Correlation in Complex Networks

by

George Tsering Cantwell

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Physics)
in the University of Michigan
2020

Doctoral Committee:

Professor Mark Newman, Chair
Professor Charles Doering
Assistant Professor Jordan Horowitz
Assistant Professor Abigail Jacobs
Associate Professor Xiaoming Mao

George Tsering Cantwell

gcant@umich.edu

ORCID iD: 0000-0002-4205-3691

ACKNOWLEDGMENTS

First, I must thank Mark Newman for his support and mentorship throughout my time at the University of Michigan. Further thanks are due to all of the people who have worked with me on projects related to this thesis. In alphabetical order they are Elizabeth Bruch, Alec Kirkley, Yanchen Liu, Benjamin Maier, Gesine Reinert, Maria Riolo, Alice Schwarze, Carlos Serván, Jordan Snyder, Guillaume St-Onge, and Jean-Gabriel Young.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

ABSTRACT

Network representations are now ubiquitous across science. Indeed, they are the natural representation for complex systems—systems composed of large numbers of interacting components. Occasionally systems can be well represented by simple, regular networks, such as lattices. Usually, however, the networks themselves are complex—highly structured but with no obvious repeating pattern. In this thesis I examine the effects of correlation and interdependence on network phenomena, from three different perspectives.

First, I consider patterns of mixing within networks. Nodes within a network frequently have more connections to others that are similar to themselves than to those that are dissimilar. However, nodes can (and do) display significant heterogeneity in mixing behavior—not all nodes behave identically. This heterogeneity manifests as correlations between individuals' connections. I show how to identify and characterize such patterns, and how this correlation can be used for practical tasks such as imputation.

Second, I look at the effects of correlation on the structure of networks. If edges within a relational data set are correlated with each other, and if we construct a network from this data, then several of the properties commonly associated with real-world complex networks naturally emerge, namely heavy-tailed degree distributions, large numbers of triangles, short path lengths, and large connected components.

Third, I develop a family of technical tools for calculations about networks. If

you are using a network representation, there's a good chance you wish to calculate something about the network—for example, what will happen when a disease spreads across it. An important family of techniques for network calculations assume that the networks are free of short loops, which means that the neighbors of any given node are conditionally independent. However, real-world networks are clustered and clumpy, and this structure undermines the assumption of conditional independence. I consider a prescription to deal with this issue, opening up the possibility for many more analyses of realistic and real-world data.

CHAPTER 1

Introduction

The study of networks is now a well established field, complete with dedicated textbooks, journals, international conferences, popular writings, and research centers [1–14]. Networks are used across scientific disciplines including in biology, neuroscience, public health and medicine, ecology, sociology, political science, engineering and economics [15–33]. In this thesis I will present contributions I have made over the last four years to the theoretical study of networks.

First, we should discuss what a network actually is. Central to the concept of a network is the mathematical notion of a *graph* [34]. A graph is a mathematical object constructed from a set of *nodes* (or *vertices*), and a set of *edges* between nodes. One writes $G = (V, E)$ to represent that graph G contains the nodes in set V, and the edges in E. An example of a graph is shown in Fig. 1.1.

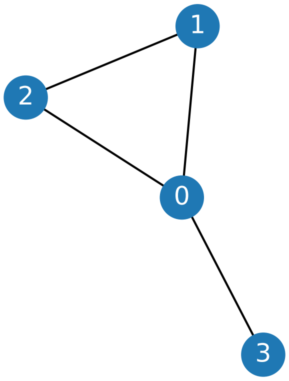


Figure 1.1: An example graph. The node set is $V = \{0, 1, 2, 3\}$ and the edge set is $E = \{(0, 1), (0, 2), (0, 3), (1, 2)\}$.

A graph has an associated *adjacency matrix* [1–4],

$$A_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are connected;} \\ 0 & \text{otherwise.} \end{cases} \tag{1.1}$$

For the graph of Fig. 1.1 this would be

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \tag{1.2}$$

There are countless ways in which we can decorate graphs with extra structure [1–3]. Particularly important among these: We could allow there to be multiple distinct types of node, or distinct types of edge; Edges could be assigned weights so that not all edges are equal; We could allow (or forbid) more than one edge between any pair of nodes; Or, edges could be imbued with a direction, so that if a connects to b it does not necessarily follow that b connects to a.

The word "network" is often used as a synonym for "graph", but usually carries a different connotation. Instead, "network" usually connotes some system, object, or thing, that is both naturally and usefully represented as a graph. The internet is a network—a network of computers. It can be quite naturally represented as a graph whose nodes are computers and edges are connections between them. Brains are networks. They can be represented as graphs in which nodes are neurons and edges are synaptic connections. Likewise for man-made infrastructure, proteins, ecosystems, societies, and so forth [1–3, 15–33].

Empirical network science involves the study of specific real-world networks. Theoretical network science involves the study of models and synthetic data, designed to emulate some aspect of real-world networks or network phenomena. In the chapters that follow, we will consider the effect of correlation and interdependence within networks. We will develop theoretical insight and practical tools for understanding the rich structure of interdependence within networks.

1.1 Why study networks?

The fact that a lot of effort has been put in to understanding networks isn't, in and of itself, a particularly compelling motivation as to why *we* should study them.

Likewise, the observation that many interesting things—brains, food webs, social interactions, etc.—can be conceptualized as networks isn't, in and of itself, an argument that we should study networks qua networks. No doubt networks (or graphs) are useful data structures, but are they more than this? Should scientists be devoting time and effort to study networks in their own right?

One argument that we should appeals to perceived similarities between the networks of disparate systems. Isn't it remarkably curious if the structure of the brain and the structure of social interactions are, in some sense, similar to one another? Real-world network structure is certainly not random, but why would it be similar across domains? Such questions were instrumental to the ratcheting up of network science in the late 1990s [2].

But perhaps more important than the universal properties of networks are the particular differences. For many processes the specific details of the network structure actually matter. To make this claim less abstract let's consider a concrete example. Particularly salient at the time of writing (spring/summer 2020) is the spread of infectious disease, so let's consider this.

1.1.1 Example: Modeling the spread of disease

The SIR model provides a simple but instructive framework for understanding the spread of disease [35, 36]. At any given time (according to the model) an individual is either: Susceptible; Infectious; or Recovered.[1] The model assumes that reinfection is not possible and thus an individual either remains in state S indefinitely, or progresses through $S \rightarrow I \rightarrow R$.

The standard analysis assumes we are studying a large population and tracks the proportion of people in each state [35,36]. People who are currently susceptible become infected through contact with infectious individuals, and we assume the population is "well-mixed", i.e. that everyone has a fixed probability of coming into contact with anyone else. Thus, the rate at which susceptible people are infected is proportional to the current number of infectious people—the more infectious people there are, the more likely you are to come into contact with one. We also assume that infectious people recover (or are removed) at some constant rate.

[1] R may also be used to refer to the rather more ominous "removed" and includes the dead.

Formally,

$$\frac{dS}{dt} = -\beta IS \tag{1.3}$$
$$\frac{dI}{dt} = \beta IS - \gamma I \tag{1.4}$$
$$\frac{dR}{dt} = \gamma I, \tag{1.5}$$

where S, I, and R are the proportion of people who are currently susceptible, infectious, and recovered.[2] These differential equations describe the evolution of the epidemic.

The parameters β and γ control the rate at which infections are passed on and the rate at which people recover, respectively. Their absolute values set the timescale on which the epidemic progresses. From a mathematical perspective this is not hugely consequential. We can decide to measure time in seconds or in years; the math doesn't much care. More important is their ratio. The ratio

$$R_0 = \frac{\beta}{\gamma} \tag{1.6}$$

is hugely consequential, and makes the difference between a local outbreak and a global pandemic [35, 36]. This number, R_0, corresponds to the average number of secondary infections due to an initial infection. If $R_0 < 1$ then we expect only isolated outbreaks. But, if $R_0 > 1$ then we expect the disease to spread across a sizable fraction of the population.

To see this mathematically, consider what happens at the very start of the outbreak. Initially almost everyone is susceptible, $S \approx 1$, and the rate of change of the number of infectious individuals is

$$\frac{dI}{dt} = \left(\beta - \gamma\right) I. \tag{1.7}$$

The disease spreads if this rate of change is positive, i.e if $\beta > \gamma$, or equivalently if $R_0 > 1$. The point at which $R_0 = 1$ represents the epidemic threshold—if R_0 is any larger then we expect to see an epidemic, but if it is smaller, we don't.

This is all well and good but of course provides a hugely simplified picture. The biological details of the disease have been completely swept under the rug. For example, we have assumed that the recovery process is well described by a

[2] Since these are proportions, we necessarily have $S + I + R = 1$.

single recovery rate, γ, and we have not accounted for differences in the population (maybe in the real-world, recovery depends strongly on age). Just as important an oversight, however, are the completely unrealistic sociological assumptions.

The canonical SIR model assumes a well-mixed population, i.e. that anyone in the world is equally likely to come into physical contact with anyone else. This is obviously wrong—and it matters. To improve upon the well-mixed assumption, let's assume that there is some network of physical contact between people [37]. Since the disease requires physical contact to spread it can only spread along edges of this physical contact network. The model is otherwise much the same. A susceptible person can catch the disease from any of its infectious contacts, and this happens at some fixed rate b. Infectious people recover at a fixed rate, g.

The equations that describe the evolution of the epidemic are now more complicated but it's straightforward to run simulations [37,38]. For the sake of argument, let's consider a population of 10 000 people and assume that, on average, people come into close physical contact with 16 other people. In Fig. 1.2 we show the results of simulating the disease on three randomly generated networks that match these assumptions.

First, we consider an entirely random network, in which everyone has exactly 16 randomly chosen contacts [39]. From a sociological perspective this is only a small improvement on the well-mixed assumption. Rather than catching the disease from anyone, you can now only catch the disease from a small number of people with whom you are in regular physical contact. However, the structure of this random network is completely unrealistic.

One feature of this first network that is particularly unrealistic is that everyone has exactly 16 close physical contacts. In reality there will be a significant degree of heterogeneity in the number of contacts people have [1,2]. Some people, for example people who work from home, may come into close contact with substantially fewer people. Others will come into contact with substantially more—possibly hundreds or even thousands. The second network we consider respects this observation [40]. On average people still have 16 contacts, but now the most well connected person has 454 while the most common number of connections is 8.

Another unrealistic feature of both the first and second networks is that they contain extremely few triangles. If we randomly choose two people, from anywhere in the world, it is of course extremely unlikely that they will be in direct physical contact with one another. If we pick two of your friends, however, then it's quite likely they will be in contact. The fact that two of your contacts are more likely

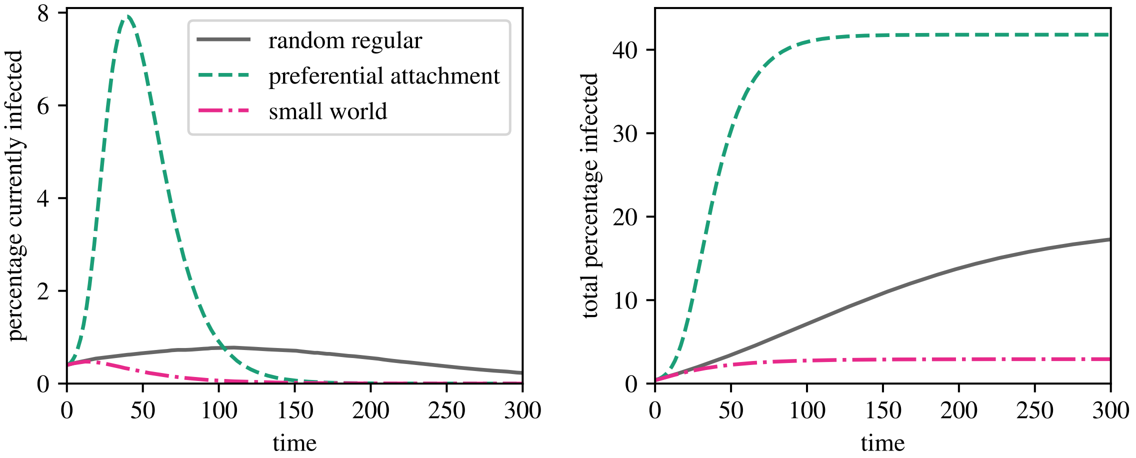


Figure 1.2: Simulations of the SIR model on a network. We set the disease transmission rate $b = 5/64$ and recovery rate $g = 1$. We considered three different networks. A *random regular* graph, in which all people have precisely 16 randomly selected connections [39]. A network created by *preferential attachment*, in which people *on average* have 16 connections, but some individuals have many more [40]. And a *small world* network (a Watts-Strogatz network), which has a lot of triangles [41]. We show the percentage of people currently infected and the total percentage of people who have ever been infected, plotted against time (in arbitrary units). Results are averaged over 500 simulations of the disease spreading on each of the 3 different networks. Each simulation starts with 40 random infections in the population.

to be in contact with each other corresponds to a large number of triangles in the contact network: if three people are all mutually connected, they form a triangle of connections. So, the third network we consider in Fig. 1.2 has a large number of triangles [41].

If you are not already familiar with networks, and you have understood the results of Fig. 1.2, you should find them surprising. We have run simulations of *the exact same disease* spreading on three networks. All three networks are exactly the same size and have the exact same density of physical contact. Further, the disease itself is extremely simplistic, devoid of any biological complexity whatsoever. And yet, the disease behaves radically differently in each case.

In one case (random regular), new infections continue to occur, slowly but steadily, throughout the whole simulation. If we make the network more realistic by adding variation to the number of contacts (preferential attachment), we see a colossal spike of infections and over 40% of the population infected at some point. On the other hand, when we make the network more realistic in a different way—by adding triangles (small world)—the disease simply can't take hold and dies out almost immediately.

The result is that any epidemic threshold—any measure such as R_0 that seeks to predict whether a disease will spiral into a devastating epidemic—cannot be constructed from properties of the disease and contact density alone. Instead, the microscopic details of the network appear to matter. Precisely who is connected to whom actually matters. This realization gets to the heart of deep problems in network science and hopefully provides the reader with sufficient justification that network structure matters.

We have looked at infectious disease as a case study, but the same concerns generalize. Network structure is important for the robustness and optimality of infrastructure [31], brain function [17], the health of ecosystems [21], and social and political power [42], among other things. In fact, networks are useful because structure often matters. They are used precisely because assumptions such as the well-mixed hypothesis are often inappropriate.

But if we conclude that the exact details of network structure are important, are we now at a loss? Is there anything we can say in general? Or, must we consign ourselves to running detailed simulations anew, for each and every network? Thankfully, we do not.

One route to progress is to develop solutions (or approximate solutions) for problems on general networks. Sticking with epidemics for now, a mean-field analysis exposes the intimate relationship between the largest eigenvalue of the adjacency matrix and disease spreading [37]. A refinement to this calculation clarifies that it is not actually the eigenvalue of the adjacency matrix that matters, but that of the closely related *non-backtracking matrix* [43]. In Ch. 5, we will further refine these calculations.[3]

Mathematical analyses allow us to connect particular properties of networks (such as associated eigenvalues) to phenomena of interest. Thus, while the details of network structure do matter we can often summarize the relevant features succinctly. This is welcome news. Grasping a handful of numbers is something we are good at; directly comprehending a large graph is not. Accordingly, significant effort in network science has been dedicated to the development of useful metrics and summary statistics. Such statistics communicate key information about structure while remaining intelligible.

1.2 Measures and metrics

Measures of *centrality* are particularly widely used [1, 46, 47]. These metrics assign scores to nodes (or more rarely edges, e.g. [48]) in order to characterize how central or important they are to the network. Different metrics correspond to different notions of what it means to be central or important.

The simplest centrality measure is degree centrality [1]. The degree of a node is the number of edges it has. The degree of node i is often denoted k_i where

$$k_i = \sum_j A_{ij}. \tag{1.8}$$

When we use degree to measure centrality, the implicit assumption is that nodes that have lots of connections are more prominent or central. This view, however, counts all connections equally and doesn't consider to which nodes a node is connected.

An alternative logic of centrality relies on a positively circular definition. What does it mean to be an important person? Well, one could argue that an important person is anyone who is well connected to the important people. This might seem inadequate to a lexicographer but we can make good sense of it mathematically. For example, the eigenvector centrality [49] assumes that the centrality of a node is

[3] We will not explicitly consider epidemics, but rather the mostly equivalent problem of bond percolation. See [44, 45] for discussion of the equivalence.

proportional to the sum of its neighbors' centralities. Node i's centrality, x_i, is thus

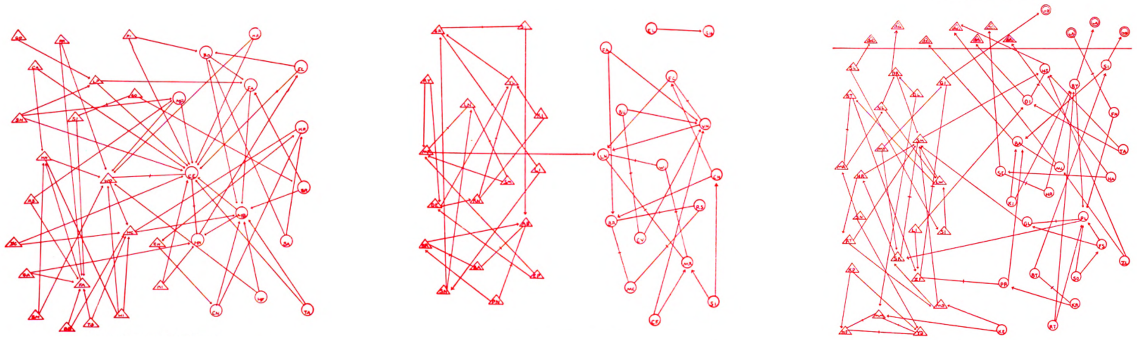
$$x_i = \frac{1}{\lambda} \sum_j A_{ij} x_j. \tag{1.9}$$

Assuming this centrality score isn't negative, Eq. (1.9) has a single solution: $\boldsymbol{x}$ is the leading eigenvector of the adjacency matrix, A. Variations on this theme are the centrality of Katz [50], the HITS algorithm [51], or the PageRank algorithm of Google [52].

Many further alternatives exist for defining centrality, for example those based on flows or paths (see [1,47]). Such centrality measures are focused on ranking the centrality of individual nodes (or edges), but taking a wider view leads to the idea of a dense *core* of a network, in contrast to its *periphery* [53]. A network may be extremely large, but perhaps only a fraction of it is well connected, while the rest is naught but hangers-on. By precisely defining a metric this can be quantified and cores can be identified and studied [54–56].

Regardless of precisely how we define the notion of a densely connected core, we will have in mind some notion of density for moderately sized subsections of the original network. However, we can also consider density from a more local perspective. As we have already discussed, any two of your friends are relatively more likely to be friends with each other. A simple measure of this is the transitivity or clustering coefficient. You, along with every pair of your friends, can potentially form a triangle. The transitivity coefficient is the fraction of these possible triangles that actually exist [1]. Or, we can apply this same concept to individual nodes and assign each node its own local clustering value [41]. High clustering scores correspond to networks that are *locally dense*. To make this connection clear, consider just the small network made up of you and your friends. If many of your friends are also friends with each other then there will be a relatively large number of edges in this local network.

The measures listed above merely quantify differences in centrality and density, but variations in density generally aren't random. Rather, they often display systematic patterns. Networks frequently display *assortative mixing* or *homophily* where significantly more network connections occur between nodes that are alike in some manner than between those that aren't [57,58]. For example, while high schools in America are no longer segregated in theory, they may be somewhat segregated in practice. Even within ethnically diverse schools students disproportionately form within-group friendships; mixed schools need not be particularly well mixed [59].



Class structure, 1st grade

Class structure, 4th grade

Class structure, 8th grade

Figure 1.3: Moreno's sociograms, taken from [60]. These drawings show the structures of a 1st, 4th, and 8th grade class. Boys are marked with triangles, girls with circles. Sorting by gender is weak in the youngest children and increases significantly before decreasing again as they become teenagers.

Modularity and the *assortativity coefficient* provide popular measures to quantify such effects [1,58].

Analyses of mixing are foundational to network science. We have presented mixing as simply one of many things one may wish to measure in a network. A more historical presentation might give the reverse impression: networks were actually developed in order to study social patterns of mixing. Jacob Moreno's *sociograms* are arguably the first appearance of networks in the scientific literature [60]. Figure 1.3 shows examples of these sociograms, visually displaying patterns of gender mixing. Despite this long history, less attention has been paid to mixing from the viewpoint of individuals, a point to which we shall return in Ch. 2.

We have been brief in our overview of network measures. Overviews of prominent measures can be found in [1,61]. Less prominent ones abound. While a lot of effort has been expended in developing metrics, many are ultimately ad-hoc, codifying particular intuitions of researchers. Some are considerably more principled, for example centrality measures defined with respect to specific dynamic processes [62]. But ultimately, metrics are not explanatory. We can rank how central each node is in any given network, but why are some nodes more central than others? Are there really significant differences, or are we investing effort into the study of statistical noise? If we dutifully collate network data sets and tabulate their properties, a general understanding of networks is unlikely to jump out at us. Instead, for theoretical enlightenment we need to model network structure itself—model how this or that structure might arise.

1.3 Models of networks

Network models specify how network structure itself is generated. We have, in fact, already used three: the random regular graph, preferential attachment, and the Watts-Strogatz model. Modeling the structure of complex networks dates back to at least the 1950s with the work of Solomonoff and Rapoport [63].

The work of Erdős and Rényi [64, 65] is particularly well known.[4] The model starts from the assumption that we want a network with n nodes and m edges. To construct such a network, we simply place the m edges uniformly at random between nodes. This model is (unsurprisingly) fairly boring, but not entirely so.

The first interesting property of this model is the component structure. A component in a network is a set of nodes that are reachable from one another. If someone is a friend of a friend of a friend... of a friend of yours, they are in your component. Unless you have literally no friends, you can probably reach most people in the world through chains of this sort.

When m is extremely small, Erdős-Rényi networks are composed of many small fragments that are disconnected from one another. However, once $m > n/2$ we will generally observe a large connected component—a significant fraction of nodes will be reachable from each other. And, once $m > n \log n$, this large component will most likely contain every single node. For comparison, there are $\binom{n}{2} = (n^2 - n)/2$ possible locations for an edge, a quantity that is much larger than either $n/2$ or $n \log n$ (for large n). These results show that even fairly sparse networks (networks with relatively few edges) are generally well connected—most nodes can generally reach most others through some chain of connection.

A second interesting property of these networks is how short these chains actually are. Any node in the large component can typically reach any other in only a few hops. Specifically, the number of hops needed grows logarithmically with the number of nodes, which gives rise to a *six degrees of separation* effect [1]. If it is true that we can reach anyone in the world through only six links, then in a galactic empire with ten billion times the population of Earth, the Erdős-Rényi model predicts we would still only need twelve. Whatever the exact numbers turn out to be, short paths are not surprising once we have studied the Erdős-Rényi model.

While the Erdős-Rényi model is generally quite bad at modeling real-world

[4] The model itself is almost equivalent to the models of Solomonoff and Rapoport [63], and Gilbert [66].

networks, variants improve it significantly. The first model we used in our earlier discussion—what we called random regular—is a simple variant. Instead of only fixing the number of edges in the network we also fixed the exact number of edges each node had. We chose this to be the same for each and every node, but in general this may vary, which leads to the configuration model [39,67]. In the configuration model, we connect nodes at random, but insist on precisely fixing the number of edges that each node has. Analysis of this model allows us to understand the generic effects of differently distributing the edges between nodes [68].

Another important variant is the stochastic block model (or planted partition model) [69,70]. In this model we imagine nodes are sorted into blocks or groups, and the number of edges within and between groups is adjusted. This serves as a basic tool for understanding the effects of different mixing rates. Or, we can combine both the configuration model and the stochastic block model, and account for both an uneven distribution of edges and non-trivial mixing patterns [71].

The second model we used in our earlier discussion—preferential attachment—has a more mechanistic flavor [40,72]. This model starts with an arbitrary small network and then sequentially adds to it. Nodes are introduced one at a time, and each new node chooses some number of old nodes to connect to. However, instead of choosing totally at random, nodes *preferentially attach* to nodes that already have lots of edges—a "rich get richer" effect. In the standard formulation, new nodes connect to old nodes proportional to their degree.

Preferential attachment leads to a handful of nodes with an extremely large number of edges, while most nodes have only a few. The distribution of edges (the degree distribution) follows a power law. The probability that a node has degree k, for large k, is

$$p_k \propto k^{-3}. \tag{1.10}$$

Among other things, this provides an explanation for why some scientific papers receive thousands of citations while others (of seemingly similar quality) may receive almost none [72,73].

The Watts-Strogatz network [41], the third model we used earlier, doesn't posit a mechanism per se but is built from a highly ordered structure. We start by lining up the nodes, one after another, and then connect each node to those close by. Each node is initially connected to the k nodes closest to it—$k/2$ to the left, and $k/2$ to the right. This produces a highly ordered network that is locally dense. Suppose that you are one of the nodes in this network, connected to the $k/2$ nodes to your right, and $k/2$ to your left. The node directly to your right will also be connected to all of your other neighbors on your right, and all but one of those on your left. This leads to lots of triangles—the friends of your friends are statistically likely to be friends.

However, such an ordered lattice is extremely unlike a real-world social network. In particular, you would need a large number of hops to reach most other nodes in the network. To rectify this we randomly re-wire some of the edges. Watts and Strogatz noticed that with just a small amount of random re-wiring, we achieve networks with short distances between nodes, *and* lots of triangles (a situation that had once seemed theoretically puzzling [74]). Two of the most striking properties of real-world networks can be simultaneously produced by a large degree of order and a small sprinkling of randomness.

As for measures and metrics, there are too many network models to cover in any single review. The most prominent models are covered in detail in [1–3]. New models are posted weekly (or even daily) to arXiv. Modeling, however, is not merely a theoretical tool for understanding. It also allows us to infer things about data that are not otherwise directly observed or observable.

1.4 Inference

The topic of inference is well illustrated by example. As we have discussed, networks often display assortative mixing—nodes that are similar connect to each other at higher rates than those that are dissimilar. The modularity metric measures the extent to which this is true—it measures the extent to which a network breaks apart into tight-knit communities of similar nodes—but this assumes we already know which nodes are in which groups. If we *don't* already know the division of nodes into groups, we can turn modularity on its head and use it to *discover* a division of the nodes into groups. Instead of considering the group of each node to be some fixed property, we can consider many different assignments of groups to the nodes. By picking an assignment that maximizes modularity we find a good division of the network nodes into communities [75, 76]. This procedure (maximizing the modularity) is simply one of many *community detection* methods [77].

Today, the go-to method for rigorous community detection is built on the stochastic block model, and uses statistical tools such as maximum likelihood estimation [69,71,77].[5] The stochastic block model can fit arbitrary mixing patterns,

[5] Modularity maximization was initially motivated by intuition. It was later put on a firm footing

and by fitting the model to data we can make principled inferences about what we have observed.

The general form of these inference procedures is as follows. The model under consideration (be it the stochastic block model or any other one), assigns a probability to observing adjacency matrix A given some parameters θ. This is written $P(A|\theta)$. Our job is to make inferences about θ. The method of maximum likelihood tells us to estimate θ by maximizing $P(A|\theta)$. Alternatively, we can use Bayes' rule to infer a distribution for θ ,

$$P(\theta|A) = \frac{P(A|\theta)P(\theta)}{P(A)} \tag{1.11}$$

(see Ref. [79] for an overview of these ideas).

By working with well defined statistical models we are able to make full use of modern inferential statistics. For example, it is usually unclear a priori how many communities one should look for in a network. By framing our problem as a statistical inference, we can make principled decisions and develop methods that automatically detect how many communities there are [80,81]. Such inferences allow us to perform more advanced data analyses which both helps uncover relations that are otherwise obscure, and quantifies our confidence in such results.

Inference problems are also often interesting in their own right. For example, they may display *phase transitions*. For one set of model parameters the inferences may be easy, while for another, they may be impossible [82]. For the stochastic block model, it has been shown that unless the network has sufficiently strong community structure, recovering the groups is impossible [83]. This impossibility is quite profound. If people do, in fact, preferentially form within-group attachments, but if these within-group attachments are not considerably more prevalent than out-group attachments, then no network process would ever know the difference—the communities might as well not exist.[6]

For narrative simplicity, we have mostly focused our discussion of inference on the problem of community detection. However, inference in networks is a broad and growing field. Other inference tasks that have received considerable attention are: missing data, hierarchy, latent space, edge prediction, spreading processes,

by demonstrating its equivalence to maximum likelihood estimation of a particular variant of the stochastic block model [78].

[6] This result relies on the assumption that all individuals within a group are statistically identical. In Ch. 3 we'll return to this and show a more nuanced picture arises once one allows for individual variation.

network evolution, and even inferring the network itself [84–90].

1.5 Contributions of this thesis

Before outlining my own contributions, it is appropriate to consider oversights of the network science literature. Time and again, models and theoretical tools neglect the highly correlated nature of networks. For example, in the stochastic block model nodes are randomly assigned to groups. Once these groups have been assigned all edges are statistically independent. This leads to networks that look completely different to real-world networks. Independence assumptions are almost never plausible and are made not on the basis of sound scientific arguments but for mathematical convenience.

Well aware of the implausibility of independence assumptions, sociologists frequently use a class of models—exponential random graph models [91,92]—that can easily introduce correlations. Unfortunately, these have significant flaws and have largely been abandoned by theoretical network science [93–95].

The more mechanistic models, such as preferential attachment, do not usually entail independence and produce networks with strong but subtle interdependence. However, these models tend to be mathematically intractable and thus aren't actually useful for analyzing real data (or at least, we don't yet know how to do this). Recently, some progress has been made with new computational techniques, although these still require massive computational power (e.g. [89,96]).

Not only do the models of network science assume independence, so too do the mathematical tools. Mean-field analyses along with more advanced tools such as message passing methods, entail strong independence assumptions that break down in the presence of triangles. Key results of theoretical network science—such as calculations for the spectra of networks [97]—break down with the introduction of triangles, the most simple non-trivial structure.

As a general theme, this thesis presents work that takes correlation and interdependence seriously. We will develop measures, models, and mathematical tools that proceed from an assumption of interdependence. By embracing correlation, rather than neglecting it, we hope to narrow the gap between networks *in theory* and networks in the real-world.

In Ch. 2 we return to the study of mixing patterns but reject the conditional independence underlying many measures. Instead, we allow for the fact that otherwise similar nodes may systematically differ in their behavior—individual nodes may connect to others at higher (or lower) rates than expected. Such systematic differences correspond to correlations between the connections of any individual. Our analyses of these correlations provide principled estimates of mixing, at both a global and individual level, along with confidence intervals on our estimates. In contrast, most previous measures either over-fit the data or simply average out any diversity. This chapter is based on previously published work with Mark Newman [98].

In Ch. 3 we push these analyses of mixing further. We show how our insights about mixing can be used to recover missing data or perform community detection. While traditional methods (based on the stochastic block model) only perform well at these tasks when the mixing is sufficiently (dis-)assortative, our methods may still work even in the total absence of assortative structure. We locate a phase transition for inference when individuals' choices are correlated and show that such patterns actually make recovery tasks easier. Thus, so long as individuals are not all exactly alike, the pessimistic results based on the stochastic block model do not necessarily carry over to the real world. The results presented in this chapter are currently not published elsewhere.

In Ch. 4 we consider a very simple null model for networks. We start from the quite reasonable assumption that our data is drawn from a multivariate normal distribution, with one free parameter (a covariance parameter). Then, we create the network by simply thresholding this data: large values correspond to edges, while small values are non-edges. This is almost the simplest model for how a network data-set might be constructed, and yet we find the introduction of correlations leads to several of the properties commonly associated with complex networks: namely heterogeneous degree distributions, short geodesic paths, and relatively large numbers of triangles. The work presented in this chapter was conducted with Yanchen Liu, Ben Maier, Alice Schwarze, Carlos Serván, Jordan Snyder, and Guillaume St-Onge, and has been published [99].

In Ch. 5, we turn our attention away from network models and towards theoretical tools. Message passing, a fundamental technique for performing calculations on networks and graphs, has long been recognized to work poorly on networks that contain short loops. We develop a framework to address this issue and create methods that work on arbitrary networks. We exemplify this with two applications. First, we derive new results for bond percolation on networks—a process that serves as a basic model of the spread of disease. Second, we show how to approximate the spectral density of a sparse matrix extremely efficiently—heuristic arguments suggest O($n \log n$) time. This chapter is again based on previously published work with Mark Newman [100].

Following on, in Ch. 6, we show how these same insights can be used to compute partition functions for high-dimensional complex models. We derive a factorization for probability distributions defined on networks and we demonstrate how to apply this with the Ising model as a case study. The method is broadly applicable, and should help future analysis of network models to escape from "locally tree-like" assumptions. In this chapter I make use of currently unpublished work with Alec Kirkley and Mark Newman.

Finally, in Ch. 7, I conclude with some reflections on the field and its future direction.

CHAPTER 2

Mixing Patterns and Individual Differences

This chapter is adapted from the published results of G. T. Cantwell and M. E. J. Newman, Mixing patterns and individual differences in networks. *Physical Review E* **99**(4), 042306 (2019) [98].

As we discussed in the introduction, a common feature of many networks is *assortative mixing*, the tendency of network nodes to be connected to others that are similar to themselves in some way [57, 58, 101, 102]. On the World Wide Web, for instance, one might expect web pages to link to others written in the same language more than they do to ones in different languages. In friendship networks (where the phenomenon is also known as *homophily*) many individuals have a preference for friends who are similar to themselves in terms of age, race, educational level, and other characteristics [57–59, 101]. One can also encounter *disassortative* mixing, the tendency for nodes to connect to unlike others [58, 102].

Assortative mixing has been studied widely. Researchers have examined and quantified assortativity as it occurs in a wide variety of real-world networks [57,58, 101] and created mathematical models such as the planted partition model [103,104] and the stochastic block model [69] that can mimic both assortative and disassortative behaviors. These methods and models, however, capture only the average mixing behavior of nodes, the average preference for members of one group to forge connections with another. There can be, and in many cases is, substantial variation about the average; all members of a group do not necessarily behave exactly alike.

