when the number of runs of the Louvain algorithm [12] is small, so we believe that such a variant could be very performant in practice. Moreover, the manner in which the domains of optimality and resolution parameter estimates are updated could be used to guide the choice of new resolution parameters during the runs of modularity maximization procedures.

Additional future work could focus on analyzing the equivalences between modularity maximization and SBM inference in more detail, especially when the number of communities is not fixed. We briefly discuss these ideas in appendix A, starting in section A.5 with an overview of the single-layer equivalence when the number of communities is allowed to vary. We also outline some methods to transfer between regions of resolution parameters and SBM parameters in section A.3 and section A.4, but do not fully analyze their utility. These results could be used to “rescale” a partition’s domain of optimality to the region of Ω parameters for which it has a greater likelihood fit to an underlying SBM than any other observed partitions. Hence, such a rescaling might allow for a statistically principled comparison of the areas of domains of optimality across a wide range of resolution parameters.

APPENDIX A

A Closer Look at the Details and Consequences of Newman's Equivalence

Here, we discuss some details of Newman's equivalence [14] that were glossed over in our discussion in section 1.3.

Recall from section 1.2 and section 1.3 that finding the maximum likelihood fit of a planted partition, degree-corrected SBM amounts to maximizing

$$\begin{aligned}\ln P(\mathbf{A} \mid \Omega, \mathbf{g}) = \frac{1}{2} \ln\left(\frac{\omega_{\text{in}}}{\omega_{\text{out}}}\right) \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \cdot \frac{\omega_{\text{in}} - \omega_{\text{out}}}{\ln \omega_{\text{in}} - \ln \omega_{\text{out}}}\right] \delta(g_i, g_j) \\ + m(\ln \omega_{\text{out}} - \omega_{\text{out}})\end{aligned} \tag{A.1}$$

and this becomes equivalent to the optimization of (single-layer) modularity

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{i,j} - \gamma \cdot \frac{k_i k_j}{2m}\right] \delta(c_i, c_j) \tag{A.2}$$

when the resolution parameter takes value

$$\gamma = \frac{\omega_{\text{in}} - \omega_{\text{out}}}{\ln \omega_{\text{in}} - \ln \omega_{\text{out}}}\,. \tag{A.3}$$

A.1 Equivalence to Modularity Maximization Holds Only When $\omega_{\text{in}} > \omega_{\text{out}}$

First, this equivalence is to modularity *maximization* only when $\omega_{\text{in}} > \omega_{\text{out}}$ so that the underlying SBM exhibits assortative community structure (i.e. its communities are more densely connected internally than they are to the rest of the network).

Earlier, we had ignored the leading $\ln(\omega_{\text{in}}/\omega_{\text{out}})$ term in Equation A.1, claiming that it does not affect optimization. However, consider the case in which the SBM has disassortative structure with $\omega_{\text{out}} > \omega_{\text{in}}$ so that nodes are more preferentially connected to nodes in other groups than nodes in their own group. In this case, the leading multiplicative prefactor becomes negative!

Indeed, when $\omega_{\text{out}} > \omega_{\text{in}}$, the maximization of the likelihood is actually equivalent to the *minimization* of modularity. When we refer to "community detection", we are almost always referring to the assortative case, so we had ignored this situation in the introductory chapter.

A.2 "Assumptions of Modularity"

Some variations on the stochastic block model include tunable parameters that allow for controlling the sizes of the ground truth groups [32]. However, as Newman notes, "the version [...] to which modularity maximization is equivalent, includes no such parameters, [...] which in effect means that a priori the sizes of all groups are the same and hence that modularity maximization implicitly prefers groups of uniform size" [14].

Moreover, when the number of communities is fixed and $\gamma > 0$, modularity maximization is always equivalent to the maximum likelihood fit of some planted partition SBM – for arbitrary $\gamma' > 0$ there are an infinite number of choices for ω_{in} and ω_{out} that will make the estimated $\gamma = \gamma'$ in Equation A.3 (see section A.3).

In this way, modularity maximization

- "Assumes" that all communities in a network are "statistically similar" (in the equivalence to a planted partition, all communities share the same intra-group and inter-group connection propensities ω_{in} and ω_{out}).

- "Assumes" that all communities in a network are the same size (since the SBM of interest includes no parameters controlling group size). This means that a change in the number of communities can make a large difference in the "assumed group size" (and thus, estimates for ω_{in} and ω_{out}).

- Makes no assumption on the degree distribution of the network (since the equivalence is to a degree-corrected SBM).