As an example, networks of romantic interaction between individuals are mostly disassortative by gender: a majority of individuals have a preference for romantic engagements with members of the opposite sex. On the other hand, some people prefer romantic engagements with the same sex. Standard measures of overall assortative mixing would thus say that the average individual has a small fraction of same-sex relationships and the rest are opposite-sex. But this is misleading: in fact, many individuals have strong preferences for one or the other, so the "average preference" does not, in this case, provide a good description of individual behaviors.

Furthermore, there can be interesting mixing patterns even when there is little or no average assortativity in a network. For example, a recent study of friend networks on Facebook showed little to no gender assortativity on average, yet *some* people do appear to have preferences [105, 106]. Some individuals on Facebook strongly prefer either male or female friends—it is only when we average over the whole population that we see no effect. Thus, traditional measures of average assortativity do not tell the whole story.

There has been some previous literature discussing these phenomena and advocating a move beyond average measures of assortativity. In the study of Facebook mentioned above, Altenburger and Ugander [106] introduced the concept of *monophily*, the extent to which people's friends are similar to one another, while Peel *et al.* [107] define a variant assortativity coefficient that characterizes assortativity within a local neighborhood in a network. Other approaches have defined an assortativity coefficient at the level of individual nodes [108, 109].

In this chapter we demonstrate that inferring and quantifying individual differences in mixing is not trivial, in practice or in principle, an observation that bears emphasizing. The difficulty is not simply due to a lack of data. Even for arbitrarily large networks naive approaches will fail. To address these issues we introduce a principled and general method for analyzing mixing patterns in networks that does not require large amounts of data or lengthy computations. Our solution employs a generative stochastic model of individual-level mixing, showing how it can be used to model and analyze empirical network data. Crucially, the model allows for arbitrary mixing patterns and does not assume that individuals behave in accordance with the average within their group. By fitting the model to data using statistical methods we infer quantities that have straightforward interpretations and can thus be used to characterize mixing patterns, in much the same way that the parameters of a normal distribution characterize mean and variance.

The model we study is conceptually similar to others that have been studied previously. It shares with the well-known stochastic block model [69] the ability to represent arbitrary mixing patterns at the group level, but also goes further, allowing for individual variation within the groups. A model for individual variation was introduced previously in [106], but it does not allow for arbitrary mixing patterns, nor was a direct method proposed to fit the model to data. Variation within groups can be approximated with mixed membership models [110, 111], in which network nodes can be members of multiple groups and inherit the mixing patterns of all of their groups. Two nodes in a given group might, for instance, be in different other groups and hence need not mix equivalently. This approach is of little use, however, when group memberships are already known or the categories are known to be distinct and non-overlapping. If we want to model individual differences in the social mixing patterns of men and women, for instance, we are not at liberty to re-assign genders so that our model fits.

As a demonstration of our methods, we apply them to two example networks, a friendship network of high school students and a linguistic network of word adjacencies in English text. We find that there is indeed substantial individual variation in mixing patterns in both networks, implying that traditional average measures of mixing offer an incomplete description of network structure.

2.1 Individual preferences and patterns of connection

We consider networks in which the nodes are divided into a number of discrete, non-overlapping groups, types, or categories, and where individual nodes have *preferences* about the types of the nodes with which they have network connections. We will focus on labeled networks, meaning ones in which the type of every node is known in advance—we are told the sex of each individual in a social network, for example, or the language that each web page is written in. Our network could be directed or undirected, but we will concentrate primarily on the directed case here, treating the undirected one as the special case when all edges are reciprocated.

In the context of such labeled network data, how should one define preference? By any reasonable definition, if a node has a strong preference to connect to others of a certain type then we should expect there to be a relatively large number of edges to that type. Let us denote the number of edges from node i to nodes of type s by k_{is} and the total number of edges from i to nodes of all types by $k_i = \sum_s k_{is}$. Then the ratio k_{is}/k_i is the fraction of edges from node i to nodes of type s.

This ratio, however, is not necessarily an accurate guide to i's preference for connections to type s. We should expect there to be some statistical fluctuations in the network formation process, so that high or low values of k_{is} could occur just by chance. Let us define a quantity x_{is} to represent i's underlying preference for nodes of type s, which will be equal to the expected value of the ratio k_{is}/k_i, averaged over these fluctuations:

$$x_{is} = E[k_{is}/k_i], \tag{2.1}$$

where we restrict ourselves to nodes i with non-zero degree (the value of x_{is} is not well-defined when $k_i = 0$). Note that x_{is} as defined is automatically normalized so that $\sum_s x_{is} = 1$. Note also that the ratio k_{is}/k_i is, by definition, an unbiased estimator of x_{is}, though it is not necessarily a good estimator. In fact, as we demonstrate below, for many purposes it is highly misleading.

One way to think about Eq. (2.1) is to imagine creating the same network many times over and averaging over the randomness in the creation process to calculate x_{is}. Unfortunately, in the real world we normally get to observe a network only once and hence we cannot perform the average. This is the root cause of the difficulty with estimating preferences that we mentioned above.

To proceed any further we need to know more about the nature of the fluctuations in the values of the k_{is}. If we can define a sensible model for these fluctuations then we can make progress estimating x_{is} using the tools of statistical inference.

2.1.1 Preference-based network model

How is k_{is} generated? We could imagine that node i considers every other node in turn and connects to those in group s with some probability λ_{is}, which measures i's affinity for group s. Then the edges of the network would be Bernoulli random variables with means λ_{is}, which in standard statistical notation would be written $A_{ij} \sim \text{Bernoulli}(\lambda_{ig_j})$, where A_{ij} is an element of the adjacency matrix, having value one if there is an edge from i to j and zero otherwise, and g_j is the group or type label of node j.

This, however, is unsatisfactory for two reasons. First, as is often the case, it is simpler to use a Poisson rather than Bernoulli distribution: $A_{ij} \sim \text{Poisson}(\lambda_{ig_j})$. In a sparse network where $\lambda_{is} \ll 1$ the two distributions are nearly identical, but the Poisson distribution offers significant technical advantages. Second, and more importantly, many networks have broad degree distributions that are not well captured by either the simple Bernoulli or Poisson model. This issue can be dealt with by "degree-correction" [70, 71, 77], which in this context involves the introduction of two additional parameters ϕ_i and θ_i for each node i, which respectively control the in- and out-degrees of the node. (In an undirected network, the two would be equal $\phi_i = \theta_i$.) With

$$\Phi_s = \sum_{i \in s} \phi_i \tag{2.2}$$

denoting the sum of all ϕ_i for nodes in group s, we let

$$A_{ij} \sim \text{Poisson}\left(\frac{\theta_i \phi_j x_{i g_j}}{\Phi_{g_j}} \right). \tag{2.3}$$

This definition does not unambiguously set the values of the parameters, since we can multiply the values of all the ϕ by any constant factor without affecting the A_{ij} or any other property of the model. One can fix this by choosing a normalizing condition for the ϕ_i, such as requiring that they sum to 1, but this will not be necessary for any of the calculations presented here.

Note that the choice of a Poisson rather than a Bernoulli distribution in Eq. (2.3) implies that the network may have multiedges—there may be two or more edges running between the same pair of nodes, so that $A_{ij} > 1$. On a sparse network, however, this happens vanishingly often and multiedges can normally be neglected [71].

For a better intuition on the role of the parameters in the model, it is instructive to consider the distributions of the quantities k_{is} and k_i. Given that the A_{ij} are independent Poisson random variables and that a sum of Poisson variables is itself Poisson, the distributions for k_{is} and k_i are also Poisson:

$$k_{is} \sim \text{Poisson}\big(\theta_i x_{is}\big), \tag{2.4}$$

and

$$k_i = \sum_s k_{is} \sim \text{Poisson}\left(\theta_i\right). \tag{2.5}$$

Thus θ_i is equal to the expected out-degree at node i, independent of the node's preferences. A simple further computation verifies that x_{is} is indeed the expected value of k_{is}/k_i, consistent with the definition of preference, given in Eq. (2.1).

We favor this model for the intuitive interpretation of its parameters along with the mathematical simplicity of the Poisson distribution.

2.1.2 Inferring individual preferences

Given the types of the nodes, we can now write down the probability of observing any given pattern of connections at node i:

$$\begin{aligned} P(\boldsymbol{A}_i|\boldsymbol{x}_i, g, \theta, \phi) &= \prod_j P(A_{ij}|\boldsymbol{x}_i, g, \theta, \phi) \\ &= e^{-\theta_i} \prod_j \left(\frac{\theta_i \phi_j x_{ig_j}}{\Phi_{g_j}} \right)^{A_{ij}} \frac{1}{A_{ij}!}, \end{aligned} \tag{2.6}$$

where $\boldsymbol{A}_i$ denotes the ith row of the adjacency matrix and $\boldsymbol{x}_i$ is the vector with elements x_{is}. The probability of observing the whole network is then the product

$$P(A|x, g, \theta, \phi) = \prod_i P(\boldsymbol{A}_i|\boldsymbol{x}_i, g, \theta, \phi). \tag{2.7}$$

The terms in Eq. (2.7) that depend on θ and ϕ can be factored out from those that depend on x and thus one can write

$$P(A|x, g) = \frac{1}{Z} \prod_{i,s} x_{is}^{k_{is}} \tag{2.8}$$

where Z is a constant that depends on A and g but not x.

Given both the categories and the network structure, we can use the model to infer the preferences $\boldsymbol{x}_i$. A tempting approach is to use maximum-likelihood estimation. However, maximization of Eq. (2.8) with the constraint that $\sum_s x_{is} = 1$ just leads back to the estimate $\hat{x}_{is} = k_{is}/k_i$. As we now argue, if we want to learn about the distribution of preferences, these estimates may be misleading.

Consider, for illustrative example, the case in which all nodes in a group have the same parameter values. (This case is equivalent to the stochastic block model.) Even though all nodes have the same preferences, k_{is}/k_i will not be the same for every node, since it is a random variable. Worse, it will often have significant variation. Figure 2.1 shows an example of this situation.

Things are not too bad if we only want to measure the average preferences in a group: we can average over the values of k_{is}/k_i for all i in the group in question and the fluctuations will average out. For anything beyond average-level behavior, however, we are not so lucky. As demonstrated in Fig. 2.1, even something as simple as the variance of x_{is} is not straightforward to estimate from k_{is}/k_i.

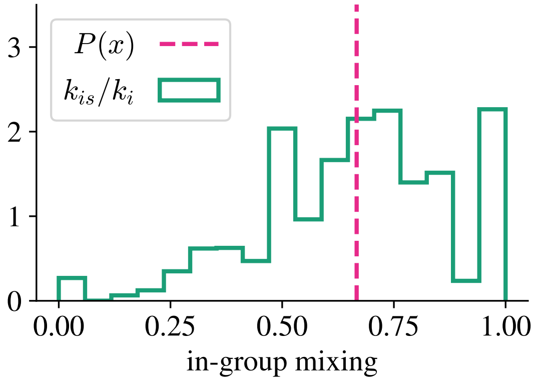


Figure 2.1: Histogram for k_{is}/k_i in the model of Eq. (2.3). We set $\theta_i = 6$ and $x_{is} = \frac{2}{3}$. The dashed line is at $\frac{2}{3}$, the true value of x_{is}. For an arbitrarily large network with these parameters, the dashed line is the true distribution of preferences, while the histogram corresponds to the inferred distribution, if we used the maximum-likelihood estimator $\hat{x}_{is} = k_{is}/k_i$. The distribution of k_{is}/k_i is clearly a poor approximation for the true distribution $P(x)$.

The root of the problem is the sparsity of the network. When we only have a handful of connections for each node, the ratio k_{is}/k_i will be broadly distributed even when all x_{is} are the same. This is not due to our networks being too small. The amount of network data we have grows larger as the network does, but so too does the number of parameters we are estimating, and it is straightforward to show that the expected variation of the individual estimates k_{is}/k_i will not vanish even in the large size limit.

To get around this issue we need some way to accurately characterize individual preferences that does not require an extensive number of parameters. Here we do this by inferring the underlying distribution from which the $\boldsymbol{x}_i$ are generated. We describe this procedure in the next section.

2.2 Distributions of preferences

Suppose the preference variables x_{is} for nodes in group r are drawn from a distribution $P(\boldsymbol{x}|\boldsymbol{\alpha}_r)$, where $\boldsymbol{\alpha}_r$ is a set of parameters for the distribution. If we know this distribution then we can integrate over the unobserved preferences in Eq. (2.8) and compute the likelihood of the network thus,

$$P(A|\alpha, g) = \frac{1}{Z} \prod_i \int \Big(\textstyle\prod_s x_s^{k_{is}}\Big) P(\boldsymbol{x}|\boldsymbol{\alpha}_{g_i})\, d\boldsymbol{x}. \tag{2.9}$$

Rather than infer the individual preferences directly we can, using this likelihood, infer their distribution by fitting the parameters α.

The only constraints on $\boldsymbol{x}$ are that $x_s > 0$ for all s and $\sum_s x_s = 1$, meaning that the vector $\boldsymbol{x}$ lies on the standard unit simplex and $P(\boldsymbol{x}|\boldsymbol{\alpha}_r)$ can be any distribution on the simplex. Here we make the simple and common assumption that $P(\boldsymbol{x}|\boldsymbol{\alpha}_r)$ is a Dirichlet distribution [79]. For a network with c groups the Dirichlet distribution takes the form

$$P(\boldsymbol{x}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{s=1}^{c} x_s^{\alpha_s - 1}, \tag{2.10}$$

where $\alpha_s > 0$ for all s and $B(\boldsymbol{\alpha})$ is the multi-dimensional beta function

$$B(\boldsymbol{\alpha}) = \frac{\prod_s \Gamma(\alpha_s)}{\Gamma(\alpha_\Sigma)}, \tag{2.11}$$

with $\alpha_\Sigma = \sum_s \alpha_s$ and $\Gamma(\alpha)$ being the gamma function. The Dirichlet distribution is a convenient and flexible distribution that allows us to vary the weight placed on each of the x_s independently. In the case of two groups, $c = 2$, the Dirichlet distribution is equivalent to the beta distribution. The expected value of $\boldsymbol{x}$ within the distribution is $\boldsymbol{\alpha}/\alpha_\Sigma$, and α_Σ controls the width of the variation about that value. In the limit of large α_Σ the variance tends to zero and the distribution of $\boldsymbol{x}$ is tightly clustered around the mean. Conversely, as α_Σ tends to zero almost all the probability density is in the corners of the simplex, as far away as possible from the mean.

We allow each group or type s to have a different distribution of preferences and hence a different set of Dirichlet parameters $\boldsymbol{\alpha}_s$, so that the prior on $\boldsymbol{x}_i$ is

$$\boldsymbol{x}_i \sim \text{Dirichlet}(\boldsymbol{\alpha}_{g_i}). \tag{2.12}$$

This is a natural choice: one can well imagine, for instance, that the men and women within a population have different preferences for male and female friends.

With this choice we can now complete the integrals in Eq. (2.9) and we find that

$$P(A|\alpha, g) = \frac{1}{Z}\prod_i \frac{B(\boldsymbol{\alpha}_{g_i} + \boldsymbol{k}_i)}{B(\boldsymbol{\alpha}_{g_i})}, \tag{2.13}$$

where $\boldsymbol{k}_i$ is the vector with elements k_{is}. Estimates for the α parameters can now be obtained by maximizing this likelihood.

Under certain circumstances, Eq. (2.13) may lack a well-defined maximum. To deal with this one can add a regularization term. The full details are given in Appendix A.1, but the end result is that one determines the estimated value $\hat{y}_{rs}$ by maximizing

$$L(y) = \sum_i \left[\ln B(e^{\boldsymbol{y}_{g_i}} + \boldsymbol{k}_i) - \ln B(e^{\boldsymbol{y}_{g_i}})\right] - \lambda \sum_{r,s} y_{rs}^2, \tag{2.14}$$

where λ is a small positive constant, and our estimate of α_{rs} is given by $\hat{\alpha}_{rs} = \exp \hat{y}_{rs}$. From a Bayesian perspective the quadratic regularization term is equivalent to a log-normal prior on α_{rs}.

Our reasoning up to this point can be summarized as follows. When we try to directly infer node preferences we find that the distribution of our estimates does not in general resemble the true underlying distribution, even for arbitrarily large networks. In contrast, maximization of Eq. (2.14) should give accurate estimates of α, at least for large networks, and to the extent that the underlying distribution can be well fit by the hypothesized Dirichlet distribution, these parameters will describe the shape of that distribution. Thus, it is now possible to infer preference distributions accurately so long as the network is sufficiently large.

In the real world we don't have arbitrarily large networks and so a different source of error could arise: the inability to make accurate estimates of α because our data are limited. One way to get around this problem is to take a Bayesian approach.

Bayes' theorem states

$$P(\alpha|A, g) = \frac{P(A|\alpha, g)P(\alpha)}{P(A|g)}. \tag{2.15}$$

The distribution $P(\alpha)$ is the prior distribution for the parameters, which we have to choose. Since the regularization term introduced in Eq. (2.14) is equivalent to a log-normal prior for α_{rs}, we propose using this form as a prior. More details are given in Appendix A.1.

A posterior distribution on α as above allows us to make estimates of quantities of interest without having to estimate α itself—we can average over it instead. In the next section we define some useful metrics that can be evaluated within the posterior distribution, and can thus be inferred in a parameter-free way.

2.3 Measures of assortativity and variation of preferences

In the previous section we described a procedure for inferring preference distributions in networks. The full multi-dimensional distribution, however, is difficult to interpret, so simple summary statistics are also useful. In this section we propose two specific measures that quantify the average assortativity in the network and the variation of preferences around that average.

Assortative mixing occurs when nodes have a preference for connecting to others of the same type. A natural measure of assortativity is the expected value of the in-group preference parameters. As discussed in Section 2.2, the expected value of the preference parameter x_{is} describing the preference of a node i in group r for connections to group s is $\alpha_{rs}/\alpha_{r\Sigma}$ where $\alpha_{r\Sigma} = \sum_s \alpha_{rs}$. The expected in-group preference of nodes in group r—their preference to connect to other members of the same group—is then equal to $\alpha_{rr}/\alpha_{r\Sigma}$, and the average in-group preference over all nodes in all groups is

$$a = \sum_r p_r \frac{\alpha_{rr}}{\alpha_{r\Sigma}}, \tag{2.16}$$

where p_r is the fraction of nodes that fall in group r.

In a perfectly assortative network all nodes connect only to their own group and $a = 1$, while in a perfectly disassortative network $a = 0$. For most real-world networks we expect the value to lie between these extremes, with higher values indicating more assortativity. A natural question to ask is what kinds of values do we expect to see? What constitutes a "high" value of a? One way to answer this question is to calculate the expected value within a null model.

A suitable null model in this case is one in which nodes are connected according to their expected degrees,

$$A_{ij} \sim \text{Poisson}\left(\frac{k_i k_j^{\text{in}}}{m}\right), \tag{2.17}$$

where k_i denotes the out-degree of node i, as previously, k_i^{in} denotes the in-degree, and $m = \sum_i k_i$ is the expected number of edges in the network. This is in essence just a directed version of the standard random network model in which we fix the expected degrees of all nodes, sometimes called the Chung–Lu model after two of the first researchers to examine its properties [112].

Applying the definition of preference from Eq. (2.1), all nodes in group s have the same preference in this null model, $x_{is} = K_s/m$, where $K_s = \sum_{i \in s} k_i^{\text{in}}$. Hence in this model the average in-group preference is

$$a_{\text{null}} = \sum_r p_r \frac{K_r}{m}. \tag{2.18}$$

The difference between the observed value of a, Eq. (2.16), and the expected value within the null model is then

$$a - a_{\text{null}} = \sum_r p_r \left(\frac{\alpha_{rr}}{\alpha_{r\Sigma}} - \frac{K_r}{m}\right). \tag{2.19}$$

When this quantity is greater than zero the preferences are more assortative than we would expect by chance. When it is less than zero the preferences are less assortative (or more disassortative) than expected. If we wish, we can normalize the difference so that it takes a maximum value of 1 at perfect assortativity, and thus define a preference assortativity coefficient

$$R(\alpha) = \frac{\sum_r p_r(\alpha_{rr}/\alpha_{r\Sigma} - K_r/m)}{\sum_r p_r(1 - K_r/m)}. \tag{2.20}$$

The range of allowed values is $R \in [R_{min}, 1]$, where in general $R_{min} \neq -1$ and depends on the network in question. (A similar behavior is seen for the conventional coefficient of assortativity defined in [58], which is essentially a Pearson correlation.)

In order to estimate the value of R, we need first to estimate the α parameters. As discussed in the previous section, we could do this by maximizing the likelihood of Eq. (2.13), but this may give poor estimates in cases, such as smaller networks, where the amount of available data is limited. An alternative approach is to compute the expected value of R in the posterior distribution of Eq. (2.15), thus:

$$R = \int R(\alpha)P(\alpha|A, g)\, d\alpha. \tag{2.21}$$

We can also compute the standard deviation of R in the posterior which makes it easy to state estimates with error bars. More details on this calculation are given in Appendix A.2.

The quantity R, however, only measures traditional assortativity. As we have said, our main purpose is to examine variation of individual preferences about group means. The variance of a Dirichlet distribution can be quantified by the mean-squared distance from its average. In group r this is

$$\sigma_r^2 = E_r\Big[\big(\boldsymbol{x} - E_r[\boldsymbol{x}]\big)^2\Big] = \frac{1 - \sum_s (\alpha_{rs}/\alpha_{r\Sigma})^2}{\alpha_{r\Sigma} + 1}. \tag{2.22}$$

As discussed in Section 2.2, the maximum value of the variance occurs when $\alpha_{r\Sigma} \rightarrow 0$, which gives $\sigma_r^2 = 1 - \sum_s (\alpha_{rs}/\alpha_{r\Sigma})^2$. One can divide by this maximum to give a normalized variance

$$V_r = \frac{\sigma_r^2}{1 - \sum_s (\alpha_{rs}/\alpha_{r\Sigma})^2} = \frac{1}{\alpha_{r\Sigma} + 1}, \tag{2.23}$$

which lies between zero and one and also has the nice property of being independent of the mean. Finally, we define an overall normalized variance coefficient by

$$V(\alpha) = \sum_r p_r V_r = \sum_r \frac{p_r}{\alpha_{r\Sigma} + 1}, \tag{2.24}$$

which also lies between zero and one. V can be estimated in the same way as R by averaging its value in the posterior distribution. See Appendix A.2 for further details of this calculation.

The quantity V represents the normalized mean-square distance between the preferences and their group means, averaged over all groups. When V is close to zero every node in every group has preferences close to the group mean. If preferences are homogeneous in this way then the network is well described by the group average mixing parameters and individuals' preferences are well described by simply stating which group they belong to. Such a finding could be informative for instance in a social network: it would tell us a lot about a population if we found that their preferences were entirely determined by, say, gender or race.

Network	R	V
College football [75]	0.60 ± 0.015	0.01 ± 0.004
Karate club [113]	0.72 ± 0.063	0.07 ± 0.059
Political books	0.72 ± 0.028	0.12 ± 0.034
Political blogs [114]	0.80 ± 0.010	0.15 ± 0.012
High school race or ethnicity [115]	0.55 ± 0.012	0.16 ± 0.011
Provisional IRA affiliation [116]	0.62 ± 0.025	0.22 ± 0.026
Word adjacency [117]	−0.27 ± 0.018	0.30 ± 0.024

Table 2.1: Estimates of normalized preference assortativity R and preference variance V for a selection of networks with known group assignments. Results are computed from the posterior distribution and stated as $\mu_X \pm \sigma_X$. Numbers in brackets indicate references for each network, except for the network of political books, which was compiled by Valdis Krebs and is currently unpublished.

At the other extreme, when V approaches one, node preferences are as far away from the group mean as possible, and nodes, even within the same group, are very unlike each other in their preferences. In this scenario mixing is poorly described by average rates, since virtually no nodes behave according to the average for their group.

2.4 Examples

Table 2.1 shows results for the preference assortativity and variance measures, R and V, for a selection of previously studied networks with known group assignments, listed in order of increasing variance. As the table shows, all of the networks are highly assortative by our measure, except for the word adjacency network, which is disassortative.

The normalized variances V take a range of values from zero up to 0.3. Recall that low normalized variance indicates a network in which the members of a group have similar preferences; high variance indicates that they have widely varying preferences. Thus, for instance, the "karate club" network, which is a social network of university students, appears to have no significant variance, meaning it shows traditional community structure in which the members of a community are roughly alike in their preferences. The network of high school students, on the other hand, which one might expect to be similar, shows higher variance. We discuss the high school and word networks in more detail below.

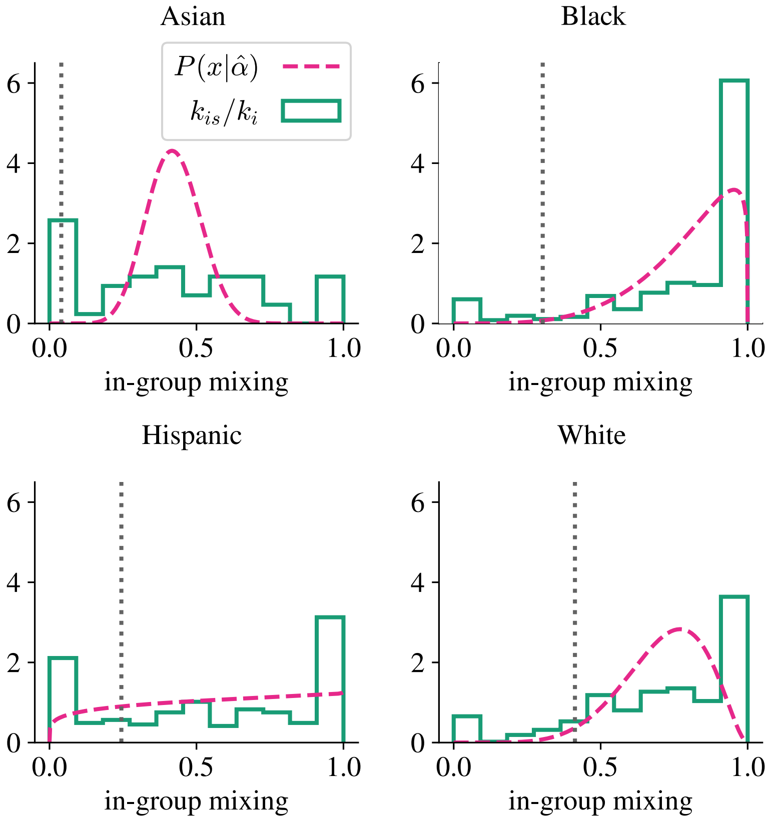


Figure 2.2: Friendship preferences by race or ethnicity in a US high school. We show separate results for Asian, Black, Hispanic, and White students. For each race or ethnicity the histogram (in green) shows the observed distribution of k_{ig_i}/k_i, the naive estimate of within-group preference. The red dashed line is the inferred preference distribution from a point estimate of α, found by maximizing Eq. (2.14). The gray vertical line is where the average preference would be, in the absence of assortativity.

2.4.1 High school friendships and ethnicity

The network denoted “High school race or ethnicity” in Table 2.1 is a network of self-reported friendships between students in a US high school, taken from the National Longitudinal Study of Adolescent to Adult Health [115] (commonly known as the “Add Health” study). The node labels in this case represent the (self-identified) ethnicities of the students, which take values “Asian,” “Black,” “Hispanic,” “White,” “other,” and “missing.” In our analysis we discard the “other” and “missing” categories and focus on the remaining four. The particular school we look at is chosen for its diverse racial and ethnic composition.

The value of $R = 0.55 \pm 0.012$ for this network indicates that the school is strongly assortative by race, meaning that students had more within-group friendships than would be expected by chance. However, the groups also display differences in the inferred distributions of their preferences, which are plotted in Fig. 2.2. Hispanic students, for instance, show a larger range of preferences than others. Note that this doesn’t necessarily imply that Hispanic students individually have diverse friendship groups—some of them do, but others show a strong preference for having mainly Hispanic friends, or for having few.

Also shown in Fig. 2.2 are histograms of the naive preference estimates k_{is}/k_i, which look quite different from the inferred distributions. This discrepancy is expected: as discussed in Section 2.1, the distribution of naive estimates is an unreliable indicator of the true preference distribution.

2.4.2 Word adjacencies

The *Brown corpus* is a widely used data set consisting of samples of written English text compiled by researchers at Brown University in the 1960s [117]. Words in the data set are labeled with their part of speech—noun, adjective, verb, etc. Working from the fiction text contained in the corpus, we create a directed word adjacency network in which nodes represent words (limited to nouns, adjectives, and verbs) and there is a directed edge from word i to word j if word i is followed by word j at any point in the text.

Figure 2.3 shows the inferred distributions of preferences within this network for nouns, verbs, and adjectives to be followed by nouns. For example, since adjectives normally come before nouns in English we would expect adjectives to have a preference for being followed by nouns. And indeed this is what we see—the red curve in the third panel of Fig. 2.3 shows that most adjectives have a high

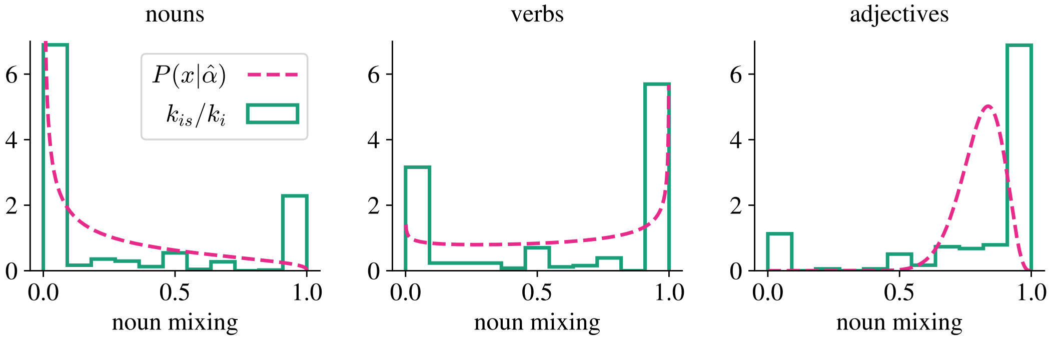


Figure 2.3: “Preferences” of different parts of speech to be followed by nouns. Each word is followed by a noun some proportion of the time, and this proportion is different for different words. For each type of word the histogram (in green) shows the observed distribution of $k_{i,\text{noun}}/k_i$, the naive estimate of noun preference. The red dashed line is the inferred preference distribution from a point estimate of α, found by maximizing Eq. (2.14). The three plots represent the distributions for nouns, verbs, and adjectives from the fiction portion of the Brown corpus of English text [117].

preference for being followed by nouns. Nouns, on the other hand, aren’t usually followed by other nouns, although they can be: the distribution (shown in the first panel of the figure) takes its most likely value around a preference of zero, but is spread across the whole range and there is still a relatively large density around preference 1, which is to say that some nouns strongly prefer to be followed by other nouns. Classic examples are titles such as “Mr.” and “Mrs.,” which are almost always followed by proper nouns. Likewise, although most verbs prefer to be followed by nouns, there are a handful that have a strong preference to be followed by another verb. These are typically auxiliary verbs, such as “has” and “was”, in sentences like “He was sleeping.”

2.5 Discussion

In this chapter we have considered the problem of characterizing mixing patterns in networks. Average mixing patterns have a long history of study and can be quantified using standard methods, but anything beyond the average requires additional machinery for its description. We analyze within-group variation in mixing using a model of individual preferences in networks, showing how to fit the model to data using Bayesian methods. The parameters of the fit have simple interpretations and we use them to define coefficients that quantify the average assortativity and variation of preferences. The method is computationally efficient, with running time growing linearly in the size of the data set, which puts applications to large networks within reach.

We have given applications of our methods to a range of social and information networks. We find that some, though not all, of these networks do display significant within-group variation in their mixing patterns, and that where such variation is present the mixing is not well described by traditional community structure. Even when there is little or no variation in preferences the analysis is still informative, since it implies that preferences are well described solely by which group a node belongs to.

A limitation of our approach is the assumption that the preferences are drawn from a Dirichlet distribution, which rules out multimodal distributions for example. One natural avenue of extension for the approach would be to experiment with other choices of distribution. Instead of a single Dirichlet distribution, for example, one could use a mixture (i.e., a linear combination) of two or more. This would allow us to model more complex behaviors, at the expense of a more complicated fitting procedure. So far, we have also considered only the case in which the label or group membership of every node is known. In the next chapter we generalize our methods to deal with cases in which all or some of the data are unknown.

CHAPTER 3

Inference and Individual Differences

In the previous chapter we considered how mixing patterns may vary at an individual level. We assumed that nodes in a network had assigned categories (such as gender or ethnicity) and we wanted to characterize how nodes with different properties intermingled. Typically, one only has a small quantity of data for each individual and so naive metrics of individual mixing are liable to significantly over-fit data. To circumvent this issue we introduced a flexible model of mixing, but of course, flexible models are also liable to over-fit sparse data. However, our model was carefully constructed so that we could (analytically) integrate out most parameters and efficiently perform Bayesian analyses.

A full statistical model brings further benefits. While we previously assumed the nodes' categories were known, this assumption is not necessary. Rather, we can use the model to estimate the categories. In this chapter we explore this possibility further.

We build on our model for individual mixing and derive a semi-Bayesian method for recovering missing data and performing community detection. We show that allowing for individualized differences in mixing can lead to a significant performance increase over traditional methods that assume all nodes within each group mix identically.

We then provide a theoretical justification for this improvement. Traditional models of mixing undergo a detectability phase transition: unless mixing patterns are sufficiently strong (sufficiently assortative or disassortative), community detection is not possible [83, 118, 119]. We locate a similar phase transition in the individual mixing model. However, in this case detection is only impossible if the network lacks both assortativity *and* variation. Perhaps counter-intuitively, when mixing rates randomly vary between people, we obtain a picture with more signal and less noise.

3.1 Missing data and community detection

The individual mixing model allows each node to mix at its own rate. Each node has an associated group or label, from a discrete set of c options. We use $g_i \in \{1, 2, \ldots, c\}$ to denote the group of node i and say that node i connects to group r at rate x_{ir}. To each node we assign parameters θ_i and ϕ_i to control the expected out- and in- degree, and define Φ_r to be the sum of all ϕ parameters for nodes in group r,

$$\Phi_r = \sum_{i \in r} \phi_i. \tag{3.1}$$

Finally, we assume a Poisson model for network edges,

$$A_{ij} \sim \text{Poisson}\left(\frac{\theta_i \phi_j x_{i g_j}}{\Phi_{g_j}}\right). \tag{3.2}$$

The probability for a network, given the parameters, is

$$P(A|x, g, \theta, \phi) = \prod_i \left(e^{-\theta_i} \prod_j \left(\frac{\theta_i \phi_j x_{i g_j}}{\Phi_{g_j}} \right)^{A_{ij}} \frac{1}{A_{ij}!} \right). \tag{3.3}$$

If, as before, we assume a Dirichlet prior for the x's we can integrate this expression and obtain

$$P(A|\alpha, g, \theta, \phi) = \left(\prod_i e^{-\theta_i} \theta_i^{k_i} \phi_i^{k_i^{\text{in}}} \right) \left(\prod_i \frac{B(\boldsymbol{\alpha}_{g_i} + \boldsymbol{k}_i)}{B(\boldsymbol{\alpha}_{g_i})} \right) \left(\prod_r \Phi_r^{-K_r} \right). \tag{3.4}$$

As before, in this expression $B(\boldsymbol{x})$ is the multivariate Beta function, $\boldsymbol{k}_i$ counts the number of edges from node i to members of each group, i.e. $k_{ir} = \sum_j A_{ij} \delta_{r, g_j}$, and K_r counts the total edges to group r, $K_r = \sum_i k_{ir}$. These expressions all assume the groups are fixed and known, and exactly match the results of Ch. 2.

Now, however, instead of assuming the groups are fixed and known parameters, let's assume the groups are part of the data, drawn from a categorical distribution,

$$g_i \sim \text{Categorical}(\boldsymbol{\pi}). \tag{3.5}$$

From this we get

$$P(A, g|\alpha, \theta, \phi, \pi) = P(A|\alpha, g, \theta, \phi) \prod_r \pi_r^{n_r} \tag{3.6}$$

where n_r is the number of nodes in group r. If some or all of the group assignments are unknown we can use the joint data likelihood of Eq. (3.6): To simultaneously infer parameters and recover the missing data, we maximize this quantity while summing over the values of the unknown data.

Let us denote by g' the set of *known* groups, which we will now assume to be a subset of the set g of all group data. The remaining assignments are unknown. The joint likelihood of the known data given the model parameters is

$$P(A, g'|\alpha, \theta, \phi, \pi) = \sum_{g \notin g'} P(A, g|\alpha, \theta, \phi, \pi), \tag{3.7}$$

where the sum is over all group labels g that are not in g', i.e. all unknown labels. Our procedure to estimate the parameters will be to maximize Eq. (3.7),

$$\begin{aligned} \hat{\alpha}, \hat{\theta}, \hat{\phi}, \hat{\pi} &= \underset{\alpha,\theta,\phi,\pi}{\arg\max}\, P(A, g'|\alpha, \theta, \phi, \pi) \\ &= \underset{\alpha,\theta,\phi,\pi}{\arg\max} \sum_{g \notin g'} P(A, g|\alpha, \theta, \phi, \pi) \end{aligned} \tag{3.8}$$

where $P(A, g|\alpha, \theta, \phi, \pi)$ is given by Eqs. (3.4) and (3.6). The terms in this maximization that involve θ and ϕ do not depend on g and we find

$$\hat{\theta}_i = k_i \tag{3.9}$$

$$\hat{\phi}_i = k_i^{\text{in}}. \tag{3.10}$$

To estimate the remaining parameters, α and π, we need to maximize

$$\hat{\alpha}, \hat{\pi} = \underset{\alpha,\pi}{\arg\max} \sum_{g \notin g'} P(A, g|\alpha, \hat{\theta}, \hat{\phi}, \pi). \tag{3.11}$$

To do this directly, one needs to differentiate the sum, which leads to a complicated implicit equation that is not easy to solve, even numerically. Instead, therefore, we borrow a trick from the statistics toolbox and apply Jensen's inequality, which states that for any distribution of a positive random variable X we have $\ln(E[X]) \geq E[\ln X]$. Applying this inequality to the log of Eq. (3.7) yields

$$\ln \sum_{g \notin g'} P(A, g|\alpha, \theta, \phi, p) \geq \sum_{g \notin g'} q(g) \ln \frac{P(A, g|\alpha, \theta, \phi, p)}{q(g)}, \tag{3.12}$$

where $q(g)$ is any probability distribution over g satisfying $\sum_{g \notin g'} q(g) = 1$. One particularly useful choice of $q(g)$ is

$$q(g) = \frac{P(A, g|\alpha, \theta, \phi, p)}{\sum_{g \notin g'} P(A, g|\alpha, \theta, \phi, p)}, \tag{3.13}$$

which makes the left- and right-hand sides of (3.12) exactly equal, and hence also maximizes the right-hand side of Eq. (3.12) with respect to $q(g)$. A further maximization with respect to the parameters α and π will then give us the answer we seek. To put that another way, a double maximization of the right-hand side with respect to both $q(g)$ and the parameters is equivalent to maximization of the left-hand side, which is the maximization we need to perform.

This leads us to an iterative algorithm for estimating the parameters, known as an expectation–maximization (or EM) algorithm [79, 120, 121], in which we perform the double maximization by simply maximizing alternately over $q(g)$ (using Eq. (3.13)) and then over the parameters, repeating until the numbers converge. In detail the algorithm is as follows:

1. Set $\theta_i = k_i$, $\phi_i = k_i^{\text{in}}$ and make an initial guess $\alpha^{(0)}, \pi^{(0)}$ for the other parameters (for instance the uniform choice $\alpha_{rs}^{(0)} = 1$, $\pi_r^{(0)} = 1/c$). Set $t = 1$.

2. Set
$$q(g) = \frac{P(A, g|\alpha^{(t-1)}, \theta, \phi, \pi^{(t-1)})}{\sum_{g \notin g'} P(A, g|\alpha^{(t-1)}, \theta, \phi, \pi^{(t-1)})}.$$

3. Set
$$\alpha^{(t)}, \pi^{(t)} = \arg\max_{\alpha, \pi} \sum_{g \notin g'} q(g) \ln P(A, g|\alpha^{(t-1)}, \theta, \phi, \pi^{(t-1)}).$$

4. Increase t by 1.

5. Repeat steps 2 to 4 until convergence is achieved.

The most difficult step of the algorithm is step 3, since the sum over groups often has too many terms to be evaluated exactly. Nevertheless, good approximations can be made by Monte Carlo sampling using a standard Metropolis–Hastings algorithm [120, 122].

3.1.1 Monte Carlo algorithm to sample from $q(g)$

The Metropolis–Hastings Monte Carlo algorithm for sampling from $q(g)$ proceeds as follows. Let $U = \{i : g_i = \text{unknown}\}$ be the set of nodes for which we do not know g_i, and initialize g by choosing $g_i \sim \text{Categorical}(\boldsymbol{\pi})$ for each $i \in U$. Then carry out the following steps:

1. Pick an i uniformly at random from U.

2. Propose a new group g'_i for i, uniformly at random from the set of all groups.

3. Accept the move with probability $q(g')/q(g)$, otherwise reject it. If the move is accepted i is moved to group g'_i; if it is rejected i remains in its current group for this Monte Carlo step.

4. Repeat from step 1.

This process continues until a suitable number of independent samples have been drawn from the distribution of group assignments. Like most Monte Carlo algorithms, the "suitable" number of samples is not rigorously defined, but one samples until fluctuations become sufficiently small [122, 123].

To update the estimates for α and π, we need to maximize the expected value of $\ln P(A, g \mid \alpha, \theta, \phi, \pi)$. To do this, we compute the following three averages within the Monte Carlo sample:

$$q_{ir} = \left\langle \delta_{g_i r} \right\rangle, \tag{3.14}$$

$$X_{rk} = \left\langle \sum_i \delta_{g_i r} \delta_{k_i k} \right\rangle, \tag{3.15}$$

$$Y_{rsk} = \left\langle \sum_i \delta_{g_i r} \delta_{k_{is} k} \right\rangle, \tag{3.16}$$

where i is a node label, r and s are group labels, and k is a node out-degree, with values running from 0 to the maximum out-degree for the network. Once we have estimates for these quantities, our estimates for π and α are:

$$\hat{\pi}_r = \frac{1}{n} \sum_i q_{ir}, \tag{3.17}$$

$$\hat{\boldsymbol{\alpha}}_r = \arg\max_{\boldsymbol{\alpha}_r} L_r(\boldsymbol{\alpha}_r), \tag{3.18}$$

where $L_r(\boldsymbol{\alpha}_r)$ represents the terms in the expected log-likelihood that depend on $\boldsymbol{\alpha}_r$:

$$L_r(\boldsymbol{\alpha}_r) = \sum_{s=1}^{c} \sum_{k=0}^{k_{\max}} Y_{rsk} \ln \Gamma(\alpha_{rs} + k) - \sum_{k=0}^{k_{\max}} X_{rk} \ln \Gamma(\alpha_{r0} + k) - \sum_{i=1}^{n} q_{ir} \ln B(\boldsymbol{\alpha}_r). \quad (3.19)$$

The full EM algorithm consists of initializing π and α to uniform values $\pi_r = 1/c$, $\alpha_{rs} = 1$, then iteratively updating q_{ir}, X_{rk}, and Y_{rsk} by Monte-Carlo sampling then fixing $\hat{\pi}_r = \frac{1}{n} \sum_i q_{ir}$ and $\hat{\boldsymbol{\alpha}}_r$ by numerical optimization (e.g. Newton's method), using Eq. (3.19).

In principle, this iterative process should be repeated until L no longer increases. At this point, we should be at a (local) maximum and we will not be able to further improve our estimates. In practice, noise from the Monte Carlo algorithm may cause L not to increase before we reach a maximum. To account for this, we do not terminate the procedure after the first iteration in which L fails to increase. Instead, we proceed until L fails to increase for 5 consecutive iterations.

Once the EM algorithm converges, not only does it provide estimates for the parameters but also the values of the unknown group labels g, since, from Eq. (3.13)

$$q(g) = \frac{P(A, g|\alpha, \theta, \phi, p)}{P(A, g'|\alpha, \theta, \phi, p)} = P(g|A, g', \alpha, \theta, \phi, p). \quad (3.20)$$

In other words, $q(g)$ is precisely the posterior distribution over the unknown group labels. The quantities q_{ir} from the Monte Carlo samples provide the posterior probability that node i is in group r.

Even in cases where all of the group assignments are unknown, the algorithm will still return best estimates of their posterior distribution, effectively functioning as a kind of community detection algorithm [77]. This scenario is the standard use for the stochastic block model [71]. Like the individual mixing model we have been studying, the stochastic block model assumes all nodes in a network are in one of c distinct groups. Nodes connect to each other probabilistically—nodes in group r connect to those in group s at a rate ω_{rs}. In the degree-corrected stochastic block model [71] network edges are also Poisson,

$$A_{ij} \sim \text{Poisson}\left(\theta_i \phi_j \omega_{g_i g_j}\right). \quad (3.21)$$

Comparing this with Eq. (3.2) makes it clear that the stochastic block model is simply the special case of the individual mixing model in which all nodes within any given group have identical preferences. Thus, the stochastic block model is a natural model to compare the individual mixing model against.

In contrast to Ch. 2, the algorithm above is not fully Bayesian. Our estimates of the mixing parameters, α, are maximum likelihood estimates. However, we have still integrated out the individual mixing rates and so we only need to estimate a small number of parameters. Thus we have still avoided the major pitfall of a naive individualized mixing model, which could easily end up with more parameters than data. If there are c groups, we only estimate $c \times c$ parameters. Even on sparse networks, so long as the number of nodes $n \gg c$, this should be doable.

3.1.2 Performance

To test the EM algorithm described above, we performed cross-validation experiments on four real-world and two synthetic network data sets. The data sets we chose contained additional (non-network) data about the properties of nodes, such as their gender or ethnicity. The networks all displayed conventional (dis-)assortative mixing for the properties we considered but they also displayed individual differences in this behavior. In terms of the coefficients developed in Ch. 2, this means assortativity $R \neq 0$ and variation $V > 0$, where

$$R = \frac{\sum_r n_r(\alpha_{rr}/\alpha_{r\Sigma} - K_r/m)}{\sum_r n_r(1 - K_r/m)} \tag{3.22}$$

and

$$V = \sum_r \frac{n_r/n}{\alpha_{r\Sigma} + 1}. \tag{3.23}$$

For the cross-validation experiments we randomly removed group labels for 1/8th of the nodes. Then, we attempted to recover these groups using either the stochastic block model or the individual mixing model. The two algorithms provide (different) posterior probabilities for each node to be in each group. We measured the performance of the two models by computing the posterior probability for nodes to be in their correct group. In other words, we measure performance by

$$F(q) = \sum_{i \in U} q_{i,g_i}. \tag{3.24}$$

If $F(q)$ is close to 1 then the model is very good. Conversely, if $F(q)$ is close to $1/c$ (where c is the number of groups) then we are doing no better than random

Network	grouping	n	R	V	SBM	IM
Lawyers [125]	status	71	-0.13	0.08	0.57	0.73
Word adjacency [117]	part of speech	1089	-0.24	0.35	0.60	0.73
High school [115]	ethnicity	1603	0.55	0.16	0.71	0.72
Facebook [126]	gender	27785	0.10	0.06	0.64	0.72
Synthetic 1	—	4000	0.20	0.23	0.67	0.86
Synthetic 2	—	4000	0.20	0.01	0.67	0.67

Table 3.1: Missing data recovery for six example networks. For each network we list the number of nodes n, the assortativity R, the mixing variance V, and stochastic block model (SBM) and individual mixing (IM) recovery performance. For each of the four real-world networks and two synthetic networks, we randomly removed 1/8th of the data then attempted to recover it. Recovery performance is measured using the mean posterior probability for a node to be in its correct group, i.e. $\sum_i q_{ig_i}$. Across all examples, the individual mixing model performs equally to or better than the conventional block model.

guessing. Each cross-validation experiment was run 100 times for each network, and results were averaged over these runs.

The results of the cross-validation experiments are presented in Table 3.1. We find that the individual mixing model consistently performs as well as or better than the stochastic block model. This demonstrates that the individual mixing model is a promising candidate for real-world inference problems.[1]

The key result of our cross-validation experiments is that the individual mixing model appears to systematically outperform the stochastic block model. In the next section we present a theoretical analysis of why this is the case.

3.2 Theoretical analysis

In the previous section we showed that allowing for individualized mixing rates can improve data recovery. This is not totally surprising. The individual mixing model is substantially more flexible than a conventional block model. By integrating out the individual mixing rates we achieve this flexibility without increasing the number of parameters. A more flexible model with the same number of pa-

[1] This task is different to traditional data imputation tasks [124]. Traditional techniques for imputation exploit correlations between different attributes. For example, we might use someone's age and profession to predict their salary. In contrast, we have not used people's known properties to predict their unknown ones. Instead, we have used the network to recover the unknown properties. If accuracy was of primary importance, a more sophisticated procedure could make use of both network structure and correlations between attributes. However, we do not explore that possibility here.

rameters seems like an obvious improvement. Still, the model introduces another source of random noise—individual preferences are now assumed to be randomly distributed. It is not immediately obvious why *more* randomness helps data recovery.

Intuitively, the variation helps because it strengthens correlations between second neighbors (people who share a mutual friend). If there is little to no assortativity on average, and there is no variation, then each and every individual will have roughly uniform preferences. However, even if there is no assortativity on average, variation implies that many individual nodes do display (dis-)assortative mixing. If a node has strong preferences then all of its neighbors are likely to be similar, and we can use this fact to make classifications (see [106] for a discussion of this phenomenon).

In this section we make this intuition more rigorous. Specifically, we analyze the community detection task and consider under what circumstances we can reliably recover communities. The stochastic block model undergoes a detectability phase transition [83,118,119]. Unless the networks are sufficiently (dis-)assortative, recovering the underlying groups from the network alone is impossible. For weak to moderately assortative networks, it is impossible to even beat random guessing.

Since the stochastic block model is a special case of the individual mixing model—the case in which all preferences within each group are identical—the individual mixing model must undergo the same phase transition when there is no variation in mixing, when $V = 0$. When $V > 0$, as it usually will be, the behavior is less obvious. We will see that when the variance is larger than the reciprocal of degree, $V > 1/\theta$, detection is always possible—there is only one phase. For intermediate values we precisely locate the phase transition using stability analysis of the belief propagation equations.

To simplify the analysis we will only consider the two-block[2] fully symmetric case. We set

$$\theta_i = \phi_i = \theta \tag{3.25}$$

$$\pi_0 = \pi_1 = 1/2 \tag{3.26}$$

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix} \tag{3.27}$$

and we will assume that we know the true values of the parameters. As we will see,

[2] We will label the two groups as group 0 and group 1 (rather than 1 and 2). This will lead to some simplifications to the later equations.

recovering the groups is often not possible even when we know the parameters, and so it is surely impossible when we don't. This is the definition of the undetectable phase. We locate the transition by deriving a *belief propagation* algorithm to recover g and then perform a stability analysis to precisely locate the phase transition.

3.2.1 Derivation of the belief propagation

Assuming all other parameters are known, and consequently dropping them from the notation, the posterior distribution for g is

$$P(g|A) \propto \frac{\prod_i B(\boldsymbol{\alpha}_{g_i} + \boldsymbol{k}_i)/B(\boldsymbol{\alpha}_{g_i})}{\Phi_0^{K_0}\Phi_1^{K_1}}. \tag{3.28}$$

(This is simply Eq. (3.6) with the parameters from Eqs. (3.25), (3.26), and (3.27) inserted.)

Our goal is to compute the posterior distribution for each node,

$$P(g_i = r|A) = \sum_g P(g|A)\delta_{g_i,r}. \tag{3.29}$$

Unfortunately, evaluating this sum directly is practically impossible. However, since Eq. (3.28) is a product of "factors", we can use belief propagation [120,127,128]. Although the equations (derived below) are somewhat involved, we can interpret the procedure as each node repeatedly asking their neighbors "based on what you currently know, what do you think the probability is that I'm in group X?" Once this iterated procedure converges we are left with a consistent set of probabilities that correspond to marginal distributions.

To derive the belief propagation, we associate each node i with a factor

$$f_i(g) = \frac{B(\boldsymbol{\alpha}_{g_i} + \boldsymbol{k}_i)}{B(\boldsymbol{\alpha}_{g_i})} \tag{3.30}$$

and one final factor is defined for the terms involving Φ,

$$\sigma(g) = \Phi_0^{-K_0}\Phi_1^{-K_1} \tag{3.31}$$

where both $\Phi_r = \phi \sum_i \delta_{g_i,r}$ and $K_r = \sum_i \delta_{g_i,r} k_i^{\text{in}}$ have implicit dependence on g. We call these terms factors since Eq. (3.28) is proportional to a product of these terms

$$P(g|A) = \frac{1}{Z}\sigma(g)\prod_i f_i(g) \tag{3.32}$$

where Z is a constant that ensures $\sum_g P(g|A) = 1$.

Each factor f_i is a function that only depends on a small subset of the groups. Specifically, f_i is a function of the group of node i and the groups of all nodes that i connects to. One refers to the functions f_i as *factors*, since they are factors of the relevant distribution, and the g_i as *variables* since they are variables of the factors. Belief propagation computes marginal distributions by "passing messages" between factors and variables (see [127]).

To derive the belief propagation, we need two sets of messages, one from the factors to variables and another from variables to factors. The message equations simply follow a standard formula [127, 128]. In full, the messages from factors to variables are

$$\mu_{i\to j}(g_j) \propto \sum_{g^{(\backslash j)}} f_i(g)\nu_{i\to i}(g_i)\prod_{k\in N_i\backslash j}\nu_{i\to k}(g_k) \tag{3.33}$$

$$\mu_{i\to i}(g_i) \propto \sum_{g^{(\backslash i)}} f_i(g)\prod_{k\in N_i}\nu_{i\to k}(g_k) \tag{3.34}$$

$$\mu_{\sigma\to j}(g_j) \propto \sum_{g^{(\backslash j)}} \sigma(g)\prod_{i\neq j}\nu_{\sigma\to i}(g_i) \tag{3.35}$$

where N_i is the set of nodes that node i connects to

$$N_i = \{j : A_{ij} = 1\} \tag{3.36}$$

and $\sum_{g^{(\backslash i)}}$ is the sum over all groups other than node i's,

$$\sum_{g^{(\backslash i)}} = \sum_{g_1=0}^{1}\cdots\sum_{g_{i-1}=0}^{1}\sum_{g_{i+1}=0}^{1}\cdots\sum_{g_n=0}^{1}. \tag{3.37}$$

The messages for variables to factors are

$$\nu_{i\to j}(g_j) \propto \mu_{\sigma\to j}(g_j)\mu_{j\to j}(g_j) \prod_{k\in M_j\setminus i} \mu_{k\to j}(g_j) \tag{3.38}$$

$$\nu_{i\to i}(g_i) \propto \mu_{\sigma\to i}(g_i) \prod_{k\in M_i} \mu_{k\to i}(g_i) \tag{3.39}$$

$$\nu_{\sigma\to j}(g_j) \propto \mu_{j\to j}(g_j) \prod_{k\in M_j} \mu_{k\to j}(g_j) \tag{3.40}$$

where M_i is the set of nodes that connect to i,

$$M_i = \{j : A_{ji} = 1\}. \tag{3.41}$$

Finally, the one-node marginal distributions are

$$\mu_j(g_j) \propto \mu_{\sigma\to j}(g_j)\mu_{j\to j}(g_j) \prod_{k\in M_j} \mu_{k\to j}(g_j). \tag{3.42}$$

So long as the network is locally tree-like, i.e. so long as it does not contain many short cycles, these equations will be correct [127].

The factors f_i represent the strong coupling between neighboring nodes. Whereas, the factor σ represents the weak coupling between all pairs of nodes. This fully connected weak coupling is an ideal candidate for mean-field approximation [127], and we do exactly that. In a mean-field approximation one replaces quantities by their expected values. The relevant equation is Eq. (3.35),

$$\begin{aligned}\mu_{\sigma\to j}(g_j) &\propto \sum_{g^{(\setminus j)}} \sigma(g) \prod_{i\neq j} \nu_{\sigma\to i}(g_i) \\ &\approx \langle\Phi_0\rangle_{g_j}^{-\langle K_0\rangle_{g_j}} \langle\Phi_1\rangle_{g_j}^{-\langle K_1\rangle_{g_j}}\end{aligned} \tag{3.43}$$

where we have introduced the notation $\langle x\rangle_{g_j}$ for expectation in the distribution $\prod_{i\neq j}\mu_i(g_i)$. Hence,

$$\langle\Phi_r\rangle_{g_j} = \delta_{g_j,r}\phi + \sum_{i\neq j}\mu_i(r)\phi \tag{3.44}$$

$$\langle K_r\rangle_{g_j} = \delta_{g_j,r}k_j^{\text{in}} + \sum_{i\neq j}\mu_i(r)k_i^{\text{in}}. \tag{3.45}$$

Using the mean-field approximation we can combine the belief propagation equations into two sets of equations,

$$\mu_{i\to j}(g_j) \propto \sum_{g^{(\setminus j)}} \Big(f_i(g) \prod_{k\in N_i\setminus j} \nu_{i\to k}(g_k)\Big)\Big(\prod_{k\in M_i} \mu_{k\to i}(g_i)\Big)\sigma_i(g_i) \tag{3.46}$$

$$\nu_{i\to j}(g_j) \propto \sum_{g^{(\setminus j)}} \Big(f_j(g) \prod_{k\in N_j} \nu_{j\to k}(g_k)\Big)\Big(\prod_{k\in M_j\setminus i} \mu_{k\to j}(g_j)\Big)\sigma_j(g_j), \tag{3.47}$$

and the equations are only defined along edges of the network, $i \to j$. The marginal distribution for a single node is

$$\mu_j(g_j) \propto \sum_{g^{(\setminus j)}} \Big(f_j(g) \prod_{k\in N_j} \nu_{j\to k}(g_k)\Big)\Big(\prod_{k\in M_j} \mu_{k\to j}(g_j)\Big)\sigma_j(g_j). \tag{3.48}$$

For each equation, most of the sums in $\sum_{g^{(\setminus j)}} = \sum_{g_1=0}^{1}\cdots\sum_{g_{j-1}=0}^{1}\sum_{g_{j+1}=0}^{1}\cdots\sum_{g_n=0}^{1}$ are trivial. For example, in Eq. (3.46) the only values of g_k that appear in the summand are for nodes k that are successors of node i. If node i has out-degree k_i then in principle there are 2^{k_i} terms to sum over. A trick using discrete Fourier transforms can compute this sum extremely efficiently.

3.2.2 Fourier transform and convolutions

In Eqs. (3.46) and (3.47) we average f_i and f_j respectively over all group assignments for the nodes in the neighborhood of i and j. The factor $f_i(g)$ only depends on g_i, and the number of successor nodes of i that are in group 1. If $g_i = 0$ then f_i is

$$f_i(g|g_i = 0) = \frac{B(\alpha + k_i - k_{i,1},\, \beta + k_{i,1})}{B(\alpha,\beta)} \tag{3.49}$$

for $g_i = 1$ one simply makes the replacement $\alpha \leftrightarrow \beta$.

We can thus write f_i as a function of g_i and an integer d, the number of successor nodes of i in group 1,

$$f_i(d, g_i) = \delta_{g_i,0}\frac{B(\alpha + k_i - d,\, \beta + d)}{B(\alpha,\beta)} + \delta_{g_i,1}\frac{B(\beta + k_i - d,\, \alpha + d)}{B(\alpha,\beta)}. \tag{3.50}$$

Re-writing Eq. (3.46) leads to

$$\mu_{i\to j}(g_j) \propto \sum_{g_i=0}^{1}\Bigg[\bigg(\sum_{d=0}^{k_i-1} f_i(d+\delta_{1,g_j},g_i)\sum_{g_{l_1}=0}^{1}\cdots\sum_{g_{l_{k_i}}=0}^{1}\delta\big(\textstyle\sum_{l\in N_i\backslash j} g_l, d\big)\displaystyle\prod_{l\in N_i\backslash j}\nu_{i\to l}(g_l)\bigg) \\ \times\bigg(\prod_{k\in M_i}\mu_{k\to i}(g_i)\bigg)\sigma_i(g_i)\Bigg]$$

$$= \sum_{g_i=0}^{1}\bigg(\sum_{d=0}^{k_i-1} f_i(d+\delta_{1,g_j},g_i)P_{\mu_{i\to j}}(d)\bigg)\bigg(\prod_{k\in M_i}\mu_{k\to i}(g_i)\bigg)\sigma_i(g_i) \qquad (3.51)$$

where $P_{\mu_{i\to j}}(d)$ is the convolution

$$P_{\mu_{i\to j}}(d) = \sum_{g_{l_1}=0}^{1}\cdots\sum_{g_{l_{k_i}}=0}^{1}\delta\big(\textstyle\sum_{l\in N_i\backslash j} g_l, d\big)\displaystyle\prod_{l\in N_i\backslash j}\nu_{i\to l}(g_l), \qquad (3.52)$$

and from the convolution theorem

$$P_{\mu_{i\to j}}(d) = \mathcal{F}^{-1}\Bigg[\prod_{l\in N_i\backslash j}\mathcal{F}\big[\nu_{i\to l}\big]\Bigg] \qquad (3.53)$$

where $\mathcal{F}$ is the discrete Fourier transform [129]. This can be computed efficiently using fast Fourier transform algorithms [130].

An exactly analogous argument leads to

$$\nu_{i\to j}(g_j) \propto \left(\sum_{d=0}^{k_j} f_j(d,g_j)P_{\nu_{i\to j}}(d)\right)\left(\prod_{k\in M_j\backslash i}\mu_{k\to j}(g_j)\right)\sigma_j(g_j) \qquad (3.54)$$

with

$$P_{\nu_{i\to j}}(d) = \mathcal{F}^{-1}\Bigg[\prod_{l\in N_j}\mathcal{F}\big[\nu_{j\to l}\big]\Bigg]. \qquad (3.55)$$

3.2.3 Verification of the belief propagation

To verify the correctness of this belief propagation we ran tests on synthetic networks. We found excellent agreement between the estimates from belief propagation and those using the Monte Carlo algorithm of Sec. 3.1. An example is shown in Fig. 3.1, which shows a scatter plot of the one-node marginal probabilities estimated using the two algorithms. On these synthetic networks with two groups the

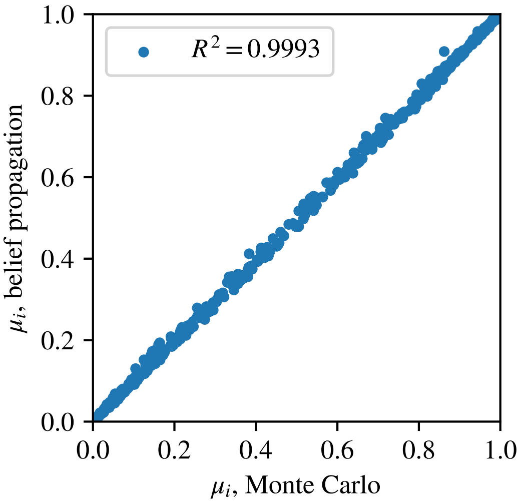


Figure 3.1: Test of the belief propagation algorithm of Sec. 3.2.1. A synthetic network was created with $n = 4,000$, $\alpha = \beta = 1$, and $\theta = \phi = 5$. We show estimates of the one-node marginal posterior probability of being in group 1. Estimates from the belief propagation algorithm are plotted against estimates from the Monte Carlo algorithm. Disagreement is small and the R^2 is almost 1.

belief propagation is considerably faster than the Monte Carlo algorithm. However, the belief propagation has a potentially fatal short-coming that leads us not to recommend its use for real-world data: the equations assume a locally tree-like network [127].[3] While this assumption is true for networks that are generated from the model it is frequently violated in the real-world, where networks often have large number of short cycles such as triangles [1]. In contrast, the Monte Carlo algorithm is guaranteed to sample correctly from the posterior distribution (once it equilibrates).

Our reason for deriving this belief propagation, however, was not to develop a new efficient algorithm. Rather, these equations lead to theoretical insight of the model—we can now perform stability analysis to locate the phase transition.

3.2.4 Symmetry breaking phase transition

In the fully symmetric case that we have been considering, the true marginals are (trivially) symmetric. For each group assignment g there is an opposite assignment

[3] In Ch. 5 we will examine this short-coming of message passing algorithms in depth.

$\neg g$,

$$\neg g_i = \begin{cases} 0 & \text{if } g_i = 1 \\ 1 & \text{if } g_i = 0 \end{cases} \tag{3.56}$$

and $P(g|A) = P(\neg g|A)$. This codifies the fact that the group labels themselves (i.e. "group 0" or "group 1") are arbitrary names. A simple consequence of this symmetry is that the true one-node marginal distribution for each node is simply

$$P(g_i = 0|A) = P(g_i = 1|A) = \frac{1}{2}. \tag{3.57}$$

This, however, is not the answer that we want. It is a trivial answer, true for all networks, and says nothing about any given network's structure.

One method to deal with this symmetry, and thus to arrive at an interesting answer, is to arbitrarily fix the label of one node. If we insist that $g_1 = 1$, i.e. that "group 1" will always refer to the group containing node 1, the symmetry is broken and we might find interesting structure. Usually, however, this purposeful breaking of the symmetry isn't necessary. Rather, it will occur quite naturally.

For example, in a Metropolis-Hastings style algorithm that samples group assignments g from $P(g|A)$, these samples will be clustered in some basin of attraction around some assignment we can call g'. An exactly equivalent basin also exists around the opposite assignment, $\neg g'$, but moving between these two basins will take an extremely long time. In other words, the Monte Carlo algorithm will automatically break the symmetry.[4]

Spontaneous symmetry breaking also manifests in the belief propagation. By the same symmetry arguments as before, the belief propagation equations have a trivial fixed point at

$$\nu_{i\to j}(0) = \nu_{i\to j}(1) = 1/2 \tag{3.58}$$

$$\mu_{i\to j}(0) = \mu_{i\to j}(1) = 1/2. \tag{3.59}$$

However, this fixed point may or may not be stable to small perturbations. If it is unstable then the belief propagation will automatically break the symmetry between groups 0 and 1 and the equations will converge to a non-trivial fixed point (assuming they converge at all).

So, to precisely locate the phase transition we should find the point at which the

[4] This is analogous to spontaneous symmetry breaking in the Ising model [123, 131].

trivial fixed point of the belief propagation switches from being stable to unstable. To this end, let's assume small perturbations to the fixed point, viz.

$$\mu_{i\to j}(g_j = 1) = \frac{1}{2} + \epsilon_{i\to j} \tag{3.60}$$

$$\nu_{i\to j}(g_j = 1) = \frac{1}{2} + \delta_{i\to j}. \tag{3.61}$$

How will these perturbations change after one iteration? Assuming that ϵ and δ quantities are arbitrarily small the updated equations will read

$$\mu_{i\to j}(g_j) = \sum_{g^{(\setminus j)}} \Big(\frac{1}{2} + \sum_{k\in M_i} \epsilon_{k\to i}(g_i) + \sum_{k\in N_i\setminus j} \delta_{i\to k}(g_k)\Big) f_i(g), \tag{3.62}$$

$$\nu_{i\to j}(g_j) = \sum_{g^{(\setminus j)}} \Big(\frac{1}{2} + \sum_{k\in M_j\setminus i} \epsilon_{k\to j}(g_j) + \sum_{k\in N_j} \delta_{j\to k}(g_k)\Big) f_j(g). \tag{3.63}$$

After some algebra, Eqs. (3.62) and (3.63) lead to

$$\epsilon_{i\to j} = \left(\frac{\alpha-\beta}{\alpha+\beta}\right) \sum_{k\in M_i} \epsilon_{k\to i} + \left(\frac{\alpha(\alpha+1)+\beta(\beta+1)-2\alpha\beta}{(\alpha+\beta)(\alpha+\beta+1)}\right) \sum_{k\in N_i\setminus j} \delta_{i\to k} \tag{3.64}$$

$$\delta_{i\to j} = \sum_{k\in M_j\setminus i} \epsilon_{k\to j} + \left(\frac{\alpha-\beta}{\alpha+\beta}\right) \sum_{k\in N_j} \delta_{j\to k}. \tag{3.65}$$

The quantities involving α and β are closely related to assortativity and variation of mixing coefficients, R and V,

$$\frac{\alpha-\beta}{\alpha+\beta} = R \tag{3.66}$$

$$\frac{\alpha(\alpha+1)+\beta(\beta+1)-2\alpha\beta}{(\alpha+\beta)(\alpha+\beta+1)} = R^2(1-V) + V. \tag{3.67}$$

These equations tell us how small perturbations affect the messages after one iteration. To analyze the stability of the whole system of equations, we should consider how these perturbations propagate through multiple iterations.