A.3 Mapping Between γ and Ω in the Equivalence

Unsurprisingly since the resolution parameter estimation

$$\gamma = \frac{\omega_{\text{in}} - \omega_{\text{out}}}{\ln \omega_{\text{in}} - \ln \omega_{\text{out}}}$$

maps a two-dimensional space of $(\omega_{\text{in}}, \omega_{\text{out}})$ to the one dimensional space of γ in a smooth way,[1] this map is not one-to-one. Indeed, for any choice of γ', there are an infinite number of $(\omega_{\text{in}}, \omega_{\text{out}})$ pairs that satisfy $\gamma = \gamma'$ in the above equation, as shown in Figure A.1.

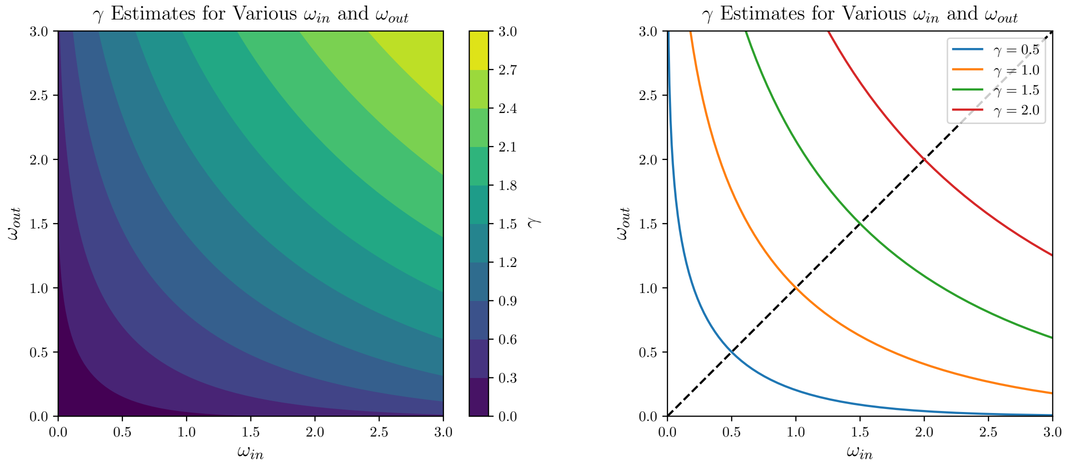


Figure A.1: Left: The value of the γ estimate as ω_{in} and ω_{out} vary. Right: The possible $(\omega_{\text{in}}, \omega_{\text{out}})$ pairs associated with various choices of γ.

This means that the equivalence with between modularity optimization and SBM inference actually holds for many possible pairs of SBM parameters ω_{in} and ω_{out}. Indeed, given any desired

[1] Ignoring the singularities when $\omega_{\text{in}} = \omega_{\text{out}}$, this function is infinitely differentiable when $\omega_{\text{in}}, \omega_{\text{out}} > 0$.

choice of γ and one of the two parameters ω_{in} or ω_{out}, we can compute the other parameter:

$$\begin{aligned}
\omega_{\text{in}}(\gamma, \omega_{\text{out}}) &= \begin{cases} -\gamma \cdot W_{-1}\left(\frac{-\omega_{\text{out}} \cdot e^{-\omega_{\text{out}}/\gamma}}{\gamma}\right), & \text{if } \omega_{\text{out}} < \gamma \\ -\gamma \cdot W_{0}\left(\frac{-\omega_{\text{out}} \cdot e^{-\omega_{\text{out}}/\gamma}}{\gamma}\right), & \text{if } \omega_{\text{out}} > \gamma \end{cases} \\
\omega_{\text{out}}(\gamma, \omega_{\text{in}}) &= \begin{cases} -\gamma \cdot W_{-1}\left(\frac{-\omega_{\text{in}} \cdot e^{-\omega_{\text{in}}/\gamma}}{\gamma}\right), & \text{if } \omega_{\text{in}} < \gamma \\ -\gamma \cdot W_{0}\left(\frac{-\omega_{\text{in}} \cdot e^{-\omega_{\text{in}}/\gamma}}{\gamma}\right), & \text{if } \omega_{\text{in}} > \gamma \end{cases},
\end{aligned} \tag{A.4}$$

where W is the "Lambert W function", the function satisfying $W_0(xe^x) = x$ for $x \geq -1$ and $W_{-1}(xe^x) = x$ for $x \leq -1$.

Note that these ω_{in} and ω_{out} functions are continuous as seen in Figure A.1 – the "principal branch" W_0 meets the "lower branch" W_{-1} when the argument is $-e^{-1}$, in which case $\omega_{\text{in}} = \omega_{\text{out}} = \gamma$ (though of course, the γ estimate is actually undefined here, but this matches the limiting behavior as $\omega_{\text{in}} \to \gamma$ and $\omega_{\text{out}} \to \gamma$).