To this end, consider the l neighborhood around a randomly chosen node, that is, the subgraph induced by all nodes a distance l or less from our random node. Such neighborhoods will be loop free up to $l \sim O(\log n)$, and so in the limit of large network size we will have a tree. We will imagine perturbing the incoming messages on the leaves of this tree, and calculate the effect on the root.

The root node will, on average, have θ out-edges and θ in-edges. Each one of these edges will lead to, on average, a further θ out-edges and θ in-edges. Ignoring the directions for now, we have a Poisson branching process [132] with an average of 2θ offspring for each node. On average there will be $(2\theta)^l$ nodes in the lth generation.

We will displace the messages on the leaves by a small, random, mean-zero displacement,

$$\begin{pmatrix}\Delta\mu_i\\ \Delta\nu_i\end{pmatrix} = \begin{pmatrix}\epsilon_i\\ \delta_i\end{pmatrix}. \tag{3.68}$$

How these perturbations propagate to the root node will depend on the direction of the edges. Each edge in the tree will point up with probability 1/2 or down with probability 1/2. Any specific order of directions from a leaf to the root (e.g. $\uparrow,\uparrow,\downarrow,\dots,\uparrow$) will occur with probability $1/2^l$. The perturbation will grow (or shrink) dependent on this order, and the transitions are

$$\uparrow,\uparrow \quad : \quad \begin{pmatrix}\epsilon\\ \delta\end{pmatrix} \to \begin{pmatrix}R\epsilon\\ 0\end{pmatrix} \tag{3.69}$$

$$\uparrow,\downarrow \quad : \quad \begin{pmatrix}\epsilon\\ \delta\end{pmatrix} \to \begin{pmatrix}0\\ \epsilon\end{pmatrix} \tag{3.70}$$

$$\downarrow,\uparrow \quad : \quad \begin{pmatrix}\epsilon\\ \delta\end{pmatrix} \to \begin{pmatrix}(R^2(1-V)+V)\delta\\ 0\end{pmatrix} \tag{3.71}$$

$$\downarrow,\downarrow \quad : \quad \begin{pmatrix}\epsilon\\ \delta\end{pmatrix} \to \begin{pmatrix}0\\ R\delta\end{pmatrix}. \tag{3.72}$$

An example illustrating these rules is shown in Fig. 3.2.

The transitions can be summarized with the transition matrix

$$\boldsymbol{T}_1 = \begin{pmatrix}R & R^2(1-V)+V\\ 1 & R\end{pmatrix}. \tag{3.73}$$

The average aggregate effect on the root due the the leaves' perturbations is

$$\langle\boldsymbol{\epsilon}_{\text{root}}\rangle = \left\langle \sum_i \frac{1}{2^l}\boldsymbol{T}_\mathbf{1}^l \begin{pmatrix}\epsilon_i\\ \delta_i\end{pmatrix}\right\rangle = \theta^l\langle\boldsymbol{T}_\mathbf{1}^l\boldsymbol{\epsilon}_i\rangle = \begin{pmatrix}0\\ 0\end{pmatrix} \tag{3.74}$$

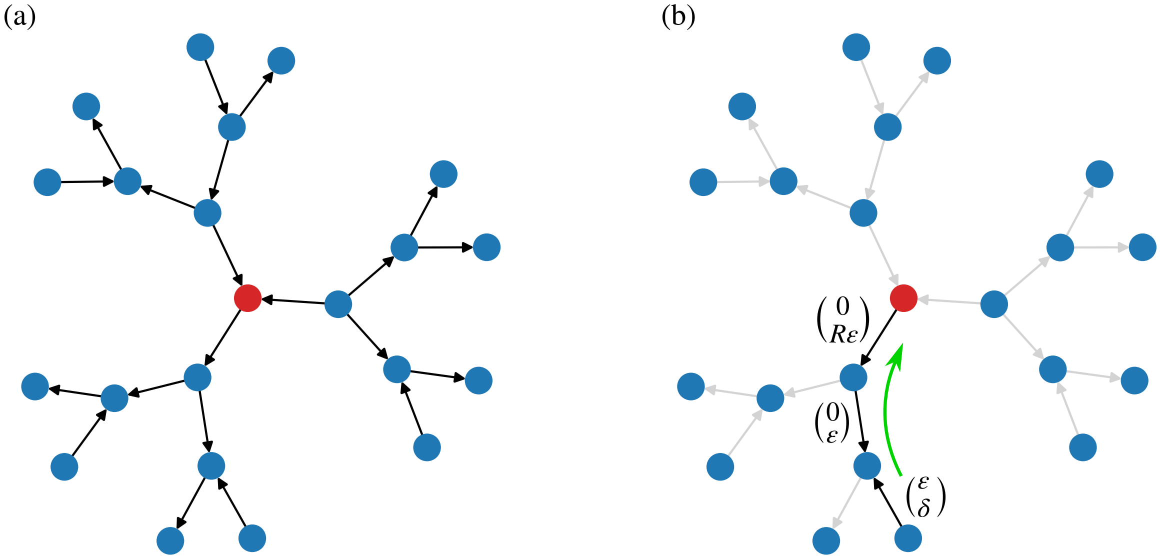


Figure 3.2: Propagation of a small message perturbation. An example neighborhood shown to $l = 3$. In (a) we draw the neighborhood of a node up to distance 3. Arrows mark the directions of edges in the directed network. In (b) we consider a small perturbation to one of the leaf messages. We show how a small $\binom{\epsilon}{\delta}$ perturbation will propagate towards the root, in accordance with the transition rules of Eqs. (3.69–3.72). Once it reaches the root the pertubation is $\binom{0}{R\epsilon}$. Note, for this example the edges are in the order ↑, ↓, ↓. A similar thought experiment is used to locate the phase transition. To do this, we imagine perturbing *all* of the leaves and compute the aggregate effect on the root node.

where the sum is over all $(2\theta)^l$ leaves, indexed by i. In expectation the effect on the root is zero because $\langle \epsilon_i \rangle = \langle \delta_i \rangle = 0$. However, the variance of the perturbations need not vanish.

To compute the variance, $\langle \boldsymbol{\epsilon}^2_{\text{root}} \rangle$, we again sum over all contributions from each leaf, and all possible configurations for the edge directions. This time, however, we pick up two factors of each coefficient on each transition (two because we square the result), and the transition matrix is

$$\boldsymbol{T}_2 = \begin{pmatrix} R^2 & \left(R^2(1-V)+V\right)^2 \\ 1 & R^2 \end{pmatrix}. \tag{3.75}$$

The final aggregate effect on the root, summed over all leaves is

$$\langle \boldsymbol{\epsilon}^2_{\text{root}} \rangle = \theta^l \left\langle \boldsymbol{\epsilon}_i^T \boldsymbol{T}_2^{\,l} \boldsymbol{\epsilon}_i \right\rangle \approx \theta^l \lambda^l \langle \epsilon_i^2 + \delta_i^2 \rangle \tag{3.76}$$

where λ is the largest eigenvalue of $\boldsymbol{T}_2$

$$\lambda = 2R^2 + V(1 - R^2). \tag{3.77}$$

Thus, perturbations grow in magnitude if

$$2R^2 + V(1 - R^2) > 1/\theta. \tag{3.78}$$

In Fig. 3.3 we verify that Eq. (3.78) correctly locates the phase transition. Note, the stochastic block model is the special case in which all nodes have identical preferences and so $V = 0$. In this case Eq. (3.78) becomes $R^2 > 1/(2\theta)$, in agreement with standard results [83].

What is the implication of Eq. (3.78)? The key point is that individual differences—non-zero V—have the effect of shifting the detectability transition down. In the stochastic block model one requires sufficiently large R, sufficiently strong (dis-)assortative mixing, in order for recovery to be possible. Once we allow for individual differences the required strength of assortative patterns decreases. In fact even when $R = 0$, when the network is completely non-assortative at the population level, recovery is still possible so long as $V > 1/\theta$, which is not a particularly stringent requirement.

For many cases of interest neither R or V will be zero [98] and the non-zero value of V makes detection easier than it would otherwise be. This comports with the observations in Table 3.1 that recovery in the individual mixing model is superior to conventional block models.

3.3 Discussion

In this chapter we have derived algorithms for missing data recovery and community detection in the presence of individualized mixing rates. When networks are only partly labeled, or completely unlabeled, the individual mixing model can be used to recover unknown node characteristics and we present an expectation–maximization algorithm for doing so. We have demonstrated the effectiveness of our methods with applications to a selection of networks, including real-world examples and synthetically generated benchmarks. We have shown that our methods work well even when average mixing patterns are weak, so long as there is variation between individuals.

We derived a belief propagation and used stability analysis to locate a phase transition in community detection. In contrast to results from the stochastic block model, we find that community detection is possible at zero assortativity, so long as there is sufficient variation. Since the assumption that all members of a group have exactly identical preferences is dubious in the real-world, the stochastic block model gives an overly pessimistic view of detectability. Instead, we can actually harness individual differences to help classification—allowing for more randomness in individual preferences actually increases the overall signal. Intuitively, the reason for this is that edges are often strongly correlated and knowledge of one edge can provide significant clues about the others. In other words, it is possible to recover more information than conventional independence assumptions allow—such assumption are not benign technicalities.

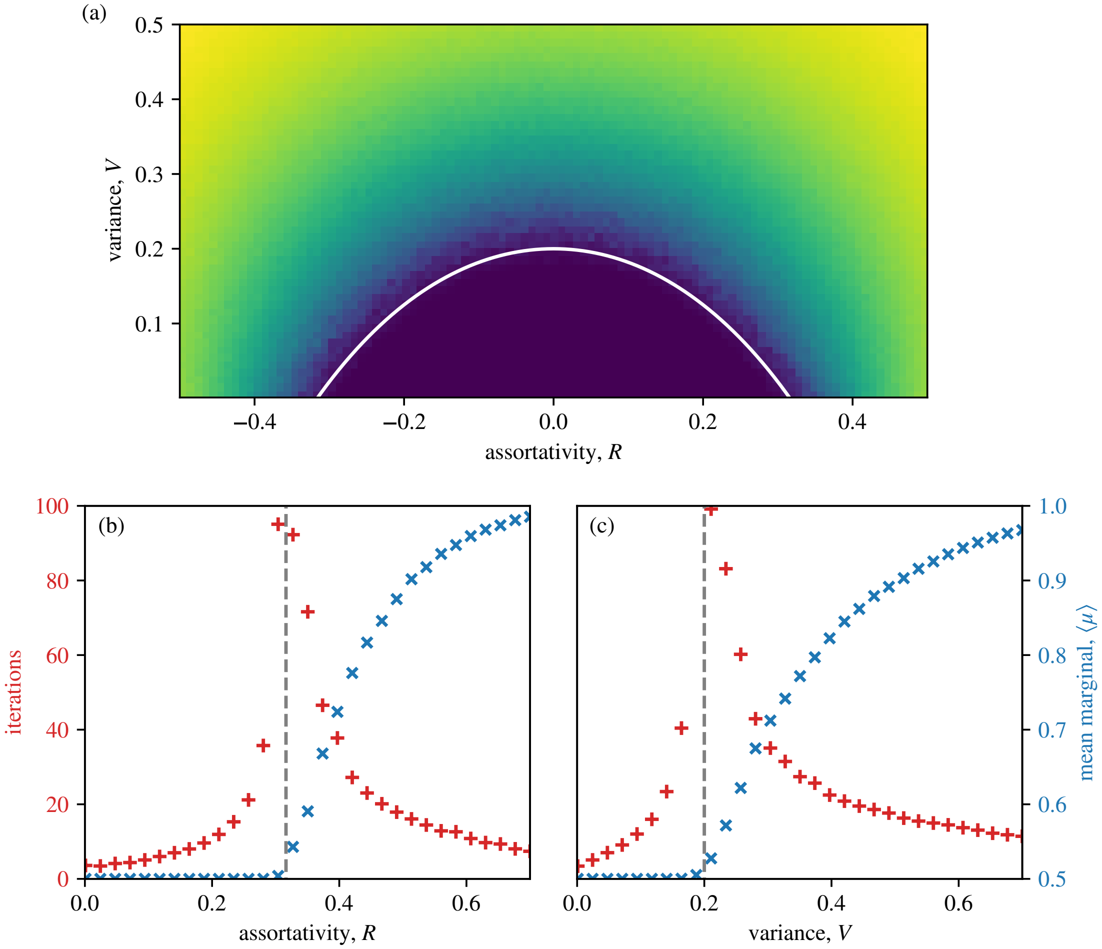


Figure 3.3: Performance in the group label recovery task with synthetic data. In (a) we show the result of simulations for different values of assortativity R and variance V. Color (brightness) at each point in the RV-plane corresponds to average recoverability. Bright regions indicate high performance, dark regions poor performance. Performance is measured using $\frac{1}{n}\sum_i \mu_i(g_i)$, the mean posterior probability for nodes to be in their true group. The white hyperbola-like shape indicates the theoretical location of the phase transition, given by Eq. (3.78). Inside the shape recovery is not possible, outside it is often straightforward. In (b) we plot recovery performance and convergence time (number of iterations until convergence) for different values of R, at $V = 0.001$. In this regime (as $V \to 0$) the model is equivalent to the conventional stochastic block model. At the phase transition (dashed vertical line), the convergence time of the belief propagation diverges. In (c) we again plot recovery performance and convergence time, this time with $R = 0$ and varying V. In this regime there is no aggregate community structure whatsoever yet we can still recover the groups so long as $V\theta > 1$. Sample networks contain 2000 nodes and we set the degree parameters $\theta = \phi = 5$. In all cases the theoretical predictions match the simulated results excellently.

CHAPTER 4

Edge Correlation and Thresholding

This chapter is adapted from the published results of G. T. Cantwell, Y. Liu, B. F. Maier, A. C. Schwarze, C. A. Serván, J. Snyder, and G. St-Onge, Thresholding normally distributed data creates complex networks. *Physical Review E* **101**(6), 062302 (2020) [99]. All authors made contributions to this project. G.T.C. and B.F.M. wrote the published manuscript.

Real-world networks tend to be *complex*—structured, but lacking any obvious repeating pattern. Common features of such complex networks are: heavy-tailed degree distributions, large numbers of triangles, large connected components, and short paths between nodes. These properties are observed in disparate networks, such as friendship networks, metabolic networks, and the World Wide Web [1]. Despite commonalities, it is clear that the details of these systems differ considerably. Quite naturally one may wonder: Why do different processes lead to the same network properties? Researchers have speculated that there may be universal network mechanisms or some form of *universality* at play.[1]

In this chapter we will explore the effect of edge correlations on network structure. Our key finding is that some of the "universal" properties of complex networks arise when edges are correlated. However, to explore the effect of correlation, by and of itself, we need an appropriate mathematical laboratory. The correct framework for studying edge correlations in networks is not immediately obvious.

Simple random graph models, such as Erdős and Rényi's [64, 65], assume edges are independent (i.e. uncorrelated), and so are clearly inappropriate for studying

[1] In this context, universality is presumably supposed to invoke the concept from statistical physics. Near a critical point, the exact details of a system are often irrelevant to its behavior. Instead, the behavior depends only on the (i) symmetry; (ii) dimensionality; (iii) nature of critical point [131]. In network science, however, the use of this term generally seems vague [133].

the effect of correlation. More complex models, however, generally introduce further structure and assumptions. As a result, most existing models—even those that imply edge correlations—are inappropriate for studying correlations. For example: attachment or copying models assert specific mechanisms [40, 72, 134]; configuration models assert degree heterogeneity by fiat [39], and thus can't explain it; stochastic block models assume community structure [69];[2] Watts-Strogatz networks assume an underlying regular lattice [41], and so forth.

The two-star model [92, 95] is a seemingly promising option. Derived from the principle of maximum entropy [135], the two-star model rigorously introduces precisely one effect—the effect of edge correlations—into otherwise maximally random graphs. To derive the model, one assumes there may be non-zero correlations between edges that "touch". That is to say, the existence of edge (i, j) might be correlated with the existence of edge (j, k), since the edges touch at node j. The probability of any particular adjacency matrix, $P(A)$, is determined by maximizing the entropy of this distribution, subject to density and correlations constraints. The result of such a procedure is

$$P(A) = \frac{1}{Z} \exp\left(\beta \sum_{i \neq j} A_{ij} + \gamma \sum_{i \neq j \neq k} A_{ij} A_{jk}\right) \tag{4.1}$$

where Z is a constant that ensures $\sum_A P(A) = 1$. The parameter β controls the network density by encouraging (or discouraging) edges. Whereas, the parameter γ controls the strength of correlations between edges that touch.

While the two-star model of Eq. (4.1) appears to be theoretically well justified, further analysis demonstrates that it too is inappropriate for studying correlation. The problem is that if one attempts to introduce even moderate correlations, one ends up with a "degenerate" distribution: almost all the the probability mass in $P(A)$ is clustered around either the complete graph or the empty graph. In other words, our attempt to introduce moderate correlations leads to a run-away process that says either all the edges must be present, or none of them [95]. Attempts have been made to fix this by introducing further constraints (involving larger structures, e.g. in Ref. [136]) but these are ultimately ad hoc and introduce further parameters.

In contrast to previous studies, we will take a step back and consider how network data sets are created. Instead of modeling the underlying structure as

[2] Arguably the stochastic block model doesn't even correlate edges. Community assignments are fixed before the network is generated, and conditional on community assignment all edges are independent. The model essentially corresponds to multiple Erdős-Rényi style random graphs.

a discrete simple graph, we consider continuous relational data—all nodes may interact but with differing strengths. A network is created from this continuous data by *thresholding* (or *dichotomizing*)—strong interactions correspond to edges, weak interactions to non-edges. Our basic model for the underlying continuous data will be very simple: data is assumed to be normally distributed. Despite this simplicity, after thresholding we find the data indeed display properties associated with complex networks.

Our decision to study thresholded real-valued data is not without justification. In many real-world settings, interactions are indeed indicated by real-valued data and so creating a simple network often requires a process of thresholding, which may take several forms [137–146]. The most obvious case is when a continuous valued data set is explicitly thresholded by deciding what level of interaction is sufficiently strong to count as an edge in the network.

A more subtle case of thresholding is when it occurs due to experimental limitation: interactions that exist but are very weak or rare may not be observed. Even for binary valued data sets, the sampling method may hide an implicit thresholding mechanism. For example, one commonly uses a combination of a yeast two-hybrid screen and biochemical assays to detect and verify edges in protein-protein interaction networks. These methods typically do not detect weak protein-protein interactions [147] and are thus equivalent to applying a threshold on the edge strength in protein-protein interaction networks. Likewise, most everyday interactions between people are presumably not strong enough to constitute friendship. At what point does an acquaintance cross over to the category of friend? When people list their friends, in a survey for instance, they will implicitly apply some criteria to filter the friends from the acquaintances.

Our basic model for relational data will be derived from three assumptions:

1. nodes are statistically identical;
2. correlations are local;
3. underlying relational data are normally distributed.

All three of these assumptions—which are no doubt violated in the real-world—are quite natural for a null model. Assumption 1, that all nodes are identical, severely constrains what correlation structures are admissible. In fact, only two free parameters remain in the covariance matrix once this assumption is made: a local correlation strength between edges that touch, and a global correlation strength between edges that do not. Assumption 2 sets the second of these to zero—edges that do not touch are uncorrelated. Our remaining freedom is to pick a distribution that is consistent with the required correlation matrix. The most obvious and simple choice is assumption 3, the multivariate normal (Gaussian) distribution.

The thresholding procedure will also be very simple: any of the relational data that falls above some threshold, t, will be said to constitute an edge in the network, and any that falls below will not. The threshold value t is a parameter of the model.

Our network ensemble on n nodes is thus defined by two parameters: the threshold, t, and a local correlation coefficient, ρ. Despite the simplicity of the model—the underlying relational data are normally distributed—we nonetheless find a number of the behaviors typically observed in complex networks, such as heavy-tailed degree distributions, short average path lengths, and large numbers of triangles. These properties are thus a natural consequence of correlated relational data. The networks do not, however, possess non-vanishing clustering or community structure in the large n limit and so cannot account for this observation in real-world data sets.

This chapter has two main parts. In Sec. 4.1 we define and justify the network model and in Sec. 4.2 we study the properties of the ensemble. We look at the density of edges, triangles and clustering, the degree distributions, shortest path lengths, and the giant component.

4.1 Model specification

4.1.1 Thresholding locally correlated data

A network can be represented by its adjacency matrix, $\boldsymbol{A}$, where $A_{ij} = 1$ if node i and j are connected and $A_{ij} = 0$ otherwise. We consider networks created by thresholding underlying relational data, $\boldsymbol{X}$, adding an edge between i and j if

$$X_{ij} \geq t \tag{4.2}$$

(see Fig. 4.1a). To fully specify the model we need to pick a distribution for $\boldsymbol{X}$. Assuming that all nodes are statistically identical—exchangeable in the parlance of statistics—constrains our choice of distribution.

If nodes are identical then the marginal distribution for X_{ij} must be the same

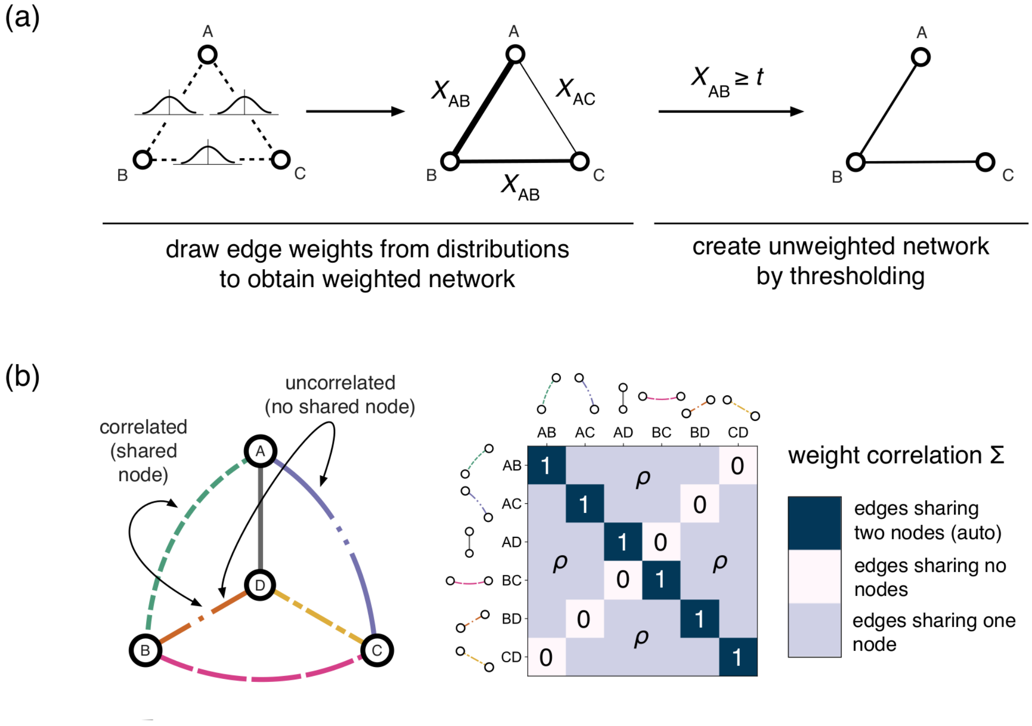


Figure 4.1: Thresholding relational data to obtain networks. Panel (a) shows a general procedure to obtain unweighted networks from edge weights. Each edge weight is hypothesized to have been drawn from a specific distribution, generating an undirected weighted network. An unweighted network is then produced by assigning an edge whenever an edge weight X_{ij} is greater than a threshold t. In panel (b) we show how edge weights are correlated in the model of Sec. 4.1 by covariance matrix $\mathbf{\Sigma}$ (Eq. (4.5)). Edge weights for edges which connect through a node have covariance $\mathrm{Cov}[X_{ij}X_{ik}] = \rho$, while edge weights not connected by a node have zero covariance.

for all (distinct) pairs i and j. Further, by a linear transform we can always set $E[X_{ij}] = 0$ and $\text{Var}[X_{ij}] = 1$. So long as the appropriate transformation is made to t, this shift will have no effect on the thresholded network. For this reason we will always assume X_{ij} has mean 0 and variance 1. Exchangeability puts further constraints on the covariance matrix, whose entries can take only three values. For i, j, k, l all distinct, these are

$$\begin{aligned} \text{Var}[X_{ij}] &= \Sigma_{(i,j),(i,j)} = 1, \\ \text{Cov}[X_{ij}, X_{ik}] &= \Sigma_{(i,j),(i,k)} = \rho, \\ \text{Cov}[X_{ij}, X_{kl}] &= \Sigma_{(i,j),(k,l)} = \gamma, \end{aligned} \tag{4.3}$$

where $\text{Cov}[X, Y]$ denotes covariance. We will assume that $\gamma = 0$ since this quantifies the correlation between two edges that do not share a node, i.e. two edges that do not touch (see Fig. 4.1b). This leaves us with two free parameters, t and ρ. The remaining task is to pick a distribution with the required covariance matrix, $\mathbf{\Sigma}$.

In principle any distribution could be used, but the obvious choice is a multivariate normal distribution. In standard notation a multivariate normal distribution (MVN) is denoted $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$. The probability density function of an N-dimensional MVN is

$$P\left(\boldsymbol{x}\right) = \frac{e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}}{\sqrt{(2\pi)^N \det(\mathbf{\Sigma})}}. \tag{4.4}$$

The normal distribution has many points in its favor. Famously it arises in the central limit theorem, which makes it a plausible model for many random processes. If the relational data $\boldsymbol{X}$ arises due to the aggregation of many independent processes then the central limit theorem implies $\boldsymbol{X}$ will be multivariate normally distributed. Further, the normal distribution is the maximum entropy distribution with the required covariance matrix, Eq. (4.3), and so could be justified as the "least informative distribution"—the model that makes the fewest extra assumptions beyond the correlation structure. We can also appeal to simple pragmatism: the multivariate normal distribution is well-studied and has convenient mathematical properties.

A concise statement of the model is as follows: given the freely chosen parameters $t \in \mathbb{R}$, $\rho \in \left[0, \frac{1}{2}\right]$, and the number of nodes n, let $\mathbf{\Sigma}$ be the matrix

$$\begin{aligned}\Sigma_{(i,j),(i,j)} &= 1,\\ \Sigma_{(i,j),(i,k)} &= \rho,\\ \Sigma_{(i,j),(k,l)} &= 0.\end{aligned} \tag{4.5}$$

Then draw a random variable $\boldsymbol{X}$ with

$$\boldsymbol{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \tag{4.6}$$

and create the network by thresholding $\boldsymbol{X}$,

$$A_{ij} = \begin{cases} 1 & \text{if } X_{ij} \geq t, \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

Note, we constrain $0 \leq \rho \leq \frac{1}{2}$ so that $\mathbf{\Sigma}$ is positive semi-definite.[3]

Even if we have good reason to believe that the marginal distributions for X_{ij} are not normal, the model may still be applicable. Consider an arbitrary cumulative distribution function, $F(x)$, and let $\Phi(x)$ denote the standard normal cumulative distribution function. If we sample $\boldsymbol{X}$ from a multivariate normal distribution, and then apply the function $F^{-1}(\Phi(x))$ to each X_{ij} we will have transformed the edge weights to the arbitrary distribution F. So long as we apply the same transformation to t, however, the resulting network after thresholding will be identical.

The upshot is that our model can be adapted for any marginal distribution, and no network properties change—the assumption that the edge weights have normally distributed marginals is of no real consequence. What *is* important is the assumption that there is some transformation of the data such that the *joint* distribution is multivariate normal. While this assumption is a limitation, the above procedure is actually one of the standard methods for creating multivariate distributions with arbitrary marginals.

[3] To see why $\rho > \frac{1}{2}$ is problematic, consider the marginal distribution for four edges, say $X_{ij}, X_{jk}, X_{kl}, X_{il}$. A simple calculation shows that the covariance matrix has a negative eigenvalue for $\rho > \frac{1}{2}$. Similarly, since Var $\left[\Sigma_j X_{ij}\right]$ must be greater than zero, ρ must be greater than $-1/(n-2)$ and so negative correlations can be vanishingly weak at most.

4.1.2 Sampling from the model

We now describe a simple algorithm to sample from the model. This algorithm also provides an intuitive model interpretation.

Let Z_i be n i.i.d. variables, $\mathcal{N}(0,1)$. Let Y_{ij} be $\binom{n}{2}$ i.i.d. variables, $\mathcal{N}(0,1)$. Then let

$$W_{ij} = \sqrt{1-2\rho}Y_{ij} + \sqrt{\rho}\left(Z_i + Z_j\right). \tag{4.8}$$

Note that W_{ij} is normally distributed with mean zero and further

$$\begin{aligned} \mathrm{Var}\left[W_{ij}\right] &= 1, \\ \mathrm{Cov}\left[W_{ij}, W_{ik}\right] &= \rho, \\ \mathrm{Cov}\left[W_{ij}, W_{kl}\right] &= 0. \end{aligned} \tag{4.9}$$

Hence, $\boldsymbol{W}$ is distributed identically to $\boldsymbol{X}$. So, to sample from the model:

1. Sample $\boldsymbol{z}$, a length n vector of i.i.d. standard normal variables.
2. For $i < j$, generate $y \sim \mathcal{N}(0,1)$, and if

   $$y > \frac{t - \sqrt{\rho}\left(z_i + z_j\right)}{\sqrt{1-2\rho}} \tag{4.10}$$

   add edge (i, j) to the network.

If $\rho = \frac{1}{2}$, generating y is unnecessary and one can simply add edge (i, j) if $\sqrt{1/2}(z_i + z_j) \geq t$.

A Python package to generate networks along with scripts for the figures in this chapter is publicly available [148].

In order to achieve the required correlations, the algorithm above separates X_{ij} into node and edge effects. Each node is given a value Z_i and X_{ij} is created by a linear combination of Z_i and Z_j plus i.i.d. random noise Y_{ij}. We can interpret the Z's as latent variables that control the propensity for individual nodes to have edges and ρ controls the relative strength of the noise process. When $\rho = 1/2$ edges are entirely determined by the values of Z, while at $\rho = 0$ edges are entirely random and independent.

Despite this equivalent formulation, our model should not be primarily understood as a latent variable model since it was not constructed as one. Rather, the equivalent latent variable model is derived and used for algorithmic convenience. In fact, the existence of this latent variable interpretation is not surprising. As $n \to \infty$ our model is in a class of models known as *exchangeable random graphs* [149, 150]. The Aldous-Hoover theorem implies that all exchangeable random graphs have an equivalent latent variable model [149–151].

4.2 Network properties

We now turn our attention to properties of the networks created by the model.

4.2.1 Edge density

Edges in the network exist whenever the corresponding weight X_{ij} is greater than t. The marginal distribution for X_{ij} is simply a standard normal distribution. Thus,

$$E[A_{ij}] = P[A_{ij} = 1] = P[X_{ij} \geq t] = 1 - \Phi(t), \tag{4.11}$$

where $\Phi(x)$ is the cumulative distribution function for the standard normal distribution $\mathcal{N}(0,1)$. When $\rho = 0$ all edges exist independently and the model is equivalent to the random graph, $G_{n,p}$, with $p = 1 - \Phi(t)$.

The mean degree is equally simple to compute. For all ρ

$$E[k_i] = \sum_{j \neq i} E[A_{ij}] = (n-1)(1 - \Phi(t)). \tag{4.12}$$

If we want to pick t for a desired mean degree $\langle k \rangle$, it is easy to invert this to obtain

$$t = \Phi^{-1}\left(1 - \frac{\langle k \rangle}{n-1}\right). \tag{4.13}$$

4.2.2 Triangles, clustering, and degree variance

Many complex networks are observed to have large numbers of triangles. The clustering coefficient or transitivity is one way to quantify this. We can quantify the clustering with the probability that a triangle is closed, given that two of its edges already exist,

$$C = P[A_{ik} = 1 | A_{ij}, A_{jk} = 1] = \frac{P[A_{ik}, A_{ij}, A_{jk} = 1]}{P[A_{ij}, A_{jk} = 1]}. \tag{4.14}$$

The numerator of this equation corresponds to the density of triangles while the denominator corresponds to the density of two-stars (which also determines the variance of the degree distribution). Note that for simplicity we shorten the logical connective "and" (or "$\wedge$") using commas, e.g. $P[A_{ij} = 1 \ \wedge \ A_{jk} = 1] \equiv P[A_{ij}, A_{jk} = 1]$.

The marginal distributions of a MVN are themselves MVN, and are found by simply dropping the unwanted rows and columns in the correlation matrix $\mathbf{\Sigma}$. Thus, $(X_{ij}, X_{ik})^T$ will be bivariate normally distributed and $(X_{ij}, X_{ik}, X_{jk})^T$ will be trivariate normally distributed, both with correlation coefficient ρ. Introducing the Hermite polynomials $H_N(x)$ as defined in Appendix B.1, one finds that

$$P[X_{ij}, X_{ik} \geq t] = \sum_{N=0}^{\infty} \frac{\rho^N}{N!} \left[\phi(t) H_{N-1}(t)\right]^2 \tag{4.15}$$

for the density of two-stars and

$$P[X_{ij}, X_{ik}, X_{jk} \geq t] = \sum_{N=0}^{\infty} \ \sum_{i=0}^{N} \sum_{j=0}^{N-i} \frac{\rho^N \phi(t)^3}{i!\, j!\, (N-i-j)!} H_{N-1-i}(t)\, H_{N-1-j}(t)\, H_{i+j-1}(t) \tag{4.16}$$

for triangles. Both sums converge for $\rho \leq 0.5$, and we can estimate them accurately with a finite number of terms [152]. Noting that there are $\binom{n-1}{2}$ potential triangles for each node, the expected number of triangles per node is simply $\binom{n-1}{2}$ times their density

$$T = \binom{n-1}{2} P[X_{ij}, X_{ik}, X_{jk} \geq t]. \tag{4.17}$$

Plots of these functions are shown in Fig. 4.2. We find that T is much larger in these networks than in the random graph $G_{n,p}$—larger by multiple orders of magnitude. In fact, while T goes to zero in the large n limit for the random graph, in this model we find that T increases with n for large values of ρ. On the other hand, the clustering coefficient C decreases with growing number of nodes for all parameter values. This leads to a slightly paradoxical result for large ρ: in the limit $n \to \infty$ the expected number of triangles at each node goes to infinity, and the clustering coefficient still goes to zero! The reason for this is that the number of two-stars diverges faster than the number of triangles.