A.4 Mapping Ranges of γ to Possible Ω Parameters of Planted Partition SBMs

This mapping between γ and possible $(\omega_{\text{in}}, \omega_{\text{out}})$ values also allows one to map entire domains of dominance in the space of γ values to the corresponding region in the $(\omega_{\text{in}}, \omega_{\text{out}})$ plane. We show this in Figure A.2.

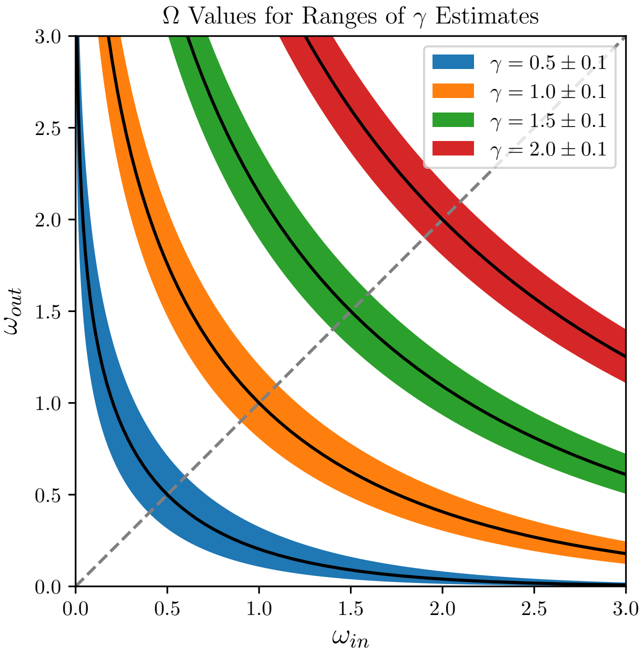


Figure A.2: Regions of possible $\omega_{\text{in}}, \omega_{\text{out}}$ values for various ranges of γ estimates.

Indeed, when a partition has a domain of optimality given by $[\gamma_{\text{center}} - r, \gamma_{\text{center}} + r]$, it appears to be the case that

$$\begin{aligned} \text{area in ``}\Omega\text{ space''} &\approx 13.1594725 \cdot r \cdot \gamma_{\text{center}} \\ &\propto (\text{area in ``}\gamma\text{ space''}) \cdot \gamma_{\text{center}} \, . \end{aligned} \tag{A.5}$$

That is, if a partition has a domain of optimality given by $\gamma_{\text{center}} \pm r$, the area of the corresponding region in the $\omega_{\text{in}}, \omega_{\text{out}}$ "planted partition parameters plane" (i.e. the area of the region for which the partition has a greater likelihood fit to an underlying SBM than other partitions of the same number of communities) scales in a manner proportional to $r \cdot \gamma_{\text{center}}$.

In this way, one could rescale domains of optimality to represent the area in Ω space that the domain represents after passing through the equivalence, though it's unclear if this would actually be useful in practice. One could also correct for the possible Ω values as K varies (see Figure 6.3).

Interestingly, this gives an alternate explanation for why the domains of optimality tend to get smaller and the stochasticity of modularity maximization procedures tends to increase as γ grows larger. As this occurs, the same area of possible $\omega_{\text{in}}, \omega_{\text{out}}$ SBM parameters takes up an ever-smaller range of the γ space.

A.5 The "Equivalence" When K is not Fixed

As we've seen in section 1.3, the equivalence between modularity optimization and SBM inference only holds when the number of blocks K is kept fixed. However, Pamfil et al. [17] used the equivalence (and its extensions) without fixing the number of communities and we have found that when the community structure is strong enough, we can sometimes detect stability under the resolution parameter estimation map even as K varies (see subsection 4.2.4 and chapter 5 in general).

To help explain this, consider comparing the full log-likelihood SBM fit to modularity (for simplicity, ignoring the self-loop terms),

$$\begin{aligned}
\ln P(\mathbf{A} \mid \Omega, \mathbf{g}) &= \frac{1}{2} \ln\left(\frac{\omega_{\text{in}}}{\omega_{\text{out}}}\right) \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \cdot \frac{\omega_{\text{in}} - \omega_{\text{out}}}{\ln \omega_{\text{in}} - \ln \omega_{\text{out}}}\right] \delta(g_i, g_j) \\
&\quad + m(\ln \omega_{\text{out}} - \omega_{\text{out}}) \\
\frac{1}{m} \cdot \ln P(\mathbf{A} \mid \Omega, \mathbf{g}) &= \ln\left(\frac{\omega_{\text{in}}}{\omega_{\text{out}}}\right) \cdot \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \cdot \frac{\omega_{\text{in}} - \omega_{\text{out}}}{\ln \omega_{\text{in}} - \ln \omega_{\text{out}}}\right] \delta(g_i, g_j) \\
&\quad + (\ln \omega_{\text{out}} - \omega_{\text{out}}) \ .
\end{aligned}$$

Then, when we take the resolution parameter to have the value from the equivalence $\gamma = (\omega_{\text{in}} - \omega_{\text{out}})/(\ln \omega_{\text{in}} - \ln \omega_{\text{out}})$, we have that

$$Q = \frac{\ln P(\mathbf{A} \mid \Omega, \mathbf{g}) + m\,(\omega_{\text{out}} - \ln \omega_{\text{out}})}{m \ln\left(\omega_{\text{in}}/\omega_{\text{out}}\right)}\,, \tag{A.6}$$

where we've rewritten $-(\ln \omega_{\text{out}} - \omega_{\text{out}})$ as $(\omega_{\text{out}} - \ln \omega_{\text{out}})$ to emphasize that this quantity is positive (in fact, $(x - \ln x) \geq 1$ for all positive x).

Of course, when K is held constant, we can repeatedly estimate ω_{in} and ω_{out} to determine γ and iterate until convergence. Indeed, recall that when a network is actually drawn from a planted partition of K blocks that this iteration should converge to the correct community structure in the limit of large node degrees.

However, when we allow the number of communities K to vary in modularity maximization, we similarly allow for ω_{in}, ω_{out} to vary in such a way that we can possibly converge to multiple correct ground truth values for ω_{in} and ω_{out}, each corresponding to an SBM with a different number of blocks (and potentially different parameters).[2]

However, if we have partitions with varying numbers of communities in which our estimates for ω_{in} and ω_{out} do not cause the $(\omega_{\text{out}} - \ln \omega_{\text{out}})$ and $\ln(\omega_{\text{in}}/\omega_{\text{out}})$ terms in Equation A.6 to vary much, the equivalence still approximately holds. This can occur, for example, if the community structure for a certain number of blocks is so strong that the log-likelihood dominates the expression.

However, if the estimates for ω_{in} and ω_{out} differ greatly as K varies (e.g. if number of communities changes rapidly as we saw in section 5.2), this equivalence can easily break down as the optimization of modularity is biased towards manipulating the $(\omega_{\text{out}} - \ln \omega_{\text{out}})$ and $\ln(\omega_{\text{in}}/\omega_{\text{out}})$ terms more than the likelihood of the SBM fit.

Unfortunately, the precise relationship between our γ, ω_{in}, and ω_{out} estimates as K increases depends heavily on the topology of the network. Hence, it is difficult to make any truly general statements about the relationship between modularity optimization and SBM inference here.

[2] We explicitly construct examples of such networks in appendix B.

APPENDIX B

Explicit Construction of a "Bistable" SBM

In general, it is possible for a network to have meaningful community structure at many different scales (for example, by considering a hierarchy of communities that are made up of smaller communities). As such, it is natural to ask whether or not a network may have multiple meaningful community structures that are simultaneously stable under our resolution parameter estimation when we allow the number of blocks K to vary.

Here, we explicitly construct such a network model in which two ground truth partitions, one with 2 communities and one with 3 communities, are simultaneously stable. Hence, we refer to such a model as being "bistable".

Consider an equal-block size SBM with 3 communities in which pairs of nodes are connected between communities by the preference matrix (chosen somewhat arbitrarily)

$$P_{ij} = \begin{bmatrix} 10/99 & 1/160 & 1/160 \\ & 5/66 & \delta \\ & & 5/66 \end{bmatrix} \approx \begin{bmatrix} 0.1010 & 0.0063 & 0.0063 \\ & 0.0758 & \delta \\ & & 0.0758 \end{bmatrix}, \tag{B.1}$$

where δ is a tunable parameter. Then, when δ is small, our SBM exhibits strong 3-block structure, but as δ increases towards $5/66$, our SBM exhibits strong 2-block structure.

For some intermediate values of δ, however, we could say that a 3-community ground truth partition coincides with the "2-community ground truth" that merges blocks 2 and 3.

Once again, we make the simplifying assumption that self loops do not exist. Then, following in much the same way as in chapter 6, consider the preference matrix

$$P_{ij} = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ & P_{22} & P_{23} \\ & & P_{33} \end{bmatrix} . \tag{B.2}$$