Equation (4.15) can also be used to compute the variance of the degree distribution. To see this note that a node of degree k has $\binom{k}{2}$ two-stars. Further, noting that

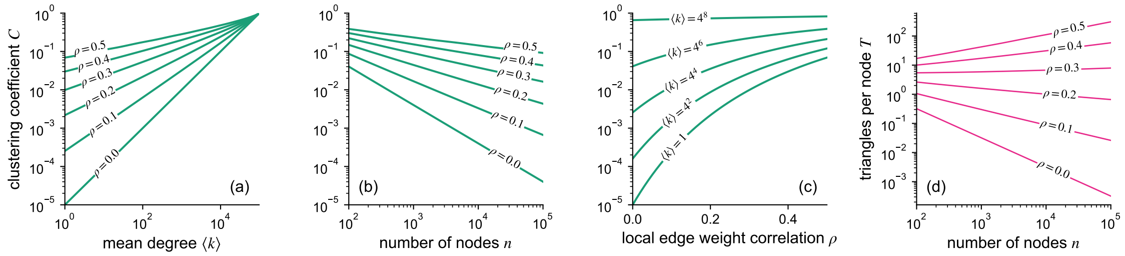


Figure 4.2: Clustering C and triangles per node T for thresholded normal data. Values are computed using the equations of Sec. 4.2.2. Clustering decreases with increasing number of nodes, however the number of triangles per node increases with growing number of nodes n for large values of ρ. Clustering increases both with increasing mean degree $\langle k\rangle$ and local edge weight correlation ρ. In panel (a) and (c) we chose $n = 100\,000$ and in panel (b) and (d), we fixed $\langle k\rangle = 4$.

there are $\binom{n-1}{2}$ potential two-stars (the same number of potential triangles) we find

$$\frac{1}{2}\left(\langle k^2\rangle - \langle k\rangle\right) = \binom{n-1}{2} P[X_{ij}, X_{ik} \geq t]. \tag{4.18}$$

Combining this with Eq. (4.12) the variance of the node degree k can be written

$$\mathrm{Var}[k] = (n-1)\Phi(t)\left[1-\Phi(t)\right] + (n-1)(n-2)\sum_{N=1}^{\infty} \frac{\rho^N}{N!}\left[\phi(t)H_{N-1}(t)\right]^2. \tag{4.19}$$

The first term is simply the variance of a binomial distribution. For $\rho = 0$ the second term vanishes and we recover the correct result for the random graph $G_{n,p}$. For $\rho > 0$ the sum is positive and monotonically increases with ρ as illustrated in Fig. 4.3.

4.2.3 Degree distribution

In the previous two subsections we gave expressions for the mean and variance of the degrees. Here we give expressions for the full distribution of degrees.

The degree distribution p_k is the probability that a node has k edges. Letting

$$f_k(y) = k \ln\left[1-\Phi(y)\right] + (n-k-1)\ln\left[\Phi(y)\right] - \frac{1}{2}\left(\frac{t-\sqrt{1-\rho}y}{\sqrt{\rho}}\right)^2, \tag{4.20}$$

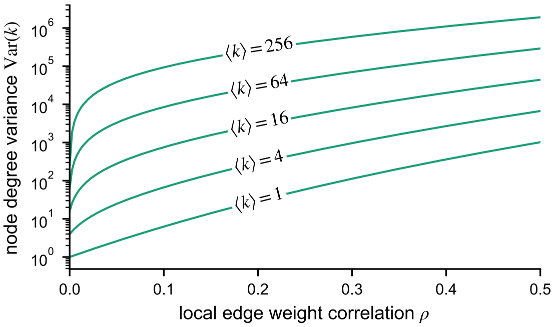


Figure 4.3: The variance of degree for thresholded normal data. The variance, Eq. (4.19), increases with ρ, the local edge weight correlation. With increasing mean degree $\langle k \rangle$, even small correlations ρ produce networks of significantly broader degree distribution than the random graph $G_{n,p}$.

the degree distribution for this model is

$$p_k = \binom{n-1}{k} \sqrt{\frac{1-\rho}{2\pi\rho}} \int_{-\infty}^{\infty} e^{f_k(y)} dy. \tag{4.21}$$

This result is derived in Appendix B.2.

The integral in Eq. (4.21) can be computed numerically to high precision using Gauss-Hermite quadrature, centered at the maximum of $f_k(y)$. Increasing the order of Gauss-Hermite quadrature (i.e. incorporating more points) increases the accuracy. The full details are in Appendix B.2.

We can also approximate the integral using Laplace's method [153], an asymptotic approximation for integrals of this form (equivalent to a first order Gauss-Hermite quadrature). The idea of the method is to replace the function $f_k(y)$ by a second order Taylor series around its maximum. For large n, the last (quadratic) term in f_k will be negligible and for $0 < k < n-1$, the maximum will be at

$$y_{0,k} = \Phi^{-1}\left(1 - \frac{k}{n-1}\right). \tag{4.22}$$

Combining this with Stirling's approximation for the binomial coefficient, we find

$$p_k \sim \frac{1}{n-1}\sqrt{\frac{1-\rho}{\rho}} \exp\left[-\left(\frac{1-2\rho}{2\rho}\right) y_{0,k}^2 + \left(\frac{t\sqrt{1-\rho}}{\rho}\right) y_{0,k} - \frac{t^2}{2\rho}\right]. \tag{4.23}$$

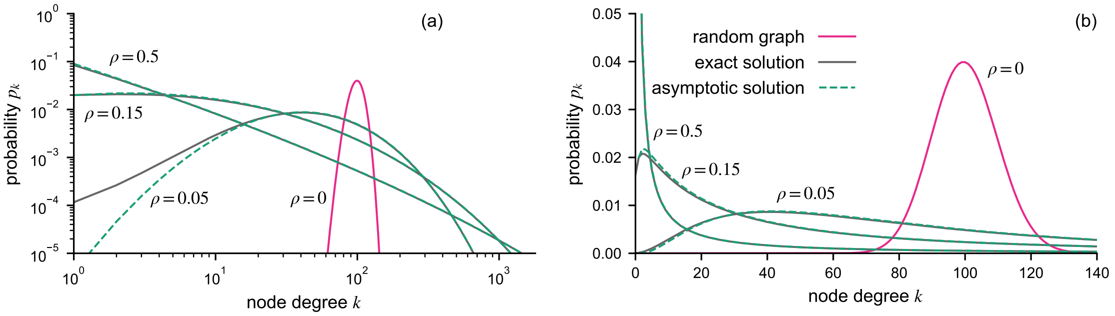


Figure 4.4: Degree distributions for thresholded normal data. We show degree distributions computed using Eq. (4.21) for $n = 100\,000$ and $\langle k \rangle = 100$ for increasing local edge weight correlation ρ in log-log (a) and linear scales (b). We also compare them to the asymptotic approximation Eq. (4.23). Note that large values of ρ produce broad degree distributions which could be easily mistaken for log-normal or power-law distributions.

Together with the closed form approximation for Φ^{-1}, given in Appendix B.3, Eq. (4.23) provides a closed form approximation for the degree distribution.

Figure 4.4 shows some example degree distributions, computed to high precision using Eq. (4.21) along with the asymptotic approximation, Eq. (4.23), where we chose $n = 100\,000$ and $\langle k \rangle = 100$.

To illustrate how these degree distributions compare to the degree distributions of real networks, we chose three data sets from different domains, and fit the model. The first data set is a network of friendships between students at a U.S. high school ($n = 2587$) [115], the second data set is a co-authorship network of researchers ($n = 16726$) [154], and the third network describes interactions between proteins ($n = 6327$) [155].

Given a number of nodes n the model under study has two free parameters, t and ρ. A simple procedure to fit the model to the data is to choose t and ρ so that the mean and variance of the model's degree distribution match the observed values. We use Eq. (4.13) to fix t and subsequently Newton's method to solve Eq. (4.19) for ρ.

The results of this exercise are shown in Fig. 4.5. The networks were chosen for their different degree distributions—note the different scales on the axes: linear, log-linear, and log-log—and the threshold model can qualitatively ape these distributions. Nevertheless, the similarity of the degree distribution should not be over-emphasized. As discussed, this model has vanishing clustering so cannot

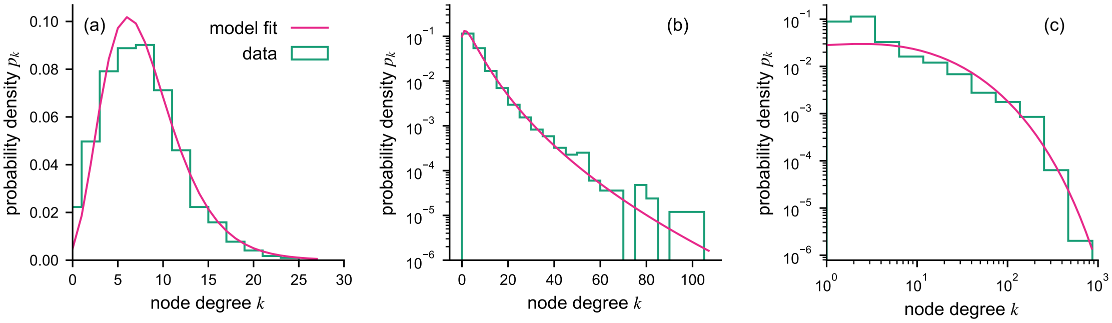


Figure 4.5: Degree histograms for three real-world networks. The networks introduced in Sec. 4.2.3 are compared against fitted distributions from the thresholded normal model. We show (a) a high school friendship network, (b) a co-authorship network between scientists, and (c) a protein–protein interaction network.

account for this observation of real-world networks.

While the degrees in the thresholded networks, $k_i = \sum_j A_{ij}$, in general follow a complicated distribution, the underlying degrees $d_i = \sum_j X_{ij}$ are always normally distributed. When $\rho = 0$, d_i is Gaussian and k_i is binomial, or Poisson in the sparse limit. When $\rho > 0$, d_i is still Gaussian, but k_i now follows a heavy-tailed distribution. Thus, the heavy-tailed distribution observed in the model is due to the combination of correlation and thresholding. Without positive correlation we observe Poisson distributions; without thresholding we observe Gaussian distributions.

4.2.4 Giant component

A well studied problem in the theory of random graphs is the formation of a large connected (giant) component. At very low densities only a handful of nodes can be reached from any other node but at some critical point a macroscopic number of nodes will be connected. For the random graph this transition occurs at a mean degree of $\langle k \rangle = 1$ [1,65,156].

To explore the effects of $\rho > 0$ we sampled from the model as described in Sec. 4.1.2 and measured the size of the second largest component as a susceptibility parameter for the phase transition. The maximum of this susceptibility parameter is used to find the transition lines in Fig. 4.6a.

We find that as ρ or n increases, the transition occurs at lower values of the mean degree. This result is in line with the configuration model for which the transition

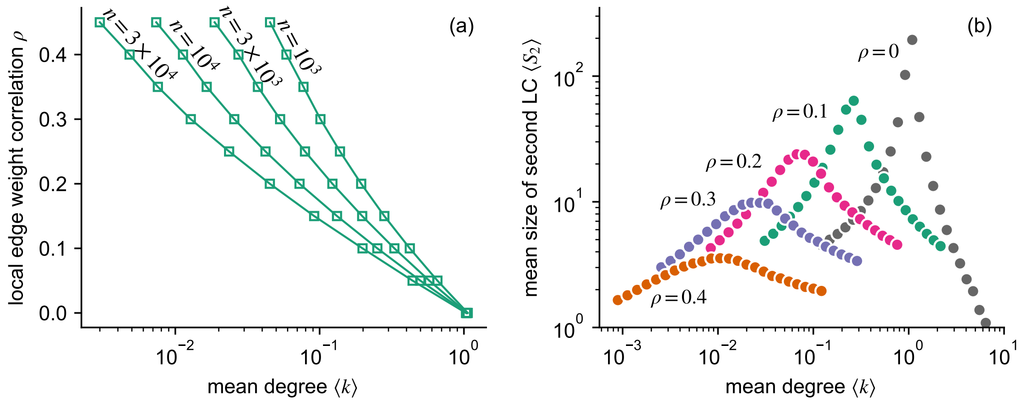


Figure 4.6: Giant component phase transition for thresholded normal data. Simulations with $1\,000 \leq n \leq 30\,000$, mean degree $10^{-3} \leq \langle k\rangle \leq 10$, $0 \leq \rho \leq 0.45$. 1000 samples were taken for each of the parameter combinations. Panel (a) shows the points of transitions for increasing number of nodes n. To the left of the line the network does not possess a giant component, while to the right it does. The transition point was computed using the mean size of the second largest component as a susceptibility parameter. Panel (b) shows an example of the susceptibility parameter for $n = 10\,000$.

point decreases with increasing variance in the degree distribution. For $\rho = 0$ we recover the standard result for the random graph.

For the other limit case, $\rho = 1/2$, recall that all edge weights can be considered to arise from node "propensities", Z_i, with $X_{ij} = \sqrt{1/2}(Z_i + Z_j)$. This implies that all nodes that are connected to any other nodes must also be connected to the node with maximum propensity $Z_{\max}$. The size of the largest component is then given by this node's degree plus 1, $k_{\max}+1$. The second largest component is then always of size 1. We therefore omit $\rho = 1/2$ in the numerical analysis.

4.2.5 Shortest path lengths

Another phenomenon well established in the complex networks literature is that randomly chosen nodes often have surprisingly short paths between them. This is often referred to as the "six degrees of separation" or "small-world" phenomenon [1, 41]. By a common definition, network models are considered to demonstrate this property if the average shortest path length $\langle d_{ij}\rangle$ between nodes grows logarithmically (or slower) as the number of nodes increases [1].

Using the method described in Sec. 4.1.2, we sampled from the threshold model to verify that it displays this property. We looked at networks with between 100

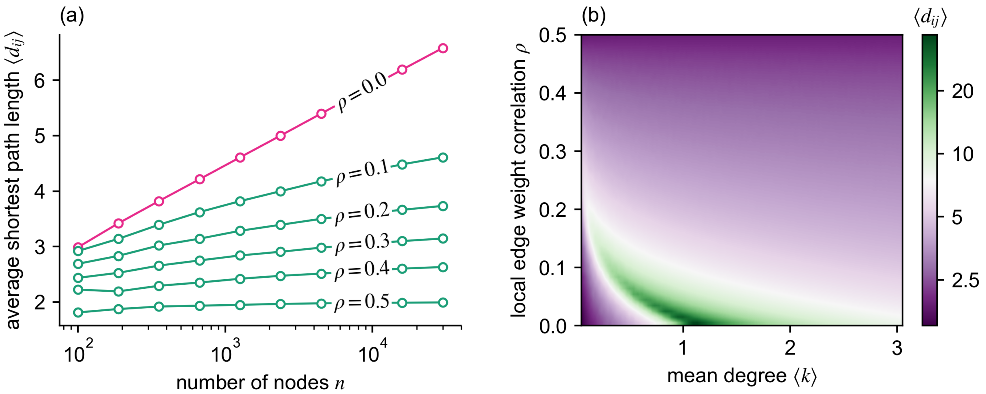


Figure 4.7: Average shortest paths in thresholded normal data. Panel (a) shows the scaling of the average shortest path in the largest connected component with the number of nodes, n. We fix the mean degree $\langle k \rangle = 5$ and each point is averaged over 200 samples. For $\rho = 0$ we recover the result for the random graph $G_{n,p}$ where $\langle d_{ij} \rangle \propto \log n$. For non-zero correlation, the average shortest path length increases slower than logarithmically. In panel (b) we show the average shortest path length for different mean degrees and values of ρ for networks with $n = 10\,000$, again sampled 200 times for each parameter combination.

and 30 000 nodes, with mean degree $\langle k \rangle = 5$, and investigated the influence of increasing edge weight correlation ρ. After sampling a network from the model we computed the average shortest path length $\langle d_{ij} \rangle$ on the largest (giant) component. For each parameter combination we computed the mean by averaging 200 sampled networks.

The results are shown in Fig. 4.7a. Since it is well known that the random graph $G_{n,p}$ has short shortest paths [1, 157] it is unsurprising that the threshold model does also (recall, for $\rho = 0$ they are equivalent, and we see the standard $\langle d_{ij} \rangle \propto \log n$ scaling behavior). For $\rho > 0$ we see that average shortest path lengths grow significantly slower than logarithmically, a behavior sometimes referred to as "ultra small-world" and often related to networks with power-law degree distribution [158, 159]. In our model, the effect appears despite the fact that the degree distribution does not follow a power-law.

As discussed, when $\rho = 1/2$ all edge weights can be considered to arise from node propensities Z_i, such that $X_{ij} = \sqrt{1/2}(Z_i + Z_j)$. All nodes are then either disconnected or part of the giant component, and the node with maximum propensity $Z_{\max}$ is connected to all nodes in the giant component. Hence, all nodes in the giant component are either directly connected or can reach each other in two steps through the maximum-degree node. So, when $\rho = 1/2$ the average shortest path length must be $1 \leq \langle d_{ij} \rangle < 2$.

4.3 Discussion

In this chapter we studied the effects of correlation on relational data. We started with a simple model of multivariate normally distributed data, with only one free parameter, ρ, controlling local correlations. We then demonstrated that thresholding this normally distributed correlated data reproduces many of the properties commonly associated with complex networks. In particular, we find that the combined effects of correlation and thresholding leads to heavy-tailed degree distributions, relatively large numbers of triangles, and short average path lengths.

The underlying data, $\boldsymbol{X}$, in the model we introduce would not usually be considered complex. It is generated from a highly symmetric multivariate normal distribution with only one free parameter. Since every pair of nodes has some level of interaction, the graphical interpretation for $\boldsymbol{X}$ would be a weighted complete graph, with all edge weights (and linear combinations thereof) normally distributed. For example, the "degrees", $d_i = \sum_j X_{ij}$, are normally distributed. And yet, after thresholding the networks show several properties commonly associated with complex networks.

One way to think about these results is in the context of the central limit theorem. Whenever interaction strengths are the aggregate result of a large number of processes then we expect $\boldsymbol{X}$ to be normally distributed. Constructing a simple graph from these data can lead to complex networks. This provides one simple explanation for the ubiquity of complex networks—they can arise as a consequence of the central limit theorem.

Of course, for most scientific questions of interest the exact details of the mechanisms and structure are what matter. In a social network, for example, answering the question "who influences whom, and why?" is far from trivial and the fact that the network has certain commonly observed properties is usually incidental.

In summary, straightforward assumptions lead to several of the properties associated with complex networks. If a network arises by a simple thresholding procedure then finding that it is "complex" need be no more surprising than finding a bell-shaped curve in a regular data set.

CHAPTER 5

Message Passing for Complex Networks

This chapter is adapted from the published results of G. T. Cantwell and M. E. J. Newman, Message passing on networks with loops. *Proceedings of the National Academy of Sciences* **116**(47), 23398–23403 (2019) [100].

Message passing [127, 160, 161], also known as belief propagation or the cavity method, is a fundamental technique for the quantitative calculation of a wide range of network properties, with applications to Bayesian inference [161], NP-hard computational problems [127,162], statistical physics [43,127,163], epidemiology [164], community detection [83], and signal processing [165, 166], among many other things. Message passing can be used both as a numerical method for performing explicit computer calculations and as a tool for analytic reasoning about network properties, leading to new formal results about percolation thresholds [43], algorithm performance [83], spin glasses [167], and other topics. Many of the most powerful new results concerning networks in recent years have been derived from applications of message passing in one form or another. Indeed, in Ch. 3 we used the formalism to provide analytic insight into data recovery tasks.

Despite the central importance of the message passing method, however, it also has a substantial and widely discussed shortcoming: it only works on trees, i.e., networks that are free of loops [127]. More generously, one could say that it works to a good approximation on networks that are "locally tree-like," meaning that they may contain long loops but no short ones, so that local neighborhoods within the network take the form of trees. However, most real-world networks that occur in practical applications of the method contain short loops, often in large numbers. When applied to such "loopy" networks the method can give poor results, and in the worst cases can fail to converge to an answer at all.

In this chapter, we consider a remedy for this problem. We present a series of methods of increasing elaboration for the solution of problems on networks with loops. The first method in the series is equivalent to the standard message passing algorithm of previous work, which gives poor results in many cases. The last in the series gives exact results on any network with any structure, but is too complicated for practical application in most situations. In between lies a range of methods that give progressively better approximations, and which can be highly accurate in practice, as we will show, yet still simple enough for ready implementation. Indeed even the second member of the series—just one step better than the standard message passing approach—already gives remarkably good results in real-world conditions. We demonstrate our approach with two example applications. The first is to the solution of the bond percolation problem on an arbitrary network, including the calculation of the size of the percolating cluster and the distribution of sizes of small clusters. The second is to the calculation of the spectra of sparse symmetric matrices, where we show that our method is able to calculate the spectra of matrices far larger than those accessible by conventional numerical means.

A number of approaches have been proposed previously for message passing on loopy networks. The most basic of these, which goes by the name of "loopy belief propagation," is simply to apply the standard message passing equations, ignoring the fact that they are known to be incorrect in general. While this might seem rash, it gives reasonable answers in some cases [166] and there are formal results showing that it can give bounds on the true value of a quantity in others [43, 127]. Perturbation theories that treat loopy belief propagation as a zeroth-order approximation have also been considered [168]. Broadly, it is found that these methods are suitable for networks that contain a sub-extensive number—and hence a vanishing density—of short loops, but not for networks with a non-vanishing density.

Some progress has been made for the case of networks that are composed of small subgraphs or "motifs" which are allowed to contain loops but which on a larger scale are connected in a loop-free way [169–171]. For such networks one can write down exact message passing equations that operate at the higher level of the motifs and which give excellent results for problems such as structural phase transitions in networks, network spectra, and the solution of spin models [163,169–172]. While effective for theoretical calculations on model networks, however, this approach is of little use in practical situations. To apply it to an arbitrary network one would first need to find a suitable decomposition of the network into motifs, and no general method for doing this is currently known, nor even whether such a decomposition exists.

A third approach is the method known as "generalized belief propagation," which has some elements in common with the motif-based approach but is derived in a different manner, from approximations to the free energy [173, 174]. This method, which is focused particularly on the solution of inference problems and related probabilistic calculations on networks, involves a hypergraph-like extension of traditional message passing that aims to calculate the joint distributions of three or more random variables at once, by contrast with the standard approach which focuses on two-variable distributions. Generalized belief propagation was not originally intended as a method for solving problems on loopy networks but can be used in that way in certain cases. It is, however, quite involved in practice, requiring the construction of a nested set of regions and sub-regions within the network, leading to complex sets of equations.

Here, we take a different approach. In the following sections we directly formulate a message passing framework that works on real-world complex networks containing many short loops by incorporating the loops themselves directly into the message passing equations. In traditional message passing algorithms each node receives a message from each of its neighbors. In our approach they also receive messages from nodes they share loops with. By limiting the loops considered to a fixed maximum length, we develop a series of progressively better approximations for the solution of problems on loopy networks. The equations become more complex as loop length increases but, as we will show, the results given by the method are already impressively accurate even at shorter lengths.

5.1 Message passing with loops

Message passing methods calculate some value or state on the nodes of a network by repeatedly passing information between nearby nodes until a self-consistent solution is reached. The approach we propose is characterized by a series of message passing approximations defined as follows. In the zeroth approximation, which is equivalent to the standard message passing method, we assume there are no loops in our network. This implies that the neighbors of a node are not connected to each other, which means they have independent states. It is this independence that makes the standard method work. In the next approximation we no longer assume that neighbors are independent. Instead, we assume that any correlation can be accounted for by direct edges between the neighbors, which is equivalent to allowing the network to contain triangles, the shortest possible kind of loop. In the next approximation after this, we assume that neighbor correlations can be accounted for by direct edges plus paths of length 2 between neighbors. Generally, in the rth approximation we assume that correlations between neighbors can be accounted for by paths of length r and shorter.

These successive approximations can be thought of as expressing the properties of nodes in terms of increasingly large neighborhoods and the edges they contain. The zeroth neighborhood $N_i^{(0)}$ of node i contains i's immediate neighbors and the edges connecting them to i, but nothing else. The first neighborhood $N_i^{(1)}$ contains i's immediate neighbors and edges plus all length one paths between neighbors of i. The second neighborhood $N_i^{(2)}$ contains i's neighbors and edges plus all length one and two paths between neighbors of i, and so forth. Figure 5.1 shows an example of how these neighborhoods are constructed.

Just as the conventional message passing algorithm is exact on trees, our algorithms will be exact on networks with short loops. We define a *primitive cycle* of length r starting at node i to be a cycle such that at least one edge is not on a shorter cycle beginning and ending at i. Then our rth approximation is exact on networks that contain primitive cycles of length $r + 2$ or less only. For networks that contain longer primitive cycles it will be an approximation, although as we will see it may be a good one.

5.2 Applications

Our approach is best demonstrated by example. In this section we derive message passing equations on loopy networks for two specific applications: the calculation of cluster sizes for bond percolation and the calculation of the spectra of sparse matrices.

5.2.1 Percolation

Consider the bond percolation process on an undirected network of n nodes, where each edge is occupied independently with probability p [175,176]. Occupied edges form connected clusters and we wish to know the distribution of the sizes of these clusters and whether there exists a giant or percolating cluster that occupies a non-vanishing fraction of the network in the limit of large network size.

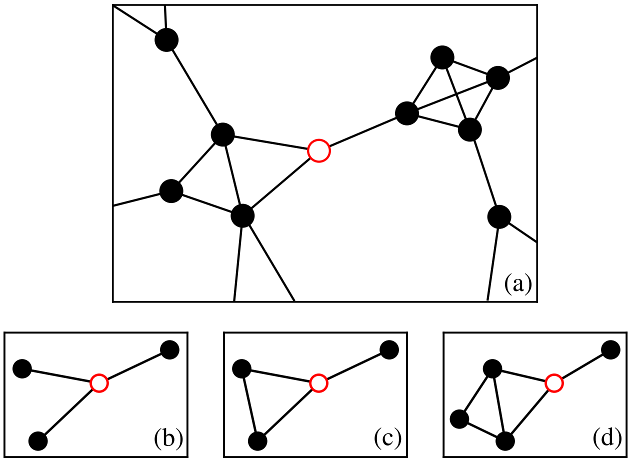


Figure 5.1: Constructing r-neighborhoods. (a) A node (open circle) and its immediate surroundings in a network. (b) In the zeroth (tree) approximation the neighborhood we consider consists of the neighbors of the focal node only. (c) In the first approximation we also include all length 1 paths between the neighbors. (d) In the second approximation we include all paths of length 1 and 2, and so forth.

Let us define the rth neighborhood $N_i^{(r)}$ of node i as previously, then define a random variable Γ_i for our percolation process to be the set of nodes within $N_i^{(r)}$ that are reachable from i by traversing occupied edges only. Our initial goal will be to compute the probability $\pi_i(s)$ that node i belongs to a non-percolating cluster of size s. We will do this in two stages. First we will compute the conditional probability $\pi_i(s|\Gamma_i)$ of belonging to a cluster of size s given the set of reachable nodes. Then we will average over Γ_i to get the full probability $\pi_i(s)$.

Suppose that node i belongs to a cluster of size s. If our network contains no primitive cycles longer than $r+2$, then the set of nodes Γ_i would become disconnected from one another were we to remove all edges in the neighborhood $N_i^{(r)}$—the removal of these edges removes any connections within the neighborhood and there can be no connections via paths outside the neighborhood since such a path would constitute a primitive cycle of length longer than $r+2$. Hence the sizes s_j of the clusters to which the nodes in $N_i^{(r)}$ would belong after this removal must sum to $s-1$ (the sth and last node being provided by i itself). We can thus relate $\pi_i(s)$ to the quantities $\pi_{i\leftarrow j}(s)$, the probability that node j is in a cluster of size s once the edges in $N_i^{(r)}$ are removed. The resulting formula is

$$\pi_i(s|\Gamma_i) = \sum_{\{s_j : j\in\Gamma_i\}} \biggl[\prod_{j\in\Gamma_i} \pi_{i\leftarrow j}(s_j)\biggr] \delta\bigl(s-1, \textstyle\sum_{j\in\Gamma_i} s_j\bigr). \tag{5.1}$$

We can now write a generating function for $\pi_i(s|\Gamma_i)$,

$$\begin{aligned} H_i(z|\Gamma_i) &= \sum_{s=1}^{\infty} \pi_i(s|\Gamma_i)\, z^s \\ &= \sum_{s=1}^{\infty} z^s \Biggl\{ \sum_{\{s_j : j\in\Gamma_i\}} \biggl[\prod_{j\in\Gamma_i} \pi_{i\leftarrow j}(s_j)\biggr] \delta(s-1, \textstyle\sum_{j\in\Gamma_i} s_j) \Biggr\} \\ &= z \prod_{j\in\Gamma_i} \sum_{s_j=1}^{\infty} z^{s_j} \pi_{i\leftarrow j}(s_j). \end{aligned} \tag{5.2}$$

To calculate the full probability $\pi_i(s)$ we average $\pi_i(s|\Gamma_i)$ over sets Γ_i to get $\pi_i(s) = \bigl\langle \pi_i(s|\Gamma_i) \bigr\rangle_{\Gamma_i}$, with the average weighted according to the sum of the probabilities of all edge configurations that correspond to a particular Γ_i. The probability of any individual edge configuration is simply $p^k(1-p)^{m-k}$, where p is the edge occupation probability as previously, m is the number of network edges in the neighborhood $N_i^{(r)}$, and k is the number that are occupied. Performing the same average on (5.2) gives us

$$H_i(z) = \sum_{s=1}^{\infty} \pi_i(s)\, z^s = zG_i\bigl(\mathbf{H}_{i\leftarrow}(z)\bigr), \tag{5.3}$$

where $G_i(\mathbf{y}) = \bigl\langle \prod_{j\in N_i^{(r)}} y_j^{w_{ij}} \bigr\rangle_{\Gamma_i}$ is a generating function for the random variable w_{ij}, which takes the value 1 if $j \in \Gamma_i$ and 0 otherwise, and $\mathbf{H}_{i\leftarrow}(z)$ is the vector with elements $H_{i\leftarrow j}(z)$ for nodes j in $N_i^{(r)}$.

To complete the calculation we need to evaluate $H_{i\leftarrow j}(z)$, whose computation follows the same logic as for $H_i(z)$, the only difference being that in considering the neighborhood of node j we must remove the entire neighborhood of i first, as described above. Doing this leads to

$$H_{i\leftarrow j}(z) = zG_{i\leftarrow j}\bigl(\mathbf{H}_{j\leftarrow}(z)\bigr), \tag{5.4}$$

where $G_{i\leftarrow j}(\mathbf{y})$ is the equivalent of $G_i(\mathbf{y})$ when $N_i^{(r)}$ is removed. (A detailed derivation of (5.4) is given in Appendix C.1.) If we can solve this equation self-consistently for $\mathbf{H}_{j\leftarrow}(z)$, we can substitute the solution into (5.3) to compute the full cluster size generating function. The message passing method involves solving (5.4) by simple iteration: we choose suitable starting values, for instance at random, and iterate the equations to convergence.

From the cluster size generating function we can calculate a range of quantities of interest. For example, the probability that node i belongs to a small cluster (of any size) is $H_i(1) = \sum_s \pi_i(s)$. If it does not belong to a small cluster then necessarily it belongs to the percolating cluster and hence the expected fraction S of the network taken up by the percolating cluster is

$$S = 1 - \frac{1}{n}\sum_i H_i(1). \tag{5.5}$$

Similarly, the average value of s_i is

$$\begin{aligned} \langle s_i \rangle &= \sum_{s=1}^{\infty} s\pi_i(s) = H_i'(1) \\ &= H_i(1) + \sum_{j\in N_i^{(r)}} H_{i\leftarrow j}'(1)\,\partial_j G_i(\mathbf{H}_{i\leftarrow}), \end{aligned} \tag{5.6}$$

where H' is the derivative of H and $\partial_j G_i$ is the partial derivative of G_i with respect to its jth argument. $H_{i\leftarrow j}'(1)$ can be found by differentiating (5.4) and setting $z = 1$ to give the self-consistent equation

$$H_{i\leftarrow j}'(1) = H_{i\leftarrow j}(1) + \sum_{k\in N_{j\setminus i}^{(r)}} H_{j\leftarrow k}'(1)\,\partial_k G_{i\leftarrow j}\left(\mathbf{H}_{j\leftarrow}\right), \tag{5.7}$$

where $N_{j\setminus i}^{(r)}$ denotes the neighborhood $N_j^{(r)}$ with $N_i^{(r)}$ removed.

While these equations are straightforward in principle, implementing them in practice presents some additional challenges. Computing the generating functions $G_i(\mathbf{y})$ and $G_{i\leftarrow j}(\mathbf{y})$ can be demanding, since it requires us to perform an average over the occupancy configurations of all edges within the neighborhoods $N_i^{(r)}$ and $N_{j\setminus i}^{(r)}$, and the number of configurations increases exponentially with neighborhood size. For small neighborhoods, such as those found on low-dimensional lattices, it is feasible to average exhaustively, but for many complex networks this is not possible. In such cases we instead approximate the average by Monte Carlo sampling of configurations—see Appendix D for details. A nice feature of the Monte Carlo procedure is that the samples need be taken only once for the entire calculation and can then be reused on successive iterations of the message passing process.