This gives the expected values

$$\begin{aligned}
m_{\text{in}}^{\text{3 community ground truth}} &= \binom{B}{2} \cdot \sum_i P_{ii} \\
m_{\text{out}}^{\text{3 community ground truth}} &= B^2 \cdot \sum_{i<j} P_{ij} \\
m_{\text{in}}^{\text{2 community ground truth}} &= \binom{B}{2} \cdot \sum_i P_{ii} + B^2 \cdot P_{23} \\
m_{\text{out}}^{\text{2 community ground truth}} &= B^2 \cdot \sum_{i<j} P_{ij} - B^2 \cdot P_{23} \\
\kappa_1^{\text{3 community ground truth}} &= B(B-1)P_{11} + B^2[P_{12} + P_{13}] \\
\kappa_2^{\text{3 community ground truth}} &= B(B-1)P_{22} + B^2[P_{12} + P_{23}] \\
\kappa_3^{\text{3 community ground truth}} &= B(B-1)P_{33} + B^2[P_{13} + P_{23}] \\
\kappa_1^{\text{2 community ground truth}} &= B(B-1)P_{11} + B^2[P_{12} + P_{13}] \\
\kappa_2^{\text{2 community ground truth}} &= B(B-1)[P_{22} + P_{33}] + B^2[P_{12} + P_{13} + 2P_{23}] \, ,
\end{aligned} \tag{B.3}$$

where $\kappa_r = \sum_i k_i \cdot \delta(g_i, r)$ is the sum of degrees in group r. Now, we rewrite modularity as

$$Q(\gamma) = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \gamma \cdot \frac{k_j k_j}{2m} \right) \delta_{g_i g_j} = \frac{m_{\text{in}}}{m} - \gamma \cdot \frac{\sum_r \kappa_r^2}{4m^2} \, , \tag{B.4}$$

which allows us to, in principle, calculate the domains of optimality for the 2-community and 3-community ground truth partitions by calculating the γ value for which their modularities become equal.

We now return to the preference matrix in Equation B.1 where we explicitly fixed all values except for $P_{23} = \delta$. Then, with N nodes and block size $B = N/3$, plugging into Equation B.3 and

Equation B.4 yields

$$\begin{aligned}
m_{\text{in}}^{\text{3 community ground truth}} &= \binom{B}{2} \cdot \frac{25}{99} \\
m_{\text{out}}^{\text{3 community ground truth}} &= B^2 \cdot \left(\frac{2}{160} + \delta \right) \\
m_{\text{in}}^{\text{2 community ground truth}} &= \binom{B}{2} \cdot \frac{25}{99} + B^2 \cdot \delta \\
m_{\text{out}}^{\text{2 community ground truth}} &= B^2 \cdot \frac{2}{160} \\
\kappa_1^{\text{3 community ground truth}} = \kappa_1^{\text{2 community ground truth}} &= B(B-1) \cdot \frac{10}{99} + B^2 \cdot \frac{2}{160} \\
\kappa_2^{\text{3 community ground truth}} = \kappa_3^{\text{3 community ground truth}} &= B(B-1) \cdot \frac{5}{66} + B^2 \cdot \left(\frac{1}{160} + \delta \right) \\
\kappa_2^{\text{2 community ground truth}} &= B(B-1) \cdot \frac{10}{66} + B^2 \cdot \left(\frac{2}{160} + 2\delta \right) ,
\end{aligned} \tag{B.5}$$

and

$$\begin{aligned}
Q^{\text{3 community ground truth}}(\gamma) &= \frac{1000(B-1)}{B(7920\delta + 1099) - 1000} - \gamma \cdot \frac{2\left(B^2\left(\delta + \frac{433}{5280}\right) - \frac{5B}{66}\right)^2 + \left(\frac{B^2}{80} + \frac{10}{99}(B-1)B\right)^2}{4\left(B^2\left(\delta + \frac{1}{80}\right) + \frac{25}{198}(B-1)B\right)^2} \\
Q^{\text{2 community ground truth}}(\gamma) &= \frac{99B}{-7920B\delta - 1099B + 1000} + 1 - \gamma \cdot \frac{\left(B^2\left(2\delta + \frac{1}{80}\right) + \frac{5}{33}(B-1)B\right)^2 + \left(\frac{B^2}{80} + \frac{10}{99}(B-1)B\right)^2}{4\left(B^2\left(\delta + \frac{1099}{7920}\right) - \frac{25B}{198}\right)^2} .
\end{aligned} \tag{B.6}$$

By equating these two modularity functions, we find that the γ value for which the 3-community ground truth partition becomes dominant over the 2-community partition is given by

$$\begin{aligned}
\gamma_{\text{cross}} &= \frac{7040B\delta(B(7920\delta + 1099) - 1000)}{(B(5280\delta + 433) - 400)^2} \\
\lim_{N\to\infty} \gamma_{\text{cross}} &= \frac{7040\delta(1099 + 7920\delta)}{(433 + 5280\delta)^2} .
\end{aligned} \tag{B.7}$$

Finally, we can compute expected ground truth estimates ω_{in}, ω_{out} and γ,

$$\begin{aligned}
\omega_{\text{in}}^{\text{3 community ground truth}} &= \frac{80(80\delta + 11)}{320\delta(80\delta + 13) + 331} \\
\omega_{\text{out}}^{\text{3 community ground truth}} &= \frac{8(80\delta + 1)(80\delta + 11)}{(160\delta + 13)(160\delta + 49)} \\
\omega_{\text{in}}^{\text{2 community ground truth}} &= \frac{256\delta + 19}{32\delta(80\delta + 13) + 25} + 1 \\
\omega_{\text{out}}^{\text{2 community ground truth}} &= \frac{80(80\delta + 11)}{320\delta(80\delta + 13) + 331} \\
\gamma^{\text{3 community ground truth}} &= \frac{8(80\delta + 11)^2(320\delta(80\delta - 7) - 549)}{(160\delta + 13)(160\delta + 49)(320\delta(80\delta + 13) + 331)\left(\log\left(\frac{8(80\delta+1)(80\delta+11)}{(160\delta+13)(160\delta+49)}\right) - \log\left(\frac{80(80\delta+11)}{320\delta(80\delta+13)+331}\right)\right)} \\
\gamma^{\text{2 community ground truth}} &= \frac{2(80\delta + 11)^2(256\delta + 19)}{9(160\delta + 13)\left(2560\delta^2 + 416\delta + 25\right)\left(\log\left(\frac{4(8\delta+1)(80\delta+11)}{2560\delta^2+416\delta+25}\right) - \log\left(\frac{160\delta+22}{1440\delta+117}\right)\right)} .
\end{aligned} \tag{B.8}$$

Note that all the quantities in Equation B.8 do not depend on the number of nodes N or block size B since they're ultimately descriptions of the underlying SBM whose expected values should not depend on the network size.

These results are plotted in Figure B.1, demonstrating that there is a large range of choices for δ that result in the stability of both the 2-community and 3-community ground truths simultaneously.

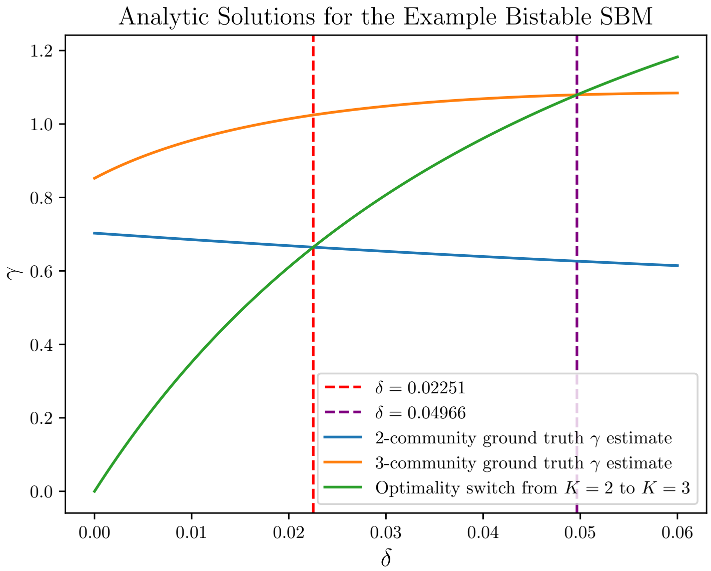


Figure B.1: Ground truth γ estimates for the 2-community and 3-community ground truth partitions in our bistable SBM as $N \to \infty$. The range of choices for δ which give expected bistability are shown with dashed vertical lines.

We also drew networks from this SBM in simulation to demonstrate that our analytic results appear in practice. This is shown in Figure B.2.

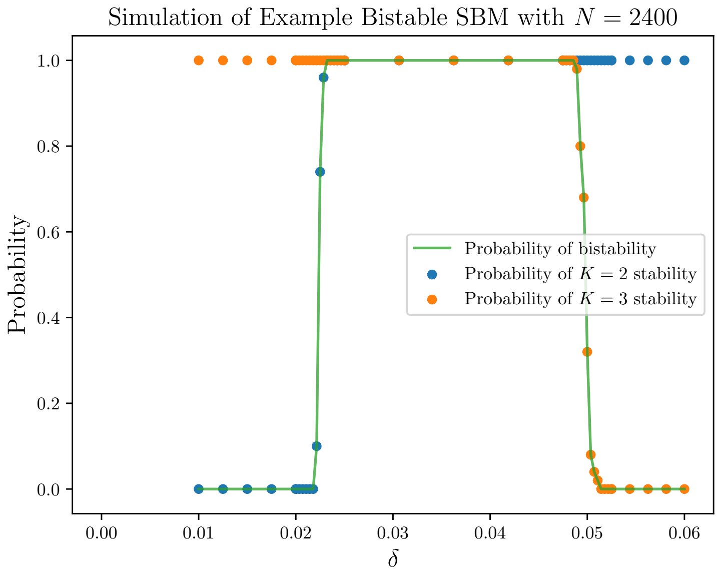


Figure B.2: Probability of stability for the 2-community and 3-community partitions returned by the Louvain algorithm on realizations of our example bistable SBM with $N = 2400$ (taken over 100 trials for each choice of δ).