In practice the method gives excellent results. We show example applications to two real-world networks in Fig. 5.2, the first a social network of coauthorship relations between scientists in the field of condensed matter physics [154] and the second a network of trust relations between users of the Pretty Good Privacy (PGP) encryption software [177]. Both networks have a high density of short loops. For each network the figure shows, as a function of p, several different estimates of both the average size $\langle s \rangle$ of a small cluster and the size S of the percolating cluster as a fraction of n. First we show an estimate made using standard message passing (dashed line)—the $r = 0$ approximation in our nomenclature—which ignores loops and is expected to give poor results. Second, we show the next two approximations in our series, those for $r = 1$ and $r = 2$ (dotted and solid lines respectively), with $G_i(\mathbf{y})$ and $G_{i\leftarrow j}(\mathbf{y})$ estimated by Monte Carlo sampling as described above. We use only eight samples for each node i but the results are nonetheless impressively accurate. Third, we show for comparison a direct numerical estimate of the quantities in question made by conventional simulation of the percolation process.

For both networks we see the same pattern. The traditional message passing method fares poorly, as expected, giving estimates that are substantially in disagreement with the simulation results, particularly for the calculations of average cluster size. The $r = 1$ approximation, on the other hand, does significantly better and the $r = 2$ approximation does better still, agreeing closely with the numerical results for all measures on both networks. In these examples at least, it appears that the $r = 2$ method gives accurate results for bond percolation, where standard message passing fails.

The message passing algorithm is relatively fast. For $r \leq 1$ each node receives a message from each neighbor on each iteration, and so on a network with mean degree c there are cn messages passed per iteration. For $r \geq 2$ the number of messages depends on the network structure. On trees the number of messages remains unchanged at cn as r increases but on networks with loops it grows and for large numbers of loops it can grow exponentially. In the common sparse case where the size of the neighborhoods does not grow with n, however, the number of messages is linear in n for fixed r and hence so is the running time for each

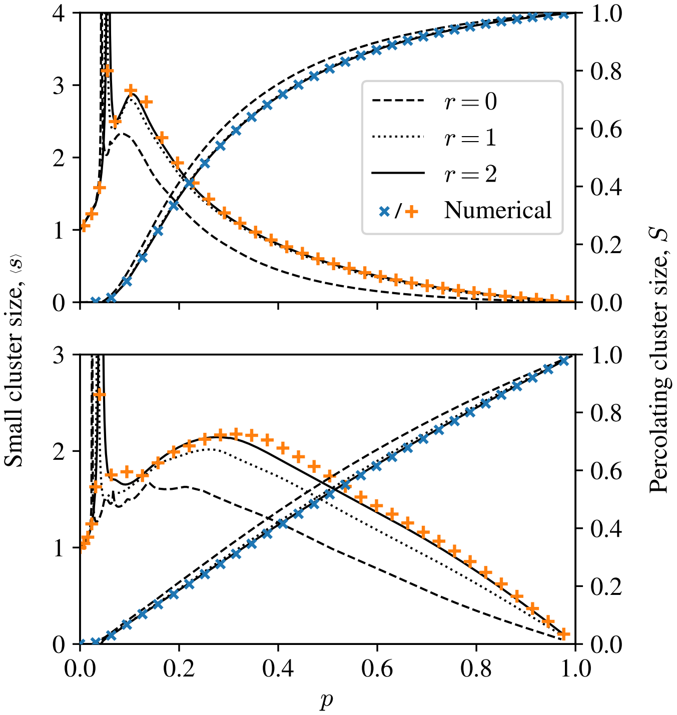


Figure 5.2: Percolation simulations and equation based results. Percolating cluster size (× symbols) and average cluster size (+ symbols) for two real-world networks. Top: the largest component of a coauthorship network of 13,861 scientists [154]. Bottom: a network of 10,680 users of the PGP encryption software [177].

iteration. It is not known in general how many iterations are needed for message passing methods to reach convergence, but elementary heuristic arguments suggest the number should be on the order of the diameter of the network, which is typically $O(\log n)$. Thus we expect overall running time to be $O(n \log n)$ for sparse networks at fixed r.

This makes the algorithm quite efficient, although direct numerical simulations of percolation run comparably fast, so the message passing approach does not offer a speed advantage over traditional approaches. However, the two approaches are calculating different things. Traditional simulations of percolation perform a calculation for one particular realization of bond occupancies. If we want average values over many realizations we must perform the average explicitly, repeating the whole simulation for each realization. The message passing approach, on the other hand, computes the average over realizations in a single calculation and no repetition is necessary, making it potentially the faster method in some situations.

In the next section we demonstrate another example application of our method: the calculation of the spectrum of a sparse matrix. For this application traditional and message passing calculations differ substantially in their running time, the message passing approach being much faster, making calculations possible for large systems whose spectra cannot be computed in any reasonable amount of time by traditional means.

5.2.2 Matrix spectra

For our second example application we show how the message passing method can be used to compute the eigenvalue spectrum of a sparse symmetric matrix. Any $n \times n$ symmetric matrix can be thought of as an undirected weighted network on n nodes and we can use this equivalence to apply the message passing method to such matrices.

The spectral density of a symmetric matrix $\mathbf{A}$ is the quantity

$$\rho(x) = \frac{1}{n} \sum_{k=1}^{n} \delta(x - \lambda_k), \tag{5.8}$$

where λ_k is the kth eigenvalue of $\mathbf{A}$, and $\delta(x)$ is the Dirac delta function. Following standard arguments [178], we can show that the spectral density is equal to the imaginary part of the complex function

$$\rho(z) = -\frac{1}{n\pi}\sum_{k=1}^{n}\frac{1}{z-\lambda_k} = -\frac{1}{n\pi}\operatorname{Tr}(z\mathbf{I}-\mathbf{A})^{-1}$$
$$= -\frac{1}{n\pi z}\sum_{i=1}^{n}\sum_{s=0}^{\infty}\frac{X_i^s}{z^s}, \tag{5.9}$$

where $X_i^s = [\mathbf{A}^s]_{ii}$ is the ith diagonal element of $\mathbf{A}^s$, and $z = x + i\eta$ and we take the limit as $\eta \longrightarrow 0$ from above. The imaginary part η acts as a resolution parameter that broadens the delta-function peaks in (5.8) by an amount roughly equal to its value.

The quantities $X_i^s = [\mathbf{A}^s]_{ii}$ can be related to sums over closed walks in the equivalent network. If we consider the "weight" of a walk to be the product of the matrix elements on the edges it traverses, then X_i^s is the sum of the weights of all closed walks of length s that start and end at node i.

A closed walk from i need not visit i only at its start and end, however. It can return to i any number of times over the course of the walk. We will call the the simplest case, where it returns just once at the end of the walk, an *excursion*. A more general closed walk that returns to node i exactly m times can be thought of as a succession of m excursions. Such a walk will have length s if those m excursions have lengths $s_1 \ldots s_m$ with $\sum_{u=1}^{m} s_u = s$.

With this in mind, let Y_i^s be the sum of the weights of all *excursions* of length s that start and end at node i. Then the sum X_i^s over *closed walks* of length s can be written in terms of Y_i^s as

$$X_i^s = \sum_{m=0}^{\infty}\left[\sum_{s_1=1}^{\infty}\cdots\sum_{s_m=1}^{\infty}\delta\big(s, \textstyle\sum_{u=1}^{m} s_u\big)\displaystyle\prod_{u=1}^{m} Y_i^{s_u}\right]. \tag{5.10}$$

Using this result, and defining the function

$$H_i(z) = \sum_{s=1}^{\infty}\frac{Y_i^s}{z^{s-1}}, \tag{5.11}$$

we find after some algebra that

$$\rho(z) = -\frac{1}{n\pi}\sum_{i=1}^{n}\frac{1}{z - H_i(z)}. \tag{5.12}$$

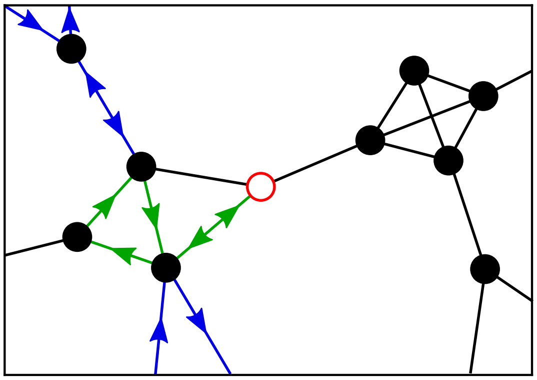


Figure 5.3: An example of an excursion. An excursion from the central node (open circle) is equivalent to an excursion inside the neighborhood, shown with green arrows, plus closed walks to regions outside of the neighborhood, shown in blue.

(See Appendix C.2 for a detailed derivation.) Thus, if we can calculate $H_i(z)$ then we can calculate $\rho(z)$. This we do as follows.

Consider the neighborhood $N_i^{(r)}$ around i. If there are no primitive cycles of length longer than $r+2$ in our network then all cycles starting at i are already included within the neighborhood, which means that any excursion from i takes the form of an excursion w within the neighborhood plus some number of additional closed walks outside the neighborhood each of which starts at one of the nodes in w and returns some time later to the same node—see Fig. 5.3. The additional walks must necessarily return to the same node they started at since if they did not they would complete a cycle outside the neighborhood, of which by hypothesis there are none.

Let the length of the excursion w be $l+1$, meaning that it visits l nodes $j_1 \ldots j_l$ (not necessarily distinct) within the neighborhood other than the starting node i, and let s_j be the length of the external closed walk (if any) that starts at node j, or zero if there is no such walk. The total length of the complete excursion from i will then be $l+1+\sum_{j\in w} s_j$ and the sum of the weights of all excursions of length s with w as their foundation will be

$$|w| \sum_{\{s_j:j\in w\}} \delta\big(s, l+1+\textstyle\sum_{j\in w} s_j\big) \displaystyle\prod_{j\in w} X_{i\leftarrow j}^{s_j}, \tag{5.13}$$

where $|w|$ is the weight of w itself and $X_{i\leftarrow j}^{s}$ is the sum of weights of length-s walks from node j if the neighborhood $N_i^{(r)}$ is removed from the network. By a similar argument to the one that led to (5.10), we can express $X^s_{i\leftarrow j}$ in terms of the sum $Y^s_{i\leftarrow j}$ of excursions from j thus:

$$X^s_{i\leftarrow j} = \sum_{m=0}^{\infty} \left[\sum_{s_1=1}^{\infty} \cdots \sum_{s_m=1}^{\infty} \delta\big(s, \textstyle\sum_{u=1}^{m} s_u\big) \displaystyle\prod_{u=1}^{m} Y^{s_u}_{i\leftarrow j} \right]. \tag{5.14}$$

And the quantity Y^s_i appearing in (5.11) can be calculated by summing (5.13) first over the set of excursions of length $l+1$ in the neighborhood of i and then over l. This allows us to write (5.11) as

$$H_i(z) = \sum_{w\in W_i} |w| \prod_{j\in w} \frac{1}{z - H_{i\leftarrow j}(z)}, \tag{5.15}$$

where W_i is the complete set of excursions of all lengths in the neighborhood of i and we have defined

$$H_{i\leftarrow j}(z) = \sum_{s=1}^{\infty} \frac{Y^s_{i\leftarrow j}}{z^{s-1}}. \tag{5.16}$$

Following an analogous line of argument for this function we can show similarly that

$$H_{i\leftarrow j}(z) = \sum_{w\in W_{j\setminus i}} |w| \prod_{k\in w} \frac{1}{z - H_{j\leftarrow k}(z)}. \tag{5.17}$$

Equation (5.17) defines our message passing equations for the spectral density. By iterating these equations to convergence from suitable starting values we can solve for the values of the messages $H_{i\leftarrow j}(z)$, then substitute into Eqs. (5.12) and (5.15) and to get the spectral density itself.

As with our percolation example, the utility of this approach relies on our having an efficient method for evaluating the sum in (5.17). Fortunately there is such a method, as follows. Let $\mathbf{v}_{i\leftarrow j}$ be the vector with elements $v_{i\leftarrow j,k} = A_{jk}$ if nodes j and k are directly connected in $N^{(r)}_{j\setminus i}$ and 0 otherwise. Further, let $\mathbf{A}^{i\leftarrow j}$ be the matrix of the neighborhood of j with the neighborhood of i removed, such that

$$A^{i\leftarrow j}_{kl} = \begin{cases} A_{kl} & \text{for } k,l \neq j \text{ and edge } (k,l) \in N^{(r)}_{j\setminus i}, \\ 0 & \text{otherwise,} \end{cases} \tag{5.18}$$

and let $\mathbf{D}^{i\leftarrow j}(z)$ be the diagonal matrix with entries $D^{i\leftarrow j}_{kk} = z - H_{j\leftarrow k}(z)$. As shown in the Appendix C.2, (5.17) can then be written

$$H_{i\leftarrow j}(z) = A_{jj} + \mathbf{v}_{i\leftarrow j}^T \big(\mathbf{D}^{i\leftarrow j} - \mathbf{A}^{i\leftarrow j}\big)^{-1} \mathbf{v}_{i\leftarrow j}. \tag{5.19}$$

Since the matrices in this equation are the size of the neighborhood, each message update requires us to invert only a small matrix, which gives us a linear-time algorithm for each iteration of the message passing equations and an overall running time of $O(n \log n)$ for sparse networks with fixed neighborhood sizes, or for the equivalent sparse matrices.

As an example of this method, we show in Fig. 5.4 spectra for the same two real-world networks that we used in Fig. 5.2. To demonstrate the flexibility of the method we calculate different spectra in the two cases: for the coauthorship network we calculate the spectrum of the graph Laplacian; for the PGP network we calculate the spectrum of the adjacency matrix. For each network the black curve in the figure shows the spectral density calculated using the message passing method with $r = 1$. We also calculate the full set of eigenvalues of each network directly using traditional numerical methods and substitute the results into (5.9) to compute the spectral density, shown as the shaded areas in the figure. As we can see, the agreement between the two methods is excellent for both networks. There are a few regions where small differences are visible but in general they agree closely. Extending the calculation to the next ($r = 2$) approximation gives a modest further improvement in the results.

The $O(n \log n)$ running time of the message passing algorithm significantly outstrips that of traditional numerical diagonalization. Complete spectra are normally calculated using the QR algorithm, which runs in time $O(n^3)$ and is consequently much slower as system size becomes large. The Lanczos algorithm is faster, but typically gives only a few leading eigenvalues and not a complete spectrum—it takes time $O(rn)$ to compute r eigenvalues of a sparse matrix. The kernel polynomial method [179] is capable of computing complete spectra for sparse matrices, but requires Monte Carlo evaluation of the traces of large matrix powers which has slow convergence and is always only approximate, even in cases where our method gives exact results.

This opens up the possibility of using our approach to calculate the spectral density of networks and matrices significantly larger than those that can be tackled by traditional means. As an example, we have used the message passing method to compute the spectral density of one network with 317 080 nodes. This is sig-

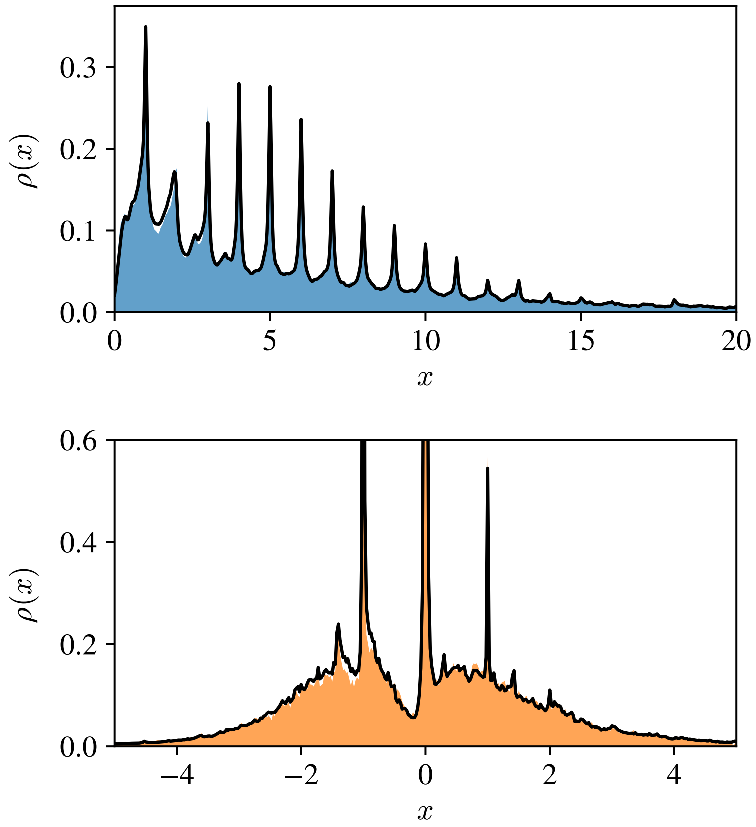


Figure 5.4: Matrix spectra for two real-world networks. We show the same two networks that were used in Fig. 5.2. Top: the spectrum of the graph Laplacian of the coauthorship network. Bottom: the spectrum of the adjacency matrix of the PGP network. The shaded areas show the spectral density calculated by direct numerical diagonalization. The black lines show the $r = 1$ message-passing approximation. The broadening parameter η was set to 0.05 in the top panel and 0.01 in the bottom panel.

nificantly larger than the largest systems that can be diagonalized using the QR algorithm, which on current (non-parallel) commodity hardware is limited to a few tens of thousands of nodes in practical running times.

5.3 Discussion

In this chapter we have described a class of message passing methods for performing calculations on networks that contain short loops, a situation in which traditional message passing often gives poor results or may fail to converge entirely. We derive message passing equations that account for the effects of loops up to a fixed length that we choose, so that calculations are exact on networks with no loops longer than this. In practice we achieve excellent results on real-world networks by accounting for loops up to length three or four only, even if longer loops are present.

We have demonstrated our approach with two example applications, one to the calculation of bond percolation properties of networks and the other to the calculation of the spectra of sparse matrices. In the first case we develop message passing equations for the size of the percolating cluster and the average size of small clusters and find that these give good results, even on networks with an extremely high density of short loops. For the calculation of matrix spectra, we develop a message passing algorithm for the spectral density that gives results in good agreement with traditional numerical diagonalization but in much shorter running times. Where traditional methods are limited to matrices with at most a few tens of thousands of rows and columns, our method can be applied to cases with hundreds of thousands at least.

There are a number of possible directions for future work on this topic. Chief among them is the application of the method to other classes of problems, such as epidemiological calculations, graph coloring, or spin models. In the next chapter, we explore how these ideas can be applied to evaluating entropy and partition functions for high dimensional models. Many other extensions of the calculations in this chapter are also possible, including the incorporation of longer primitive cycles in the message passing equations, development of more efficient algorithms for very large systems, and applications to individual examples of interest such as the computation of spectra for very large graphs. Finally, while our example applications are to real-world networks, the same methods could in principle be applied to model networks, and in particular to ensembles of random graphs, which opens up the possibility of additional analytic results about such models.

CHAPTER 6

Entropy and Partition Functions of Complex High-Dimensional Models

The partition function is one of the most important quantities associated with a model in statistical physics. Indeed, an analytic calculation of the partition function is considered to provide a solution for the model—models for which we can calculate the partition function are "solved" [180, 181]. A model is defined by its set of allowed microstates, $\mathcal{X} = \{\boldsymbol{x}\}$, and an associated energy for each one, $H(\boldsymbol{x})$. In the canonical ensemble, the probability for the system to be in any particular microstate is

$$P(\boldsymbol{x}) = \frac{e^{-\beta H(\boldsymbol{x})}}{Z}, \tag{6.1}$$

where $\beta = 1/k_B T$ is the inverse temperature. The normalizing constant Z is the *partition function*,

$$Z = \sum_{\boldsymbol{x} \in \mathcal{X}} e^{-\beta H(\boldsymbol{x})}. \tag{6.2}$$

The partition function relates to most quantities of interest. For example, the internal (average) energy is

$$U = \sum_{\boldsymbol{x}} H(\boldsymbol{x}) P(\boldsymbol{x}) = -\frac{1}{Z}\frac{\partial Z}{\partial \beta}. \tag{6.3}$$

However, when $\boldsymbol{x}$ is high-dimensional, as it usually is, directly evaluating the sum (or integral) in Eq. (6.2) is practically impossible.

In statistics and machine learning, a quantity equivalent to Z frequently arises, and is used for rigorous model fitting and model selection. In this setting, a model is defined by a probability distribution, $P(\mathcal{D}|\boldsymbol{x})$, that corresponds to the probability of observing data, $\mathcal{D}$, given the value of some parameters, $\boldsymbol{x}$. To determine the parameters we apply Bayes' rule

$$P(\boldsymbol{x}|\mathcal{D}) = \frac{P(\mathcal{D}|\boldsymbol{x})P(\boldsymbol{x})}{P(\mathcal{D})}. \tag{6.4}$$

The denominator,

$$P(\mathcal{D}) = \sum_{\boldsymbol{x}} P(\mathcal{D}|\boldsymbol{x})P(\boldsymbol{x}), \tag{6.5}$$

is an important quantity known as the *marginal likelihood* or the *model evidence* [79, 128, 182]. Again, calculating the sum (or integral) in $P(\mathcal{D})$ is often highly nontrivial, exactly analogous to Z.[1] Safe in the knowledge that these problems are essentially similar, in this chapter we will use the language of statistical physics.

Closely related to the partition function is the *entropy*,

$$S = -\sum_{\boldsymbol{x}} P(\boldsymbol{x}) \ln P(\boldsymbol{x}). \tag{6.6}$$

Second only to energy, entropy is a central concept of statistical physics and is arguably just as fundamental—the concept of entropy transcends thermal physics and has been applied across science (for example [183–186]). However, for the same reasons that computing Z can be difficult, computing S can be also. For the canonical distribution of Eq. (6.1) we have the relation

$$\ln Z = S - \beta U, \tag{6.7}$$

and thus calculating both the entropy and energy, S and U, is at least as hard as calculating Z.

So far we have been general but vague about the supposed difficulties in calculating these quantities, Z, S, and U. Let's now explore this in more detail. The microstate, $\boldsymbol{x}$, encodes a detailed description of each relevant component in the system. Its entries $(x_1, x_2, \dots)$, for example, might encode the magnetic moments of atoms in a lattice. In general, each component i has an associated variable, x_i.

In the simplest case, components do not interact with one another and the energy (i.e. the Hamiltonian) is a sum over constituent parts. In this case, the

[1] Setting $-\beta H(\boldsymbol{x}) = \ln P(D, \boldsymbol{x})$ makes this equivalence obvious.

probability of Eq. (6.1) factorizes as

$$P(\boldsymbol{x}) = \frac{\prod_i e^{-\beta H_i(x_i)}}{\sum_{\boldsymbol{x}'} \prod_i e^{-\beta H_i(x_i{}')}} = \prod_i \frac{e^{-\beta H_i(x_i)}}{\sum_{x_i{}'} e^{-\beta H_i(x_i{}')}} = \prod_i \frac{e^{-\beta H_i(x_i)}}{Z_i} = \prod_i P(x_i) \tag{6.8}$$

where Z_i is a partition function for component i and $P(x_i)$ is the probability for component i to be in state x_i. This result simply states a well-known fact: the probability for independent events to co-occur is the product of their individual probabilities. From this factorization both S and U are trivial to compute,

$$S = -\sum_i \sum_{x_i} P(x_i) \ln P(x_i) \tag{6.9}$$

and

$$U = \sum_i \sum_{x_i} H_i(x_i) P(x_i). \tag{6.10}$$

The next simplest case, conceptually, is one in which components interact in pairs. A prototypical example is the Ising model, whose Hamiltonian is

$$H(\boldsymbol{\sigma}) = -\sum_{i,j} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i, \tag{6.11}$$

where each variable takes values $\sigma_i \in \{-1, +1\}$. The matrix J_{ij} encodes the interaction strength between component i and j, and h_i is an *external field* applied to i. Pairwise interactions naturally correspond to a network—we think of components as nodes, and $\boldsymbol{J}$ as the (potentially weighted) adjacency matrix. While it is conceptually simple, the introduction of pairwise interactions completely changes both the physics, and our ability to solve these models. Importantly, once interactions are present we generally lose the ability to factorize $P(\boldsymbol{x})$ in a convenient form.

Methods to solve the Ising model (and variants) have been in active development for a century [187, 188]. A lot of this work has focused on very stylized instances, for example assuming the network of interactions has a specific lattice structure. Typically, these methods rely on ingenious insights about the specific structure and symmetry of the problem they solve. Solutions are hard to come by and celebrated—even on a 2-dimensional square lattice the problem has only been rigorously solved when there is no external field.

In practice, a more useful approach for solving problems is to accept some level of approximation. *Mean-field* approximations would usually be the first port of call [128, 131, 180, 181]. These approximations replace certain variables in the Hamiltonian with their expected values, so that we can approximate the system by an ensemble of non-interacting components. This is generally well justified when components interact with many others, and the law of large numbers permits us to replace random quantities by their expected values. Thus, mean-field approximations should be good for problems associated with dense networks.

In the sparse case—where each node or component only has a handful of interactions—naive mean-field approximations often give substantially incorrect answers. A significant improvement can be made, however, by making use of a tree ansatz [181, 189, 190]. In this approach, one starts from the assumption that the network of interactions form a tree. One derives equations that would solve the problem *on a tree*, and then applies these same equations to the network of interest, regardless of whether the real network is a tree or not. This approach gives generally excellent results so long as there is not a high density of short cycles. Because of this assumed lack of short cycles, these approximations are referred to as "locally tree-like" approximations, and are usually considered exact on locally tree-like networks in the thermodynamic limit (i.e. $n \to \infty$).

Many problems of interest, however, sit at the awkward barrier between the domains of applicability of the mean-field and locally tree-like approximations. Networks are often sparse in the sense of having low average degree, and thus outside the realm of mean-field theory. Nevertheless, they are often dense with short loops (e.g. large numbers of triangles or squares [1, 191]), and thus outside the realm of locally tree-like approximations. In this chapter we shall address such cases head on, by developing the neighborhood formalism of Ch. 5. This will provide us with a sequence of increasingly sophisticated approximations, indexed by r. Formally, as $r \to \infty$ the equations will be exact on any network. However, for the common case of *globally sparse but locally dense* networks, the approximation may be highly accurate even for small values of r.

The structure for the rest of this chapter is as follows. In Sec. 6.1, we will expound the logic of the tree ansatz for factorizing distributions and computing Z. This approach reduces calculations to 1- and 2-point marginal distributions, and we will discuss how these marginal distributions can be efficiently evaluated using a technique known as belief propagation [127, 128, 192]. All of this covers well established ground. In Sec 6.2, however, we will consider the case of very non-tree like networks, which may arise due to loops of 2-point interactions, or due to higher order interactions. We will introduce formulas approximating Z in this case, and consider a belief propagation method for computing the required marginal distributions.[2] We will also discuss the connections to the *Kikuchi free energy* and *generalized belief propagation* [174, 193, 194], to which our approach is related.

6.1 Tree ansatz

Suppose we have some network of interactions, G, and each node $i \in G$ has an associated variable x_i. Let $\boldsymbol{x}_G$ denote a full assignment of these variables. The object of present interest, then, is the distribution $P(\boldsymbol{x}_G)$. The tree ansatz assumes we can factorize $P(\boldsymbol{x}_G)$ as

$$P(\boldsymbol{x}_G) = \frac{e^{-\beta H(\boldsymbol{x}_G)}}{Z} = \frac{\prod_{(i,j)\in G} P(x_i, x_j)}{\prod_{i\in G} P(x_i)^{d_i - 1}} \tag{6.12}$$

where $(i, j) \in G$ denotes the edges of G, $i \in G$ the nodes, and d_i the degree of node i. $P(x_i)$ and $P(x_i, x_j)$ are the one- and two-node marginal distributions. Why does Eq. (6.12) correspond to a tree ansatz? The reason is that such a form is correct on a tree.

To convince ourselves that Eq. (6.12) is correct on a tree we can prove the result inductively. Suppose that Eq. (6.12) is true for all trees with fewer than n nodes, and consider an arbitrary tree G with n nodes. For an arbitrarily chosen node i, it's straightforward to write

$$P(\boldsymbol{x}_G) = P(x_i)P(\boldsymbol{x}_{G\setminus i}|x_i) \tag{6.13}$$

where $G \setminus i$ is the network that arises once node i is removed from G. Since G was a tree, removing node i will split G into one or more subtrees, each with fewer than n nodes. Let $G_{i\to j}$ denote the sub-tree that contains node j. We can then write

$$\begin{aligned} P(\boldsymbol{x}_G) &= P(x_i) \prod_{j\in N_i} P(x_j|x_i)P(\boldsymbol{x}_{G_{i\to j}}|x_j) \\ &= P(x_i) \prod_{j\in N_i} \frac{P(x_i, x_j)}{P(x_i)} \frac{P(\boldsymbol{x}_{G_{i\to j}})}{P(x_j)} \end{aligned} \tag{6.14}$$

[2] Methods for the required belief propagations are being developed with Alec Kirkley and Mark Newman. These, along with other formal results Alec Kirkley is developing, are expected to appear in A. Kirkley, G. T. Cantwell, and M. E. J. Newman, Probabilistic models on networks with loops.

and since each $G_{i\to j}$ has fewer than n nodes, by the inductive hypothesis we have

$$\begin{aligned} P(\boldsymbol{x}_G) &= P(x_i) \prod_{j\in N_i} \frac{P(x_i, x_j)}{P(x_i)} \frac{\prod_{(k,l)\in G_{i\to j}} P(x_k, x_l)}{\prod_{k\in G_{i\to j}} P(x_k)^{d_k-1}} \\ &= \frac{\prod_{(i,j)\in G} P(x_i, x_j)}{\prod_{i\in G} P(x_i)^{d_i-1}}, \end{aligned} \tag{6.15}$$

as required. The base case, a tree with a single node, is trivial.

How does all this help us to compute the partition function Z, or the entropy, S? Assuming interactions are pairwise and Eq. (6.12) is valid, we have for the entropy

$$S = -\sum_{(i,j)\in G} \sum_{x_i,x_j} P(x_i, x_j) \ln P(x_i, x_j) + \sum_{i\in G} (d_i - 1) \sum_{x_i} P(x_i) \ln P(x_i) \tag{6.16}$$

and energy

$$U = \sum_{(i,j)\in G} \sum_{x_i,x_j} P(x_i, x_j) H(x_i, x_j) + \sum_{i\in G} \sum_{x_i} P(x_i) H(x_i) \tag{6.17}$$

where $H(x_i, x_j)$ are the interaction terms in $H(\boldsymbol{x}_G)$ between i and j, and $H(x_i)$ are the terms that only involve x_i. If we can calculate the one- and two-point marginal distributions (by any method), both the entropy, S, and the energy, U, are simple to evaluate, and hence also the partition function. In the next subsection we discuss how these marginal distributions can be derived using *belief propagation*.

6.1.1 Evaluating one- and two-point marginal distributions with belief propagation

By definition, the one-point marginal is

$$P(x_i) = \frac{1}{Z} \sum_{\boldsymbol{x}_{G\setminus i}} e^{-\beta H(\boldsymbol{x}_G)} \tag{6.18}$$

and because the network is a tree the terms factorize

$$P(x_i) = \frac{1}{Z} e^{-\beta H(x_i)} \prod_{j\in N_i} \sum_{\boldsymbol{x}_{G_{i\to j}}} \left(e^{-\beta H(x_i,x_j)} e^{-\beta H(\boldsymbol{x}_{G_{i\to j}})} \right). \tag{6.19}$$

To evaluate $P(x_i)$, we only need its value up to a constant factor, since we can easily enforce the normalization condition $\sum_{x_i} P(x_i) = 1$. Noting this we can write

$$P(x_i) \propto e^{-\beta H(x_i)} \prod_{j \in N_i} \sum_{x_j} e^{-\beta H(x_i, x_j)} q_{j \to i}(x_j) \tag{6.20}$$

where by definition,

$$q_{j \to i}(x_j) = \frac{1}{Z_{j \to i}} \sum_{\boldsymbol{x}_{G_{i \to j}} \setminus j} e^{-\beta H(\boldsymbol{x}_{G_{i \to j}})} \tag{6.21}$$

is the probability for node j to be in state x_j in the system composed of the subtree $G_{i \to j}$. Since this is again a one-node marginal in a tree, we can use the same argument to evaluate $q_{j \to i}(x_j)$, and get

$$q_{j \to i}(x_j) \propto e^{-\beta H(x_j)} \prod_{k \in N_j \setminus i} \sum_{x_k} e^{-\beta H(x_j, x_k)} q_{k \to j}(x_k). \tag{6.22}$$

This provides a set of self-consistent equations, whose solutions provide us the necessary quantities to evaluate the one-node marginals.

Solving the equations of Eq. (6.22) might seem daunting. If the network has m edges then we have a system of $2m$ nonlinear equations. In practice, however, solving these equations numerically is straightforward. First, one makes an initial guess, for example set each $q_{j \to i}(x_j)$ to a random value. Then, one simply iteratively updates the q's using Eq. (6.22) until the whole system converges to a fixed point.

Once we have the q's, the two-point marginals are also simple to compute. A quick calculation establishes

$$P(x_i, x_j) \propto e^{-\beta H(x_i, x_j)} q_{i \to j}(x_i) q_{j \to i}(x_j). \tag{6.23}$$

These equations are exactly correct for trees, and the iterative update algorithm to solve Eq. (6.22) is known as belief propagation. Although it is not strictly correct, it will usually give good results for non-trees without short cycles [127,128]. Perhaps this is as good a justification as needed: the approximations are justified because they work.

A more theoretical justification for the approach relies on a variational argument [174,193]. If we cannot compute $P(\boldsymbol{x}_G)$ exactly, we can try to approximate it by some function in the form in Eq. (6.12),

$$Q(\boldsymbol{x}_G) = \frac{\prod_{(i,j)\in G} Q(x_i, x_j)}{\prod_{i\in G} Q(x_i)^{d_i-1}}. \tag{6.24}$$

Minimizing the Kullback–Leibler divergence between Q and P, while enforcing $\sum_{x_j} Q(x_i, x_j) = Q(x_i)$ and $\sum_{x_i} Q(x_i) = 1$, leads to a solution for Q that matches the belief propagation. However, since this procedure does not enforce the normalization of $Q(\boldsymbol{x}_G)$ the argument is not fully rigorous.