We note that in this case, the expected mean degree per node increases with N, but this is not the cause of the bistability. Indeed, a similar result can be found when we scale the probabilities P_{ij} down as N increases so that the expected mean degree is fixed, provided that the P_{ij} are chosen so that the ground truth partitions are detectable.

Finally, we plot layouts of some realizations from this example bistable SBM in Figure B.3. Here, the existence (or lack thereof) of "meaningful" 2-community and 3-community partitions can be visually verified as δ varies.

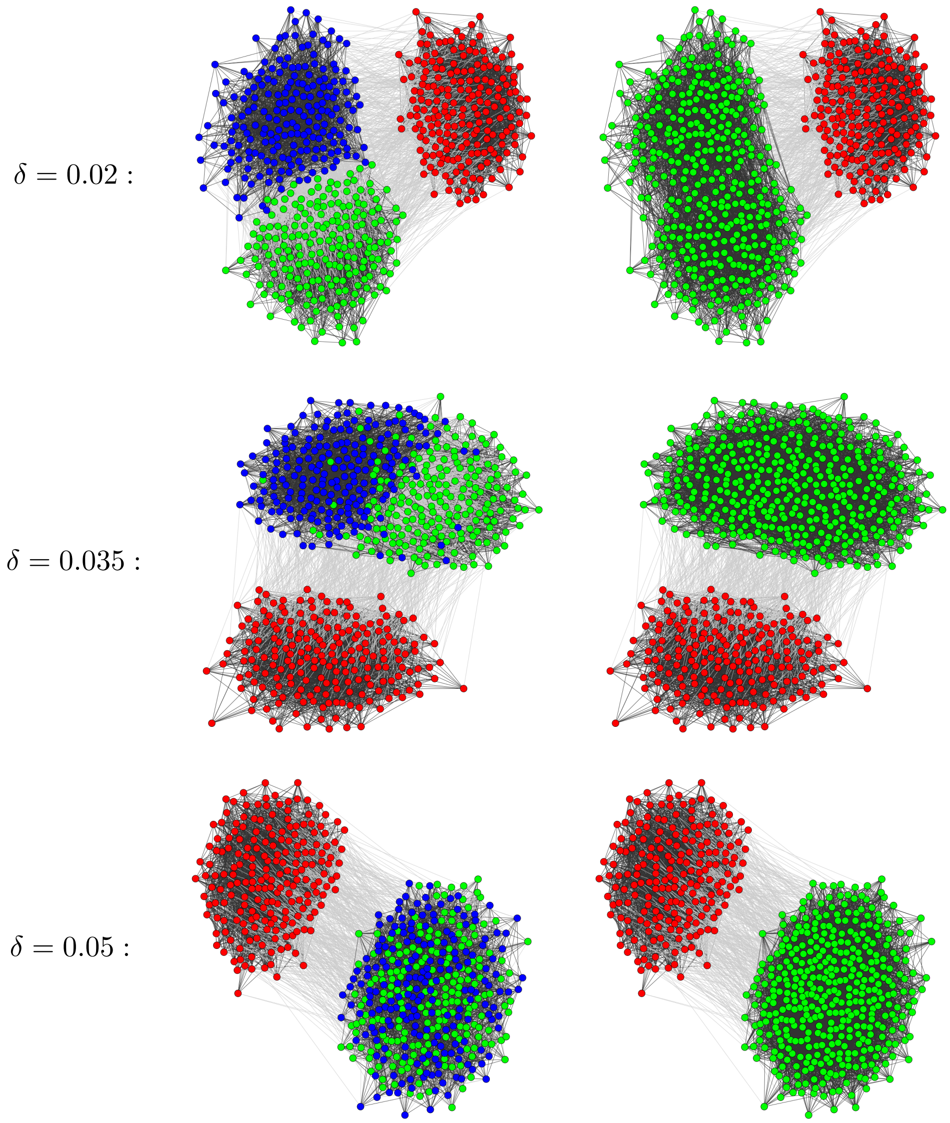


Figure B.3: Force directed layouts from igraph [25] of realizations from our bistable SBM with $N = 600$. For each δ, the 2-community ground truth is shown on the left and the 3-community ground truth is shown on the right. Top: Realization with $\delta = 0.02$, just before the start of the bistable region for δ. Center: Realization with $\delta = 0.035$, in the middle of the bistable region. Bottom: Realization with $\delta = 0.05$, just after the end of the bistable region.

We suspect that one could derive a completely general expression here for an arbitrary number of blocks K and preference matrix P, but considering the relative complexity of Equation B.6, Equation B.7, and Equation B.8, such a result might not be particularly illuminating. Moreover, we were simply concerned with demonstrating that such bistable networks do exist, and thus this explicit example is sufficient for our purposes.

BIBLIOGRAPHY

[1] Mason A. Porter, Jukka-Pekka Onnela, and Peter J. Mucha. Communities in Networks. *arXiv:0902.3788 [cond-mat, physics:nlin, physics:physics, stat]*, February 2009. arXiv: 0902.3788.

[2] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, February 2010.

[3] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, November 2016. arXiv: 1608.00163.

[4] Saray Shai, Natalie Stanley, Clara Granell, Dane Taylor, and Peter J. Mucha. Case studies in network community detection. *arXiv:1705.02305 [physics]*, May 2017. arXiv: 1705.02305.

[5] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. August 2003.

[6] Joerg Reichardt and Stefan Bornholdt. Statistical Mechanics of Community Detection. March 2006.

[7] Alex Arenas, Alberto Fernandez, and Sergio Gomez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039, May 2008. arXiv: physics/0703218.

[8] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, January 2007.

[9] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Görke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On Finding Graph Clusterings with Maximum Modularity. In Andreas Brandstädt, Dieter Kratsch, and Haiko Müller, editors, *Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science, pages 121–132. Springer Berlin Heidelberg, 2007.

[10] T. N. Dinh, X. Li, and M. T. Thai. Network Clustering via Maximizing Modularity: Approximation Algorithms and Theoretical Limits. In *2015 IEEE International Conference on Data Mining*, pages 101–110, November 2015.

[11] U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski, and D. Wagner. Maximizing Modularity is hard. August 2006.

[12] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, October 2008.

[13] Vincent Traag, Ludo Waltman, and Nees Jan van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *arXiv:1810.08473 [physics]*, October 2018. arXiv: 1810.08473.

[14] M. E. J. Newman. Equivalence between modularity optimization and maximum likelihood methods for community detection. *Physical Review E*, 94(5):052315, November 2016.

[15] Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1), January 2011. arXiv: 1008.3926.

[16] Tiago P. Peixoto. Bayesian stochastic blockmodeling. *arXiv:1705.10225 [cond-mat, physics:physics, stat]*, May 2017. arXiv: 1705.10225.

[17] A. Roxana Pamfil, Sam D. Howison, Renaud Lambiotte, and Mason A. Porter. Relating modularity maximization and stochastic block models in multilayer networks. April 2018.

[18] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community Structure in Time-Dependent, Multiscale, and Multiplex Networks. *Science*, 328(5980):876–878, May 2010.

[19] M. Bazzi, M. Porter, S. Williams, M. McDonald, D. Fenn, and S. Howison. Community Detection in Temporal Multilayer Networks, with an Application to Correlation Networks. *Multiscale Modeling & Simulation*, 14(1):1–41, January 2016.

[20] Mikko Kivelä, Alexandre Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer Networks. *Journal of Complex Networks*, 2(3):203–271, September 2014. arXiv: 1309.7233.

[21] Alessandro Lomi, Garry Robins, and Mark Tranmer. Introduction to multilevel social networks. *Social Networks*, 44:266–268, January 2016.

[22] Lucas G. S. Jeub, Marya Bazzi, Inderjit S. Jutla, and Peter J. Mucha. A generalized Louvain method for community detection implemented in MATLAB, 2011.

[23] Wayne W. Zachary. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33(4):452–473, December 1977.

[24] Vincent Traag. Implementation of the Louvain algorithm for community detection with various methods for use with igraph in python.: vtraag/louvain-igraph, March 2019. original-date: 2015-02-25T09:04:01Z.

[25] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[26] William H. Weir, Scott Emmons, Ryan Gibson, Dane Taylor, and Peter J. Mucha. Post-Processing Partitions to Identify Domains of Modularity Optimization. *Algorithms*, 10(3):93, September 2017.

[27] William H. Weir, Ryan Gibson, and Peter J Mucha. CHAMP package: Convex Hull of Admissible Modularity Partitions in Python and MATLAB, 2017.

[28] C. Bradford Barber, David P. Dobkin, David P. Dobkin, and Hannu Huhdanpaa. The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, December 1996.

[29] Amir Ghasemian, Pan Zhang, Aaron Clauset, Cristopher Moore, and Leto Peel. Detectability Thresholds and Optimal Algorithms for Community Structure in Dynamic Networks. *Physical Review X*, 6(3):031005, July 2016.

[30] Emmanuel Lazega. *The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership*. Oxford University Press, 2001.

[31] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. June 2014.

[32] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Inference and Phase Transitions in the Detection of Modules in Sparse Networks. *Physical Review Letters*, 107(6):065701, August 2011.