6.1.2 Example: the Ising model

For the Ising model Eq. (6.22) becomes

$$q_{j\to i}(\sigma_j) \propto e^{\beta h\sigma_j} \prod_{k\in N_j\setminus i} \left(e^{-\beta J\sigma_j} q_{k\to j}(-1) + e^{\beta J\sigma_j} q_{k\to j}(+1)\right) \tag{6.25}$$

and the normalization condition is

$$q_{j\to i}(+1) + q_{j\to i}(-1) = 1. \tag{6.26}$$

To demonstrate the accuracy of this approach we compare results based on these equations to Monte Carlo simulations in Fig. 6.1. The first example network contains a low density of short loops and the tree ansatz is clearly in excellent agreement with the full simulations. However, the second example network has a more realistic structure and the tree ansatz is fairly inaccurate. We rectify this in the next section.

6.2 Neighborhood ansatz

Let us turn our attention to the neighborhood approach of Ch. 5, in order to systematically improve upon the tree ansatz. Just as the locally tree-like approach began with a factorization, we do the same again here. Assuming that the network has no primitive cycles longer than $r + 2$, we can factorize $P(\boldsymbol{x}_G)$ as

$$P(\boldsymbol{x}_G) = \frac{\prod_{i\in G} P(\boldsymbol{x}_{N_i^{(r)}})}{\prod_{(i,j)^{(r)}\in G} P(\boldsymbol{x}_{\cap_{ij}^{(r)}})^{2/|\cap_{ij}^{(r)}|}} \tag{6.27}$$

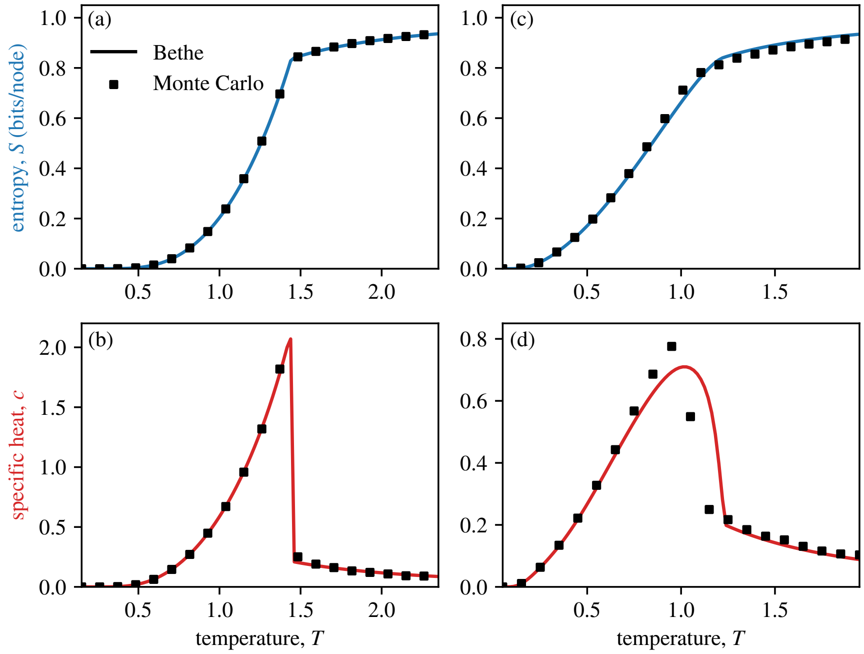


Figure 6.1: Bethe ansatz for the Ising model. Comparison of Eq. (6.25)—denoted “Bethe”—to Monte Carlo simulations. We show results for two different networks. Panel (a) and (b) show entropy and specific heat for a network with a low density of short cycles. This network has 10 000 nodes and its clustering co-efficient [1]—a measure of the density of short loops—is only 3.5×10^{-4}. Panel (c) and (d) show a network with a more realistic structure (courtesy of Alec Kirkley). This network has $9,447$ nodes and a clustering co-efficient of 0.103, in line with real-world networks [1]. For the Bethe solution, we estimate the entropy with Eq. (6.16) and the specific heat from the first derivative of Eq. (6.17). Derivatives were computed automatically using standard software [195]. For the network with a low density of short cycles—panel (a) and (b)—we see excellent quantitative agreement. However, with a more realistic structure—panel (c) and (d)—we see significant errors. While the entropy shows decent qualitative agreement, its derivatives (closely related to heat capacity) are clearly wrong. Standard Monte Carlo methods were used for the simulations [122, 123].

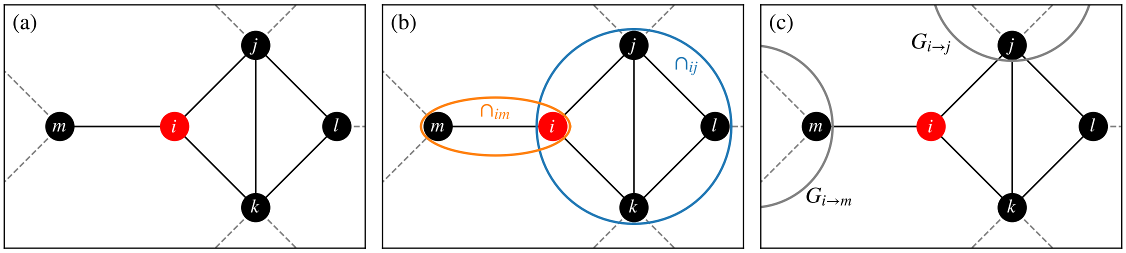


Figure 6.2: Neighborhoods and various related quantities for node i in an example network. For this example we assume that $r = 2$ is sufficient to capture all primitive cycles and thus calculations at $r = 2$ are exact. In (a) we show the neighborhood, $N_i = N_i^{(2)}$, which contains the edges and nodes shown in solid black. In (b) we show the two distinct intersections at node i. We show $\cap_{im} = N_i \cap N_m$ and $\cap_{ij} = N_i \cap N_j$. Note that all intersections between nodes in $\cap_{ij}$ are identical. In this particular example we have $\cap_{ij} = \cap_{ik} = \cap_{il} = \cap_{jk} = \cap_{jl} = \cap_{lk}$. In (c) we indicate the graphs $G_{i\to j}$ and $G_{i\to m}$, two of the disconnected components formed when the edges of N_i are removed.

where $\cap_{ij}^{(r)} = N_i^{(r)} \cap N_j^{(r)}$ and $(i,j)^{(r)}$ are pairs of nodes that are contained in each other's r-neighborhood, i.e. nodes i and j such that $i \in N_j^{(r)}$ and $j \in N_i^{(r)}$. The marginal distribution $P(\boldsymbol{x}_{N_i^{(r)}})$ is the marginal distribution for all variables in the neighborhood of node i. For convenience, in the following calculations we shall assume the neighborhood size r has already been chosen correctly and frequently drop it from the notation, hence $N_i^{(r)} = N_i$.

Before proving Eq. (6.27), it's instructive to study Fig. 6.2, which illustrates the relevant sets. In particular, as demonstrated in panel (b), many of the intersections $\cap_{ij}$ will be equivalent. In fact, for any pair $k, l \in \cap_{ij}$ we have $\cap_{ij} = \cap_{kl}$. As a result of this, we can write

$$P(\boldsymbol{x}_{\cap_{ij}}) = \prod_{(k,l)\in\cap_{ij}} P(\boldsymbol{x}_{\cap_{ij}})^{1/\binom{|\cap_{ij}|}{2}} = \prod_{(k,l)\in\cap_{ij}} P(\boldsymbol{x}_{\cap_{kl}})^{1/\binom{|\cap_{kl}|}{2}} \tag{6.28}$$

where the product is over all $\binom{|\cap_{ij}|}{2}$ pairs $(k,l) \in \cap_{ij}$.

A proof of Eq. (6.27) can be achieved with the same inductive logic as before. Assume that the formula is correct for all networks with fewer than n nodes and no primitive cycles longer than $r + 2$. If G is a network with n nodes and no primitive cycles longer than $r + 2$ then

$$\begin{aligned} P(\boldsymbol{x}_G) &= P(\boldsymbol{x}_{N_i}) \prod_{j \in N_i} P(\boldsymbol{x}_{N_j} | \boldsymbol{x}_{N_i}) P(\boldsymbol{x}_{G_{i \to j}} | \boldsymbol{x}_{N_j}) \\ &= P(\boldsymbol{x}_{N_i}) \prod_{j \in N_i} \frac{P(\boldsymbol{x}_{N_j})}{P(\boldsymbol{x}_{\cap_{ij}})} P(\boldsymbol{x}_{G_{i \to j}} | \boldsymbol{x}_{N_j \setminus N_i}), \end{aligned} \tag{6.29}$$

where the sub-graphs $G_{i \to j}$ are the resulting networks once the edges of N_i have been removed (see Fig. 6.2). Since the subgraphs $G_{i \to j}$ have fewer than n nodes and no primitive cycles longer than $r + 2$, we can use the inductive hypothesis and Eq. (6.28) to arrive at

$$\begin{aligned} P(\boldsymbol{x}_G) &= P(\boldsymbol{x}_{N_i}) \prod_{j \in N_i} \frac{1}{\prod_{(k,l) \in \cap_{ij}} P(\boldsymbol{x}_{\cap_{kl}})^{1/\binom{|\cap_{kl}|}{2}}} \left(\frac{\prod_{k \in G_{i \to j}} P(\boldsymbol{x}_{N_k})}{\prod_{(k,l)^{(r)} \in G_{i \to j}} P(\boldsymbol{x}_{\cap_{kl}})^{2/|\cap^{(r)}_{kl}|}} \right) \\ &= \frac{\prod_{i \in G} P(\boldsymbol{x}_{N_i^{(r)}})}{\prod_{(i,j)^{(r)} \in G} P(\boldsymbol{x}_{\cap_{ij}^{(r)}})^{2/|\cap_{ij}^{(r)}|}} \end{aligned} \tag{6.30}$$

as required. The base case is again trivial.

This factorization, Eq. (6.27), is correct whenever the network does not have primitive cycles longer than $r + 2$. It can be further simplified by noting that

$$\begin{aligned} P(\boldsymbol{x}_{N_i^{(r)}}) &= P(x_i) \prod_{j \in N_i^{(r)}} P(\boldsymbol{x}_{\cap_{ij}^{(r)}} | x_i)^{\frac{1}{|\cap_{ij}^{(r)}| - 1}} \\ &= P(x_i) \prod_{j \in N_i^{(r)}} \left(\frac{P(\boldsymbol{x}_{\cap_{ij}^{(r)}})}{P(x_i)} \right)^{\frac{1}{|\cap_{ij}^{(r)}| - 1}}. \end{aligned} \tag{6.31}$$

Defining the quantities

$$W_{ij}^{(r)} = 1 - \sum_{(l,m)^{(r)} \in G} \frac{1}{\binom{|\cap_{lm}^{(r)}|}{2}} \mathbf{1}_{\left\{(i,j) \in \cap_{lm}^{(r)}\right\}} \tag{6.32}$$

with $\mathbf{1}_{\{\ldots\}}$ being the indicator function and

$$C_i^{(r)} = 1 - \left(\sum_{j \in N_i^{(r)}} \frac{1}{| \cap_{ij}^{(r)} | - 1} \right) - \left(\sum_{j \in N_i^{(0)}} W_{ij}^{(r)} \right), \tag{6.33}$$

insertion of Eq. (6.31) into Eq. (6.27) yields

$$P(\boldsymbol{x}_G) = \left(\prod_{(i,j)^{(r)}\in G} P(\boldsymbol{x}_{\cap_{ij}^{(r)}})^{1/\binom{|\cap_{ij}^{(r)}|}{2}}\right)\left(\prod_{(i,j)\in G} P(x_i, x_j)^{W_{ij}^{(r)}}\right)\left(\prod_{i\in G} P(x_i)^{C_i^{(r)}}\right). \qquad (6.34)$$

Equation (6.34) will be exactly correct when there are no primitive cycles longer than $r+2$. In this case, all $W_{ij}^{(r)}$ will be zero. When the factorization is not exact—i.e. when there are primitive cycles we have not accounted for—then the $P(x_i, x_j)^{W_{ij}}$ terms ensure each edge gets correctly weighted in the factorization. These equations are identical to the conventional tree approximation when $r = 0$.

Just as before, if we know the appropriate marginal distributions we can compute S and U from the factorization, and thus also evaluate Z. For the entropy, S, we have

$$S = \sum_{(i,j)^{(r)}\in G} \frac{1}{\binom{|\cap_{ij}^{(r)}|}{2}}\Big\langle -\ln P(\boldsymbol{x}_{\cap_{ij}^{(r)}})\Big\rangle + \sum_{(i,j)\in G} W_{ij}^{(r)}\Big\langle -\ln P(x_i, x_j)\Big\rangle + \sum_{i\in G} C_i^{(r)}\Big\langle -\ln P(x_i)\Big\rangle \qquad (6.35)$$

and for the energy,

$$U = \sum_{(i,j)^{(r)}\in G} \frac{1}{\binom{|\cap_{ij}^{(r)}|}{2}}\Big\langle H(\boldsymbol{x}_{\cap_{ij}^{(r)}})\Big\rangle + \sum_{(i,j)\in G} W_{ij}^{(r)}\Big\langle H(x_i, x_j)\Big\rangle + \sum_{i\in G} C_i^{(r)}\Big\langle H(x_i)\Big\rangle. \qquad (6.36)$$

A belief propagation can be used to evaluate the required marginal distributions, as we describe in the next section.

6.2.1 Evaluating the neighborhood marginal distributions with belief propagation

Assuming that the network does not contain long primitive cycles, removing the neighborhood of node i will split the network into one or more sub-networks. As before, we can thus trace over the degrees of freedom in each of these sub-networks to arrive at

$$P(x_i) \propto e^{-\beta H(x_i)} \sum_{\boldsymbol{x}_{N_i}} \prod_{j\in N_i} e^{-\beta H(x_i, x_j)} q_{j\to i}(x_j) \qquad (6.37)$$

where

$$q_{j\to i}(x_j) = \frac{1}{Z_{j\to i}} \sum_{\boldsymbol{x}_{G_{i\to j}\setminus j}} e^{-\beta H(\boldsymbol{x}_{G_{i\to j}})}. \qquad (6.38)$$

And as before, this quantity corresponds to the probability that node j would take the value x_j in the system defined by $G_{i\to j}$. The same argument demonstrates

$$q_{j\to i}(x_j) \propto e^{-\beta H(x_j)} \sum_{\boldsymbol{x}_{N_j\setminus N_i}} \prod_{k\in N_j\setminus N_i} e^{-\beta H(x_j,x_k)} q_{k\to j}(x_k) \tag{6.39}$$

which defines a complete set of self-consistent equations. Neighborhood marginals can be evaluated with

$$P(\boldsymbol{x}_{N_i}) \propto e^{-\beta(H(x_i)+H(\boldsymbol{x}_{N_i}))} \prod_{j\in N_i} q_{j\to i}(x_j) \tag{6.40}$$

and the intersection marginals can be found from

$$P(\boldsymbol{x}_{\cap_{ij}}) \propto e^{-\beta H(\boldsymbol{x}_{\cap_{ij}})} q_{i\to j}(x_i) \prod_{k\in\cap_{ij}\setminus i} q_{k\to i}(x_k). \tag{6.41}$$

In the next sub-section we again turn to the Ising model to explore the accuracy of these approximations.

6.2.2 Example: the Ising model

For the Ising model, the equations we must solve are

$$q_{j\to i}(\sigma_j) \propto e^{\beta h\sigma_j} \sum_{\sigma_{N_j\setminus N_i}} \prod_{k\in N_j\setminus N_i} e^{\beta J\sigma_j\sigma_k} q_{k\to j}(\sigma_k), \tag{6.42}$$

and again

$$q_{j\to i}(+1) + q_{j\to i}(-1) = 1. \tag{6.43}$$

Figure 6.3 compares the solutions to these equations to the results of full Monte Carlo simulations.

So long as the neighborhoods are not too large, the sum in Eq. (6.42) can be evaluated directly. Unfortunately, if the neighborhoods are even moderately sized (larger than around 10 nodes) then even this sum may be too costly to evaluate. Although we will not consider such cases here, future work should address methods for approximating the sum, for example using Monte Carlo methods like we did for percolation in Ch. 5.

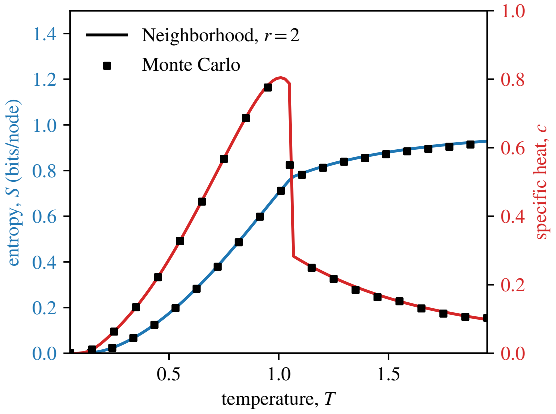


Figure 6.3: Neighborhood ansatz for the Ising model. Comparison of the $r = 2$ neighborhood calculation of Eq. (6.42) with Monte Carlo simulations. The example network is identical to Fig. 6.1(c) and (d), and has a realistically high density of short cycles. Equation (6.42) is iterated to convergence, and then the values are used to compute the entropy from Eq. (6.35) and specific heat from the derivative of Eq. (6.36). Again, standard software is used for derivatives [195] and standard Monte Carlo techniques are used for the simulations [122, 123]. We see excellent agreement between the equation based results and the simulations.

6.3 Discussion

We have considered factorizations of probability distributions, based on r-neighborhoods. From these factorizations we are able to compute quantities such as the partition function or entropy of a model.

Our approach is broadly applicable and will be most useful for complex, heterogeneous networks. It does not explicitly rely on any particular symmetry or structure of the problem at hand and so in principle can be applied to arbitrary problems. Of course, this generality has a downside: any highly symmetric and regular problem can probably be significantly simplified, which our approach will not do automatically.

While the method is formally correct for any network with $r \to \infty$, it is only of practical use for small values of r, say $r \leq 4$. For the approximations to be accurate at small r we require the problem to have a specific structure. For $s > r$ we require either sufficiently few primitive cycles of length $s + 2$, or correlations along paths of length s to be neglectable.

The scheme we have described is closely related to a general framework, known as region based approximations [174, 193]. In this approach one first defines *basic regions* of the network, and then constructs a *region graph* (see [174]). The general framework, however, does not come with a prescription by which to choose the basic regions in complex networks.

One method by which region based approximations can be constructed is the Kikuchi cluster variation method [174, 194]. In the Kikuchi framework, the tree ansatz (Bethe free energy) corresponds to picking basic regions that contain two connected nodes—every pair of connected nodes form a region. The obvious refinement to this is to enlarge basic regions to each contain three connected nodes, or four nodes, or so forth. Applying such a prescription to complex networks, however, is not necessarily wise. In networks with large numbers of short loops, the number of equations one needs to solve will grow exponentially and many regions will substantially overlap with many others.

The neighborhood framework we've discussed can be interpreted as a reasonable prescription to choose regions in complex networks. On networks without long primitive cycles, one should take the intersections of neighborhoods to form basic regions. Then, the general formulation for the Kikuchi free energy provides an equivalent expression to the one we derive. However, in most real scenarios—when the neighborhood ansatz is only an approximation—there are deviations between our formula and the Kikuchi free energy. In this case intersections will have non-trivial overlaps and directly inserting these regions into the Kikuchi free energy would lead to significant complications, with new equations for the intersections of intersections, and the intersections of the intersections of intersections, and so forth.

Our general approach should work well on clustered, complex networks, which are dense with short cycles but do not otherwise appear to have lattice like structure. Applying the approach in practice, however, may take some finesse. The key equation, Eq. (6.39), may be computationally expensive to evaluate, yet the utility of the method relies critically on fast evaluation of this quantity. Still, we only need to be able to evaluate these equations quickly on neighborhood-sized networks. Thus, even if we employ a naive (exponentially slow) algorithm, one full iteration of the belief propagation will still scale linearly with the number of nodes in the network.

Future work should be focused on two open problems. First, for real-world networks we should establish the relation between the approximation level r and the accuracy of the calculations. Our hope is that the accuracy of these approximations increases monotonically, but we have only provided some heuristic justification for this and the assumption may be substantially in error. Second, we should explore efficient schemes for evaluating or approximating the message equations, Eq. (6.39). In principle, this is a fairly straightforward problem. All we need is an algorithm that is fast on small networks—we need not worry about whether it scales efficiently. In practice, of course, highly optimizing calculations takes considerable engineering prowess.

CHAPTER 7

Conclusion

In this thesis we have critically considered the role of correlation in complex networks. We have approached the topic from multiple angles.

In Chs. 2 and 3 we noted that the behavior of individual nodes is often considerably more consistent than conventional independence assumptions assume. Within our everyday lives, this observation is practically trivial. If all of your previous romantic partners were men, it is not a tremendous insight to note that, statistically speaking, the next one is also likely to be a man. As trite as this "insight" may seem, common network tools fail to account for this phenomenon. In Ch. 2 we proposed measures to accurately characterize these effects, and in Ch. 3 we explained how this mode of thinking improves data recovery and community detection in networks.

In Ch. 4 we considered the bare effects of correlated edge data. Assuming that networks represent some underlying set of relations, and that the strength of these relations are correlated, we find that several commonly observed properties of complex networks naturally emerge. As a result, the ubiquity of these properties should not be surprising—they are a natural consequence of correlation.

In Chs. 5 and 6 we shifted focus away from the effect of correlation on network structure. Instead, we considered the network structure to be fixed and given and studied the effect of this structure on network calculations. If networks are clustered and loopy—as they almost always are—the independence assumptions of standard message passing techniques break down. We considered a procedure that accounts for correlations due to network structure and applied it to three examples: percolation, the eigenvalue spectrum of sparse matrices, and partition functions.

Each chapter of this thesis concluded with a discussion of extensions to the results therein. We shall not repeat those remarks here. Rather, it is now appropriate to consider a more holistic view.

This thesis began with some general thoughts as to why we should study networks at all. A key point that I hope to have communicated, both in the introduction and throughout the thesis, is that analysis of "who is connected to whom?" (and equally, who is *not* connected to whom) leads to an intricate and unique web of structure. Accepting that the particulars of structure matter puts us in an awkward position. While training in theoretical physics exhorts one to strive for “theory”, any theory of networks must abate two opposing forces. First, it must be general enough to account for the particularities of any real-world situation. Second, it must be specific enough to entail real consequences. As things stand, I find it difficult to even picture what such a general theory would look like, at least “theory” in the physicist’s sense. A better prospect, perhaps, is to look for a framework—a mathematical formalism that is able to fit arbitrary data once problem specific assumptions are added.

On the face of it, exponential random graphs [91,196] appear to be a reasonable candidate for such a general network framework. They are able to account for arbitrary patterns and can be tested directly against data. Currently, however, their significant technical problems [93–95] seem like an insurmountable barrier. Exchangeable random graphs [149, 150] are an alternative candidate. They also face significant technical issues, but there has been recent progress [197]. Still, exchangeability is a somewhat contrived theoretical property, closely related to the assumption of “independent and identically distributed” data [150]. Whether exchangeability assumptions can really make sense of the rich interdependence of real-world complex systems is an open question. Future work should consider the suitability of exponential random graphs, exchangeable random graphs, or any other attractive options for a general framework.

Perhaps there won’t be a fully general and well justified framework for networks, but neither is this a necessary prerequisite for their utility. The important point is that for networks to achieve a broad and lasting impact, a tighter connection between theory and data is necessary. To date, a significant portion of the field has concerned itself with modeling the structure and formation of networks but such work is usually detached from serious empirical experimentation and is instead more concerned with the mathematical properties of the models. In stark contrast, another sub-field is at essentially the opposite end of the spectrum, developing networks as a practical tool for data analysis. Insights gained from studying simple rules for constructing complex networks are often difficult to apply directly to data and for all their influence on the field, it is still not really clear how to directly apply the work of Barabási and Albert [40] or Watts and Strogatz [41]. Conversely, practical methods for specific data analysis tasks do little for general understanding. Brin and Page’s PageRank [52] is a useful innovation, but as scientists, what do we learn from it?

As crude as graphs may be for representing real-world complex systems, they turn out to be mathematically subtle. Even correctly accounting for triangles is no simple task, and in the short term I believe triangles should remain a significant area of focus. However we move forward, we must be comfortable with the fact that by their very nature, networks are strongly interdependent objects. Contributing to a general ease and familiarity with correlation and interdependence has been a goal of this thesis, but the project is far from complete.

APPENDIX A

Estimating Mixing Parameters

A.1 Point estimates for α

The maximum likelihood estimate for $\boldsymbol{\alpha}_r$ is given by the location of the maximum of

$$L_r(\boldsymbol{\alpha}_r) = \sum_{i \in r} \left[\ln B(\boldsymbol{\alpha}_r + \boldsymbol{k}_i) - \ln B(\boldsymbol{\alpha}_r) \right]. \tag{A.1}$$

Here $\ln B(\boldsymbol{x})$ is the log of the multivariate beta function,

$$\ln B(\boldsymbol{x}) = -\ln \Gamma(x_\Sigma) + \sum_s \ln \Gamma(x_s), \tag{A.2}$$

with $x_\Sigma = \sum_s x_s$. Both the Jacobian and Hessian of L_r are straightforward to compute, so in principle one could perform the maximization using optimizers such as Newton's method that require second derivatives.

There are however some technical complications with direct maximization of (A.1). First, one must impose the constraint $\alpha_{rs} > 0$, which can be done by re-parameterizing with $y_{rs} = \ln \alpha_{rs}$ and writing

$$L_r(\boldsymbol{y}_r) = \sum_{i \in r} \left[\ln B(e^{\boldsymbol{y}_r} + \boldsymbol{k}_i) - \ln B(e^{\boldsymbol{y}_r}) \right]. \tag{A.3}$$

An unconstrained maximization with respect to $\boldsymbol{y}_r$ then achieves the desired goal.

Second, and more important, under some circumstances the maximum is not guaranteed to exist and L_r can increase as $y_{rs} \to \pm\infty$. For a well-defined estimate we must insist on a maximum at a finite value of y_{rs}. A simple way to do this is to add a quadratic regularization term to the likelihood thus:

$$L_r(\boldsymbol{y}_r) = \sum_{i\in r} \bigl[\ln B(e^{\boldsymbol{y}_r} + \boldsymbol{k}_i) - \ln B(e^{\boldsymbol{y}_r})\bigr] - \lambda \sum_s y_{rs}^2, \tag{A.4}$$

where λ is a small positive constant.

From a Bayesian perspective this quadratic regularization corresponds to placing a normal prior on y_{rs} with mean zero and variance $(2\lambda)^{-1}$, or equivalently a log-normal prior on α_{rs}. As $\lambda \to 0$ the prior on y_{rs} becomes uniform, so any small fixed value of λ should give acceptable results. We use $\lambda = 2^{-7}$, equivalent to $\sigma = 8$, which implies that α_{rs} falls roughly between the 3σ bounds 10^{-10} and 10^{10}.

To find the maximum of Eq. (A.4) one can use any numerical optimization technique. For techniques that make use of the Jacobian and/or Hessian, the Jacobian is given by

$$\frac{\partial L_r}{\partial y_{rs}} = e^{y_{rs}} \sum_{i\in r} \Bigl[\psi(e^{y_{rs}} + k_{is}) - \psi\bigl(\textstyle\sum_t e^{y_{rt}} + k_i\bigr) - \psi(e^{y_{rs}}) + \psi\bigl(\textstyle\sum_t e^{y_{rt}}\bigr)\Bigr] - 2\lambda y_{rs}, \tag{A.5}$$

where $\psi(x) = \Gamma'(x)/\Gamma(x)$ is the so-called digamma function. The Hessian is given by

$$\begin{aligned}\frac{\partial^2 L_r}{\partial y_{rs}^2} &= e^{y_{rs}} \frac{\partial L_r}{\partial y_{rs}} + e^{2y_{rs}} \sum_{i\in r} \Bigl[\psi'(e^{y_{rs}} + k_{is}) - \psi'\bigl(\textstyle\sum_t e^{y_{rt}} + k_i\bigr) \\ &\qquad\qquad - \psi'(e^{y_{rs}}) + \psi'\bigl(\textstyle\sum_t e^{y_{rt}}\bigr)\Bigr] - 2\lambda, \\ \frac{\partial^2 L_r}{\partial y_{rs}\partial y_{rt}} &= e^{y_{rs}+y_{rt}} \sum_{i\in r} \bigl[\psi'\bigl(\textstyle\sum_t e^{y_{rt}}\bigr) - \psi'\bigl(\textstyle\sum_t e^{y_{rt}} + k_i\bigr)\bigr],\end{aligned} \tag{A.6}$$

where $\psi'(x)$ is the trigamma function.

A.2 Bayesian estimates for R and V

To compute an estimate of any quantity that depends on α, we can average its value over the posterior distribution. For any function $f(\alpha)$ the average is given by

$$\langle f \rangle = \int f(\alpha) P(\alpha|A, g)\, d\alpha, \tag{A.7}$$

which can also be written

$$\langle f \rangle = \frac{\int f(y) \exp\left[\sum_r L_r(\boldsymbol{y}_r)\right] dy}{\int \exp\left[\sum_r L_r(\boldsymbol{y}_r)\right] dy}, \tag{A.8}$$

where $y_{rs} = \ln \alpha_{rs}$ and $L_r(\boldsymbol{y}_r)$ is defined by Eq. (A.4).

Both R and V, as we have defined them, are averages over the groups, $R = \sum_r p_r R_r$ and $V = \sum_r p_r V_r$. For any such function we can compute the averages for the individual groups separately

$$\langle F \rangle = \sum_r p_r \langle F_r \rangle = \sum_r p_r \frac{\int F_r(\boldsymbol{y}) \exp\left[L_r(\boldsymbol{y})\right] d\boldsymbol{y}}{\int \exp\left[L_r(\boldsymbol{y})\right] d\boldsymbol{y}}. \tag{A.9}$$

Integrals of this form can be approximated using Laplace's method, which in this case gives

$$\langle F_r \rangle \simeq \sqrt{\frac{\det \boldsymbol{\Sigma}_r^*}{\det \boldsymbol{\Sigma}_r}} \exp\left[L_r^*(\hat{\boldsymbol{y}}_r^*) - L_r(\hat{\boldsymbol{y}}_r)\right], \tag{A.10}$$

where

$$L_r^*(\boldsymbol{y}) = L_r(\boldsymbol{y}) + \ln F_r(\boldsymbol{y}), \tag{A.11}$$

$$\hat{\boldsymbol{y}}_r = \arg\max_{\boldsymbol{y}} \left\{L_r(\boldsymbol{y})\right\}, \tag{A.12}$$

$$\hat{\boldsymbol{y}}_r^* = \arg\max_{\boldsymbol{y}} \left\{L_r^*(\boldsymbol{y})\right\}, \tag{A.13}$$

and $\boldsymbol{\Sigma}_r^*$ and $\boldsymbol{\Sigma}_r$ are minus the inverse of the Hessians of L_r^* and L_r at $\hat{\boldsymbol{y}}_r^*$ and $\hat{\boldsymbol{y}}_r$. In this ratio form some errors cancel and Laplace's approximation has only an $O(n^{-2})$ error [198].

Estimates for R and V can now be computed from Eqs. (A.9) and (A.10) with

$$F^{(R)} = \sum_r p_r \frac{e^{y_{rr}}}{\sum_s e^{y_{rs}}}, \tag{A.14}$$

$$F^{(V)} = \sum_r p_r \frac{1}{1 + \sum_s e^{y_{rs}}}. \tag{A.15}$$

The values of $\hat{\boldsymbol{y}}_r$ and $\hat{\boldsymbol{y}}_r^*$ along with the Hessians can be computed from Eqs. (A.5) and (A.6). Error estimates can also be computed from estimates of R^2 and V^2.

Software to compute estimates of R and V is available [199].

APPENDIX B

Technical Details for Normal Distributions

B.1 Multivariate normal integrals and Hermite polynomials

The probability of a two-star existing with nodes i, j and k as constituents is given by

$$P[X_{ij}, X_{ik} \geq t] = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_t^\infty \int_t^\infty e^{-\frac{1}{2}\left(\frac{x^2-2\rho xy+y^2}{1-\rho^2}\right)} dxdy. \tag{B.1}$$

Direct computation of the integral is not straightforward but we can compute it quickly using the Hermite polynomials [152]. A quick outline of this method: for $n \geq 0$, define the Hermite polynomials as

$$H_n(x) = (-1)^n e^{\frac{x^2}{2}} \frac{d^n}{dx^n} e^{-\frac{x^2}{2}}. \tag{B.2}$$

As the name suggests, the Hermite polynomials are in fact polynomials, for example $H_0(x) = 1$, $H_1(x) = x$, $H_2(x) = x^2 - 1$, and so on. For notational convenience also define

$$H_{-1}(x) = \frac{1 - \Phi(x)}{\phi(x)}. \tag{B.3}$$

Using the Hermite polynomials, we can expand Eq. (B.1) as an infinite sum and integrate term by term. The final result is given by Eq. (4.15). The same trick is used for the 3-dimensional integral to give Eq. (4.16).

B.2 Degree distribution

Since, by assumption, all nodes in this model are equivalent, we will simply consider the one-node marginal to compute the degree distribution. Let $\boldsymbol{U}$ be all the terms in $\boldsymbol{X}$ that are associated with node 0, i.e. $U_j = X_{0j}$. Then, $\boldsymbol{U}$ is multivariate normally distributed, $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}^{(0)})$, where $\boldsymbol{\Sigma}^{(0)}$ has ones along the diagonal and ρ everywhere else

$$\Sigma^{(0)}_{jk} = \Sigma_{(0,j),(0,k)} = \begin{cases} 1 & \text{for } j = k, \\ \rho & \text{otherwise.} \end{cases}$$

The focal node will have degree k when exactly k terms in $\boldsymbol{U}$ are larger than the threshold t. There are $\binom{n-1}{k}$ different ways this can happen and each is equally likely. So, to compute p_k we can compute the probability that the first k terms in $\boldsymbol{U}$ are larger than t and all others are smaller, and then multiply by $\binom{n-1}{k}$ to obtain

$$p_k = \binom{n-1}{k} P\left[U_1, \ldots, U_k \geq t; U_{k+1}, \ldots U_{n-1} < t\right]. \tag{B.4}$$

To solve this integral we use a standard trick [200]. First, we note that if $Z_0, Z_1, \ldots,$ Z_{n-1} are i.i.d. $\mathcal{N}(0, 1)$ then

$$((\sqrt{1-\rho}Z_1 + \sqrt{\rho}Z_0), \ldots, (\sqrt{1-\rho}Z_{n-1} + \sqrt{\rho}Z_0))^T \tag{B.5}$$

will be distributed identically to $\boldsymbol{U}$. Further, once we know the value of Z_0 then all the terms are independent, and the probability that any one of them is greater than t is the probability that $Z_1 \geq \frac{t-\sqrt{\rho}z}{\sqrt{1-\rho}}$. Given $Z_0 = z$, the probability that exactly k values will be greater than t and the rest less than t is

$$\binom{n-1}{k} \left[1 - \Phi\left(\frac{t-\sqrt{\rho}z}{\sqrt{1-\rho}}\right)\right]^k \Phi\left(\frac{t-\sqrt{\rho}z}{\sqrt{1-\rho}}\right)^{n-1-k}. \tag{B.6}$$

Averaging this quantity over z then provides us with the correct expression,

$$p_k = \binom{n-1}{k} \underbrace{\int_{-\infty}^{+\infty} \left[1 - \Phi\left(\frac{t-\sqrt{\rho}z}{\sqrt{1-\rho}}\right)\right]^k \Phi\left(\frac{t-\sqrt{\rho}z}{\sqrt{1-\rho}}\right)^{n-1-k} \phi(z) dz}_{=I_{n,k}} \tag{B.7}$$

where $I_{n,k}$ is the integral. A change of variables allows us to write

$$I_{n,k} = \sqrt{\frac{1-\rho}{2\pi\rho}} \int_{-\infty}^{\infty} e^{f_k(y)} dy \tag{B.8}$$

where

$$f_k(y) = k \ln\left[1 - \Phi(y)\right] + (n-k-1)\ln\left[\Phi(y)\right] - \frac{1}{2}\left(\frac{t - \sqrt{1-\rho}y}{\sqrt{\rho}}\right)^2. \tag{B.9}$$

A standard approach to approximate such an integral is to use Laplace's method. In this approach one expands f about its maximum and then neglects higher order terms, $f(y) \approx f(y_0) - \frac{|f''(y_0)|}{2}(y-y_0)^2$. Having done this, the integral reduces to a standard Gaussian integral. While this approach is asymptotically correct (in the large n and k limit), we can improve the approximation by including more terms using Gauss-Hermite quadrature. Re-writing the integral again, and making another change of variables:

$$I_{n,k} = \sqrt{\frac{1-\rho}{2\pi\rho|f_k''(y_0)|}} e^{f_k(y_0)} \int_{-\infty}^{\infty} e^{-\frac{x^2}{2} + R_k\left(\frac{x}{\sqrt{|f_k''(y_0)|}} + y_0\right)} dx \tag{B.10}$$

where R_k is the remaining terms of f_k after expansion:

$$R_k(y) = f_k(y) - f_k(y_0) + \frac{|f_k''(y_0)|}{2}(y-y_0)^2. \tag{B.11}$$

Now we can approximate the integral using Gauss-Hermite quadrature:

$$I_{n,k}(N) = \sqrt{\frac{1-\rho}{2\pi\rho|f_k''(y_0)|}} e^{f_k(y_0)} \left[\sum_{i=1}^{N} w_i e^{R_k\left(\frac{x_i}{\sqrt{|f_k''(y_0)|}} + y_0\right)}\right], \tag{B.12}$$

where x_i are the points for which $H_N(x_i) = 0$ and the weights w_i are

$$w_i = \frac{N!\sqrt{2\pi}}{N^2\left[H_{N-1}(x_i)\right]^2}. \tag{B.13}$$

Note that $I_{n,k}(1)$ is Laplace's approximation, i.e. Laplace's approximation is a first order Gauss-Hermite quadrature at the maximum of f_k, while $I_{n,k}(N)$ approximates the remainder terms with increasingly high order polynomials and so we expect $I_{n,k}(N) \longrightarrow I_{n,k}$ as N increases.

B.3 Approximation of inverse cumulative distribution function

The normal distribution's inverse cumulative distribution function, $\Phi^{-1}(x)$, can be approximated [201] for $0 < x \leq 0.5$ as

$$\Phi^{-1}(x) \approx \frac{a_0 + a_1 s}{1 + b_1 s + b_2 s^2} - s \tag{B.14}$$

with

$$s = \sqrt{-2 \ln(x)} \tag{B.15}$$

and

$$a_0 = 2.30753, \qquad b_1 = 0.99229, \tag{B.16a}$$

$$a_1 = 0.27061, \qquad b_2 = 0.04481. \tag{B.16b}$$

For $0.5 < x \leq 1$ we use $\Phi^{-1}(x) = -\Phi^{-1}(1 - x)$.

APPENDIX C

Derivation of the Message Passing Equations

Below we provide some additional details of the derivation of the fundamental message passing equations, Eqs. (5.4) and (5.19).

C.1 Percolation

The derivation of Eq. (5.4) follows similar lines to that of Eq. (5.3). By analogy with Eq. (5.2) we can write a generating function for $\pi_{i\leftarrow j}(s|\Gamma_{j\backslash i})$ thus:

$$\begin{aligned}
H_{i\leftarrow j}(z|\Gamma_{j\backslash i}) &= \sum_s \pi_{i\leftarrow j}(s|\Gamma_{j\backslash i})\, z^s \\
&= \sum_s z^s \left\{ \sum_{\{s_k:k\in\Gamma_{j\backslash i}\}} \left[\prod_{k\in\Gamma_{j\backslash i}} \pi_{j\leftarrow k}(s_k) \right] \delta(s-1, \textstyle\sum_{k\in\Gamma_{j\backslash i}} s_k) \right\} \\
&= z \prod_{k\in\Gamma_{j\backslash i}} \sum_{s_k} z^{s_k} \pi_{j\leftarrow k}(s_k) \\
&= z \prod_{k\in\Gamma_{j\backslash i}} H_{j\leftarrow k}(z) \\
&= z \prod_{j\in N^{(r)}_{j\backslash i}} \left[H_{j\leftarrow k}(z) \right]^{w_{j\backslash i,k}} ,
\end{aligned} \tag{C.1}$$

where in the last line we have introduced the random variable $w_{j\backslash i,k}$ which takes the value 1 if $k \in \Gamma_{j\backslash i}$ and 0 otherwise. In other words, $w_{j\backslash i,k} = 1$ if there is a path of occupied edges from j to k in $N^{(r)}_{j\backslash i}$. To compute the generating function for $\pi_{i\leftarrow j}(s)$ we simply average Eq. (C.1) over the possible realizations of $\Gamma_{j\backslash i}$, which leads to the message passing equations

$$\begin{aligned} H_{i\leftarrow j}(z) &= \sum_s \pi_{i\leftarrow j}(s)\, z^s \\ &= \Bigl\langle \sum_s \pi_{i\leftarrow j}(s|\Gamma_{j\backslash i})\, z^s \Bigr\rangle_{\Gamma_{j\backslash i}} \\ &= z \Bigl\langle \prod_{k\in N^{(r)}_{j\backslash i}} H_{j\leftarrow k}(z)^{w_{j\backslash i,k}} \Bigr\rangle_{\Gamma_{j\backslash i}} \\ &= zG_{i\leftarrow j}\bigl(\mathbf{H}_{j\leftarrow}(z)\bigr), \end{aligned} \tag{C.2}$$

as stated in the main text.

C.2 Spectrum

The derivation of the message passing equations for matrix spectra is more complex than for percolation and is given only in abbreviated form in the main text. Here we give the full derivation including intermediate algebraic steps.

As described in the main text, the spectral density of a symmetric matrix **A** is given by

$$\rho(z) = -\frac{1}{n\pi z} \sum_{s=0}^{\infty} \sum_{i=1}^{n} \frac{X_i^s}{z^s}, \tag{C.3}$$

where X_i^s is the sum of the weights of all closed walks of length s that start and end at node i. This sum can be expressed in terms of the sum Y_i^s of the weights of all *excursions* of length s by Eq. (5.10), which we repeat here for convenience:

$$X_i^s = \sum_{m=0}^{\infty} \left[\sum_{s_1=1}^{\infty} \cdots \sum_{s_m=1}^{\infty} \delta\bigl(s, \textstyle\sum_{u=1}^{m} s_u\bigr) \displaystyle\prod_{u=1}^{m} Y_i^{s_u} \right]. \tag{C.4}$$

Substituting this expression into (C.3) we get

$$\rho(z) = -\frac{1}{n\pi z} \sum_{i=1}^{n} \sum_{m=0}^{\infty} \prod_{u=1}^{m} \left[\sum_{s=1}^{\infty} \frac{Y_i^s}{z^s} \right], \tag{C.5}$$

and, defining the function

$$H_i(z) = \sum_{s=1}^{\infty} \frac{Y_i^s}{z^{s-1}}, \tag{C.6}$$

we find that

$$\rho(z) = -\frac{1}{n\pi z}\sum_{i=1}^{n}\sum_{m=0}^{\infty}\left[\frac{H_i(z)}{z}\right]^m = -\frac{1}{n\pi}\sum_{i=1}^{n}\frac{1}{z - H_i(z)}, \tag{C.7}$$

as stated in the main text.

The function $H_i(z)$ we calculate from Eq. (5.15), which tells us that

$$H_i(z) = \sum_{l=0}^{\infty}\frac{1}{z^l}\sum_{w\in W_i^l}|w|\prod_{j\in w}\sum_{m=0}^{\infty}\prod_{k=1}^{m}\sum_{s=1}^{\infty}\frac{Y_{i\leftarrow j}^s}{z^s} = \sum_{w\in W_i}|w|\prod_{j\in w}\frac{1}{z - H_{i\leftarrow j}(z)}, \tag{C.8}$$

where W_i is the set of excursions of all lengths in the neighborhood of i, $|w|$ is the weight of excursion w (i.e., the product of the matrix elements along the excursion), and

$$H_{i\leftarrow j}(z) = \sum_{s=1}^{\infty}\frac{Y_{i\leftarrow j}^s}{z^{s-1}}. \tag{C.9}$$

By an equivalent line of argument we can also show that

$$H_{i\leftarrow j}(z) = \sum_{w\in W_{j\setminus i}}|w|\prod_{k\in w}\frac{1}{z - H_{j\leftarrow k}(z)}. \tag{C.10}$$

This last expression defines the message passing equations for the spectral density calculation. For any given value of z they can be iterated to calculate the spectral density via Eqs. (C.7) and (C.8).

As discussed, the efficiency of this approach relies crucially on being able to perform the sum over excursions w from node j efficiently, which we do as follows. If excursion w returns to j after just a single step (via a self-loop) then it has weight $|w| = A_{jj}$. Otherwise, if it takes two or more steps for a total of $l+1$ steps, visiting l (not necessarily distinct) nodes $k_1, k_2, \ldots, k_l$ along the way (other than the starting node), then the weight is

$$|w| = A_{j,k_1}\left(\prod_{m=1}^{l-1}A_{k_m,k_{m+1}}\right)A_{k_l,j}. \tag{C.11}$$

Inserting these values into (C.10) we get

$$H_{i\leftarrow j}(z) = A_{jj} + \sum_{l=1}^{\infty} \sum_{w\in W_{j\backslash i}^{l}} \frac{A_{j,k_1}}{z - H_{j\leftarrow k_1}(z)} \left(\prod_{m=1}^{l-1} \frac{A_{k_m,k_{m+1}}}{z - H_{j\leftarrow k_{m+1}}} \right) A_{k_l,j} \tag{C.12}$$

where $W_{j\backslash i}^{l}$ is the set of all excursions of length $l + 1$ in $N_{j\backslash i}$. The sum over excursions is equivalent to a sum over all possible sets of l nodes $k_1 \dots k_l$ within the neighborhood, so we can write

$$H_{i\leftarrow j}(z) = A_{jj} + \sum_{l=1}^{\infty} \sum_{k_1} \cdots \sum_{k_l} \frac{A_{j,k_1}}{z - H_{j\leftarrow k_1}(z)} \left(\prod_{m=1}^{l-1} \frac{A_{k_m,k_{m+1}}}{z - H_{j\leftarrow k_{m+1}}} \right) A_{k_l,j}. \tag{C.13}$$

Defining $\mathbf{v}_{i\leftarrow j}$ to be the vector with elements $v_{i\leftarrow j,k} = A_{jk}$ if nodes j and k are directly connected in $N_{j\backslash i}^{(r)}$ and 0 otherwise, $\mathbf{A}^{i\leftarrow j}$ to be the matrix for the neighborhood of j with the neighborhood of i removed, such that

$$A_{kl}^{i\leftarrow j} = \begin{cases} A_{kl} & \text{for } k, l \neq j \text{ and edge } (k, l) \in N_{j\backslash i}^{(r)}, \\ 0 & \text{otherwise,} \end{cases} \tag{C.14}$$

and $\mathbf{D}^{i\leftarrow j}(z)$ to be the diagonal matrix with entries $D_{kk}^{i\leftarrow j} = z - H_{j\leftarrow k}(z)$, we then have

$$\begin{aligned} H_{i\leftarrow j}(z) &= A_{jj} + \sum_{l=1}^{\infty} \sum_{k_1} \sum_{k_l} v_{i\leftarrow j,k_1} \left(D_{k_1,k_1}^{i\leftarrow j} \right)^{-1} \left[\mathbf{A}^{i\leftarrow j} \left(\mathbf{D}^{i\leftarrow j} \right)^{-1} \right]_{k_1,k_l}^{l-1} v_{i\leftarrow j,k_l} \\ &= A_{jj} + \left[\left(\mathbf{D}^{i\leftarrow j} \right)^{-1} \mathbf{v}_{i\leftarrow j} \right]^{T} \left[\mathbf{I} - \mathbf{A}^{i\leftarrow j} \left(\mathbf{D}^{i\leftarrow j} \right)^{-1} \right]^{-1} \mathbf{v}_{i\leftarrow j} \\ &= A_{jj} + \mathbf{v}_{i\leftarrow j}^{T} \left(\mathbf{D}^{i\leftarrow j} - \mathbf{A}^{i\leftarrow j} \right)^{-1} \mathbf{v}_{i\leftarrow j}, \end{aligned} \tag{C.15}$$

as stated in the main text.

APPENDIX D

Monte Carlo Algorithm for Percolation Message Passing

In the message passing equations for bond percolation, Eqs. (5.3) and (5.4), the quantity $G_i(\mathbf{y})$ is a generating function encoding the probability that we can reach nodes in the neighborhood $N_i^{(r)}$ of a given node i by following occupied edges. It is defined by

$$G_i(\mathbf{y}) = \Big\langle \prod_{j \in N_i^{(r)}} y_j^{w_{ij}} \Big\rangle_{\Gamma_i}, \tag{D.1}$$

where w_{ij} is a binary (zero/one) random variable indicating whether node j is reachable from node i and the average is performed over all possible sets Γ_i of reachable nodes, each weighted by the sum of the probabilities of all edge configurations that can give rise to that particular set. The number of such configurations can become large as the size of the neighborhood grows, making exhaustive averages difficult to perform numerically. For larger neighborhoods, therefore, we employ a Monte Carlo averaging scheme as follows.

Suppose that node i has degree k_i and that there are $k_i + M$ edges in the neighborhood $N_i^{(r)}$, with k_i of them directly connected to i and M additional edges that complete cycles between i's neighbors. For locally tree-like networks there are no cycles and $M = 0$, but in general $M \geq 0$. Let $G_i(\mathbf{y}|m)$ be the value of $G_i(\mathbf{y})$ when exactly m of the M additional edges are occupied, which happens with probability $\binom{M}{m} p^m (1-p)^{M-m}$. Then we can write $G_i(\mathbf{y})$ itself in the form

$$G_i(\mathbf{y}) = \sum_{m=0}^{M} G_i(\mathbf{y}|m) \binom{M}{m} p^m (1-p)^{M-m}. \tag{D.2}$$

Our algorithm works by making a Monte Carlo estimate of $G_i(\mathbf{y}|m)$ using a version of the algorithm of Newman and Ziff [202] and then applying (D.2). The basic idea is to occupy edges one by one and keep track of the connected percolation clusters using an efficient union-find data structure based on pointers [202]. Using this data structure the algorithm is able to determine whether two nodes belong to the same cluster, or to join two clusters together, in (very nearly) constant time. To compute $G_i(\mathbf{y}|m)$ itself, the algorithm maintains a record of two quantities for each cluster, a real value x and a probability q. In detail the algorithm works as follows.

The clusters we consider are the sets of nodes in the neighborhood, other than i, that are connected via occupied edges in $N_i(r)$ but not via node i itself, i.e., via the M additional edges mentioned above. Initially none of the M edges is occupied and each node is a cluster in its own right. For each of these one-node clusters j we assign $x_j = y_j$ and we set $q_j = 1 - p$ if node j is a direct neighbor of i or $q_j = 1$ otherwise. We also compute the quantity

$$u_0 = \prod_j \left(q_j + \left(1 - q_j\right) x_j\right). \tag{D.3}$$

Now we occupy the M edges one by one in random order. Let j_1 and j_2 be the nodes at the ends of the mth edge occupied. If j_1 and j_2 are already part of the same cluster before the edge is added (which, as we have said, we can determine in time O(1)), then we set

$$u_m \leftarrow u_{m-1}. \tag{D.4}$$

Otherwise, if j_1 and j_2 are in different clusters r and s, then the addition of the mth edge joins r and s together (which again we can achieve in O(1) time) to make a larger cluster which, without loss of generality, we will label r. At the same time we set

$$u_m \leftarrow \frac{u_{m-1}}{[q_r + \left(1 - q_r\right) x_r][q_s + (1 - q_s)x_s]}, \tag{D.5}$$

$$x_r \leftarrow x_r x_s, \tag{D.6}$$

$$q_r \leftarrow q_r q_s, \tag{D.7}$$

$$u_m \leftarrow u_m\left[q_r + (1 - q_r)x_r\right]. \tag{D.8}$$

After all M edges have been occupied, the $M + 1$ quantities u_m with $m = 0 \ldots M$ give us an estimate of $G_i(\mathbf{y}|m)$, and $G_i(\mathbf{y})$ can be calculated from (D.2) as

$$G_i(\mathbf{y}) \simeq \sum_{m=0}^{M} u_m \binom{M}{m} p^m (1 - p)^{M-m}. \tag{D.9}$$

The calculation of $G_{i\leftarrow j}(\mathbf{y})$ is identical except for the replacement of the neighborhood by $N^{(r)}_{j\backslash i}$. Finally, we average the results over repeated runs of the algorithm to get our estimate of the generating functions. We find surprisingly good results with averages over a relatively small number of runs—we used just eight runs for each neighborhood to generate the results shown in Fig. 5.2.

Note that the sequence of edges added and cluster joins performed does not depend on the values of either $\mathbf{y}$ or p, which means we can use the same sequence to calculate $G_i(\mathbf{y})$ for many different $\mathbf{y}$ and p. We can also use the same sequence on successive iterations of the message passing process, which has the benefit of removing any statistical fluctuations between iterations and is useful when estimating convergence of the message passing process, which can otherwise be difficult to do.

As is often the case for Monte Carlo calculations, it is not easy to say exactly how many runs will be required to get good results. Note, however, that if we perform S runs for each neighborhood then, because neighborhoods are sampled independently, we effectively generate S^n configurations of the whole network, and this number can become very large for large n even when S is small. Thus we expect to get good answers even with quite modest values of S, and indeed this is what we see in the calculations reported in this thesis.

BIBLIOGRAPHY

[1] M. E. J. Newman, Networks. Oxford University Press, Oxford, U.K., 2nd edition (2018).

[2] A.-L. Barabási and M. Pósfai, Network Science. Cambridge University Press, Cambridge, U.K. (2016).

[3] E. Estrada and P. A. Knight, A First Course in Network Theory. Oxford University Press, Oxford, U.K., 1st edition (2015).

[4] S. N. Dorogovtsev, Lectures on Complex Networks. Oxford University Press, Oxford, U.K. (2010).

[5] E. Estrada, Journal of Complex Networks: Quo Vadis? *Journal of Complex Networks* **1**(1), 1–2 (2013), URL https://academic.oup.com/comnet/article-lookup/doi/10.1093/comnet/cnt008.

[6] A. Jadbabaie, IEEE Transactions on Network Science and Engineering. *IEEE Transactions on Network Science and Engineering* **1**(1), 2–9 (2014), URL http://ieeexplore.ieee.org/document/7000012/.

[7] U. Brandes, G. Robins, A. McCranie, and S. Wasserman, What is Network Science? *Network Science* **1**(1), 1–15 (2013), URL https://www.cambridge.org/core/product/identifier/S2050124213000027/type/journal_article.

[8] 10th International Conference on Complex Networks. URL https://complenet19.weebly.com/.

[9] Network Science Society Conference 2020 (NetSci 2020 Roma). URL https://netsci2020.netscisociety.net/.

[10] Complex Networks 2020. URL https://complexnetworks.org/.

[11] A.-L. Barabási, Linked: How Everything Is Connected to Everything Else and What It Means For Business, Science, and Everyday Life. Basic Books, New York (2014).

[12] D. J. Watts, Six Degrees: The Science of a Connected Age. Norton, New York, 1st edition (2003).

[13] Network Science Institute at Northeastern University. URL https://www.networkscienceinstitute.org/.

[14] Indiana University Network Science Institute. URL https://iuni.iu.edu/index.

[15] S. Horvath, Weighted Network Analysis: Application in Genomics and Systems Biology. Springer, New York, NY (2011).

[16] J. D. Medaglia, M.-E. Lynall, and D. S. Bassett, Cognitive network neuroscience. *Journal of Cognitive Neuroscience* **27**(8), 1471–1491 (2015), URL http://www.mitpressjournals.org/doi/10.1162/jocn_a_00810.

[17] D. S. Bassett and O. Sporns, Network neuroscience. *Nature Neuroscience* **20**(3), 353–364 (2017), URL http://www.nature.com/articles/nn.4502.

[18] E. Chautard, N. Thierry-Mieg, and S. Ricard-Blum, Interaction networks: From protein functions to drug discovery. A review. *Pathologie Biologie* **57**(4), 324–333 (2009), URL https://linkinghub.elsevier.com/retrieve/pii/S0369811408002538.

[19] T. Ideker and R. Sharan, Protein networks in disease. *Genome Research* **18**(4), 644–652 (2008), URL http://www.genome.org/cgi/doi/10.1101/gr.071852.107.

[20] R. M. May, Network structure and the biology of populations. *Trends in Ecology & Evolution* **21**(7), 394–399 (2006), URL https://linkinghub.elsevier.com/retrieve/pii/S0169534706001005.

[21] S. Proulx, D. Promislow, and P. Phillips, Network thinking in ecology and evolution. *Trends in Ecology & Evolution* **20**(6), 345–353 (2005), URL https://linkinghub.elsevier.com/retrieve/pii/S0169534705000881.

[22] D. A. Luke and J. K. Harris, Network analysis in public health: History, methods, and applications. *Annual Review of Public Health* **28**(1), 69–93 (2007), URL http://www.annualreviews.org/doi/10.1146/annurev.publhealth.28.021406.144132.

[23] J. H. Fowler, T. R. Johnson, J. F. Spriggs, S. Jeon, and P. J. Wahlbeck, Network analysis and the law: Measuring the legal importance of precedents at the U.S. Supreme Court. *Political Analysis* **15**(3), 324–346 (2007), URL https://www.cambridge.org/core/product/identifier/S1047198700006525/type/journal_article.

[24] D. R. White, V. Batagelj, and A. Mrvar, Anthropology: Analyzing large kinship and marriage networks with Pgraph and Pajek. *Social Science Computer Review* **17**(3), 245–274 (1999), URL http://journals.sagepub.com/doi/10.1177/089443939901700302.

[25] A. W. Wolfe, The rise of network thinking in anthropology. *Social Networks* **1**(1), 53–64 (1978), URL https://linkinghub.elsevier.com/retrieve/pii/0378873378900126.

[26] L. C. Freeman, The Development of Social Network ANALYSIS: A Study in the Sociology of Science. Empirical Press, Vancouver, BC (2004).

[27] P. J. Carrington, J. Scott, and S. Wasserman (eds.), Models and Methods in Social Network Analysis. Number 27 in Structural analysis in the social sciences, Cambridge University Press, Cambridge, U.K. (2005).

[28] J. Scott, Social network analysis. *Sociology* **22**(1), 109–127 (1988), URL http://journals.sagepub.com/doi/10.1177/0038038588022001007.

[29] M. D. Ward, K. Stovel, and A. Sacks, Network analysis and political science. *Annual Review of Political Science* **14**(1), 245–264 (2011), URL http://www.annualreviews.org/doi/10.1146/annurev.polisci.12.040907.115949.

[30] D. Lazer, Networks in political science: Back to the future. *PS: Political Science & Politics* **44**(01), 61–68 (2011), URL http://www.journals.cambridge.org/abstract_S1049096510001873.

[31] T. L. Friesz, Network Science, Nonlinear Science and Infrastructure Systems. Number v. 102 in International series in operations research & management science, Springer, New York (2007).

[32] D. Y. Kenett and S. Havlin, Network science: A useful tool in economics and finance. *Mind & Society* **14**(2), 155–167 (2015), URL http://link.springer.com/10.1007/s11299-015-0167-y.

[33] D. Easley and J. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, Cambridge, U.K. (2010).

[34] J. H. v. Lint and R. M. Wilson, A Course in Combinatorics. Cambridge University Press, Cambridge, U.K., 2nd edition (2001).

[35] F. Brauer, P. Van den Driessche, J. Wu, and L. J. S. Allen (eds.), Mathematical Epidemiology. Number 1945 in Mathematical biosciences subseries, Springer, Berlin (2008).

[36] A. Huppert and G. Katriel, Mathematical modelling and prediction in infectious disease epidemiology. *Clinical Microbiology and Infection* **19**(11), 999–1005 (2013), URL https://linkinghub.elsevier.com/retrieve/pii/S1198743X14630019.

[37] I. Z. Kiss, J. C. Miller, and P. L. Simon, Mathematics of Epidemics on Networks: From Exact to Approximate Models. Number Volume 46 in Interdisciplinary applied mathematics, Springer, Cham (2017).

[38] G. St-Onge, J.-G. Young, L. Hébert-Dufresne, and L. J. Dubé, Efficient sampling of spreading processes on complex networks using a composition and rejection algorithm. *Computer Physics Communications* **240**, 30–37 (2019), URL https://linkinghub.elsevier.com/retrieve/pii/S0010465519300608.

[39] B. Bollobás, A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics* **1**(4), 311–316 (1980), URL https://linkinghub.elsevier.com/retrieve/pii/S0195669880800308.

[40] A.-L. Barabási and R. Albert, Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999), URL https://www.sciencemag.org/lookup/doi/10.1126/science.286.5439.509.

[41] D. J. Watts and S. H. Strogatz, Collective dynamics of 'small-world' networks. *Nature* **393**(6684), 440–442 (1998), URL http://www.nature.com/articles/30918.

[42] J. F. Padgett and C. K. Ansell, Robust action and the rise of the Medici, 1400-1434. *American Journal of Sociology* **98**(6), 1259–1319 (1993), URL https://doi.org/10.1086/230190.

[43] B. Karrer, M. E. J. Newman, and L. Zdeborová, Percolation on sparse networks. *Physical Review Letters* **113**(20), 208702 (2014), URL https://link.aps.org/doi/10.1103/PhysRevLett.113.208702.

[44] P. Grassberger, On the critical behavior of the general epidemic process and dynamical percolation. *Mathematical Biosciences* **63**(2), 157–172 (1983), URL https://linkinghub.elsevier.com/retrieve/pii/0025556482900360.

[45] L. Sander, C. Warren, and I. Sokolov, Epidemics, disorder, and percolation. *Physica A: Statistical Mechanics and its Applications* **325**(1-2), 1–8 (2003), URL https://linkinghub.elsevier.com/retrieve/pii/S0378437103001766.

[46] L. C. Freeman, Centrality in social networks conceptual clarification. *Social Networks* **1**(3), 215–239 (1978), URL https://linkinghub.elsevier.com/retrieve/pii/0378873378900217.

[47] S. P. Borgatti and M. G. Everett, A graph-theoretic perspective on centrality. *Social Networks* **28**(4), 466–484 (2006), URL https://linkinghub.elsevier.com/retrieve/pii/S0378873305000833.

[48] M. Girvan and M. E. J. Newman, Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* **99**(12), 7821–7826 (2002), URL http://www.pnas.org/cgi/doi/10.1073/pnas.122653799.

[49] P. Bonacich, Factoring and weighting approaches to status scores and clique identification. *The Journal of Mathematical Sociology* **2**(1), 113–120

(1972), URL http://www.tandfonline.com/doi/abs/10.1080/0022250X.1972.9989806.

[50] L. Katz, A new status index derived from sociometric analysis. *Psychometrika* **18**(1), 39–43 (1953), URL http://link.springer.com/10.1007/BF02289026.

[51] J. M. Kleinberg, Authoritative sources in a hyperlinked environment. *Journal of the ACM* **46**(5), 604–632 (1999), URL http://dl.acm.org/doi/10.1145/324133.324140.

[52] S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* **30**(1-7), 107–117 (1998), URL https://linkinghub.elsevier.com/retrieve/pii/S016975529800110X.

[53] S. P. Borgatti and M. G. Everett, Models of core/periphery structures. *Social Networks* **21**(4), 375–395 (2000), URL https://linkinghub.elsevier.com/retrieve/pii/S0378873399000192.

[54] M. P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha, Core-periphery structure in networks. *SIAM Journal on Applied Mathematics* **74**(1), 167–190 (2014), URL http://epubs.siam.org/doi/10.1137/120881683.

[55] P. Holme, Core-periphery organization of complex networks. *Physical Review E* **72**(4), 046111 (2005), URL https://link.aps.org/doi/10.1103/PhysRevE.72.046111.

[56] L. Hébert-Dufresne, J. A. Grochow, and A. Allard, Multi-scale structure and topological anomaly detection via a new network statistic: The onion decomposition. *Scientific Reports* **6**(1), 31708 (2016), URL http://www.nature.com/articles/srep31708.

[57] M. McPherson, L. Smith-Lovin, and J. M. Cook, Birds of a feather: Homophily in social networks. *Annual Review of Sociology* **27**(1), 415–444 (2001), URL http://www.annualreviews.org/doi/10.1146/annurev.soc.27.1.415.

[58] M. E. J. Newman, Mixing patterns in networks. *Physical Review E* **67**(2), 026126 (2003), URL https://link.aps.org/doi/10.1103/PhysRevE.67.026126.

[59] J. Moody, Race, school integration, and friendship segregation in America. *American Journal of Sociology* **107**(3), 679–716 (2001), URL http://www.journals.uchicago.edu/doi/10.1086/338954.

[60] J. L. Moreno, Who Shall Survive? Foundations of Sociometry, Group Psychotherapy and Sociodrama. Beacon House, Beacon, NY (1953).

[61] S. P. Borgatti, M. G. Everett, and J. C. Johnson, Analyzing Social Networks. SAGE, Los Angeles, 2nd edition (2018).

[62] S. P. Borgatti, Centrality and network flow. *Social Networks* **27**(1), 55–71 (2005), URL https://linkinghub.elsevier.com/retrieve/pii/S0378873304000693.

[63] R. Solomonoff and A. Rapoport, Connectivity of random nets. *The Bulletin of Mathematical Biophysics* **13**(2), 107–117 (1951), URL http://link.springer.com/10.1007/BF02478357.

[64] P. Erdős and A. Rényi, On random graphs. *Math. debrecen* **6**, 290–297 (1959).

[65] P. Erdős and A. Rényi, On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* **5**(1), 17–60 (1960).

[66] E. N. Gilbert, Random graphs. *The Annals of Mathematical Statistics* **30**(4), 1141–1144 (1959), URL www.jstor.org/stable/2237458.

[67] B. K. Fosdick, D. B. Larremore, J. Nishimura, and J. Ugander, Configuring random graph models with fixed degree sequences. *SIAM Review* **60**(2), 315–355 (2018), URL https://epubs.siam.org/doi/10.1137/16M1087175.

[68] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Random graphs with arbitrary degree distributions and their applications. *Physical Review E* **64**(2), 026118 (2001), URL https://link.aps.org/doi/10.1103/PhysRevE.64.026118.

[69] P. W. Holland, K. B. Laskey, and S. Leinhardt, Stochastic blockmodels: First steps. *Social Networks* **5**(2), 109–137 (1983), URL https://linkinghub.elsevier.com/retrieve/pii/0378873383900217.

[70] T. P. Peixoto, Entropy of stochastic blockmodel ensembles. *Physical Review E* **85**(5), 056122 (2012), URL https://link.aps.org/doi/10.1103/PhysRevE.85.056122.

[71] B. Karrer and M. E. J. Newman, Stochastic blockmodels and community structure in networks. *Physical Review E* **83**(1), 016107 (2011), URL https://link.aps.org/doi/10.1103/PhysRevE.83.016107.

[72] D. J. de Solla Price, A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science* **27**(5), 292–306 (1976), URL http://doi.wiley.com/10.1002/asi.4630270505.

[73] D. J. de Solla Price, Networks of scientific papers. *Science* **149**(3683), 510–515 (1965), URL https://www.sciencemag.org/lookup/doi/10.1126/science.149.3683.510.

[74] I. de Sola Pool and M. Kochen, Contacts and influence. *Social Networks* **1**(1), 5–51 (1978), URL https://linkinghub.elsevier.com/retrieve/pii/0378873378900114.

[75] M. E. J. Newman and M. Girvan, Finding and evaluating community structure in networks. *Physical Review E* **69**(2), 026113 (2004), URL https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

[76] M. E. J. Newman, Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* **103**(23), 8577–8582 (2006), URL http://www.pnas.org/cgi/doi/10.1073/pnas.0601602103.

[77] S. Fortunato and D. Hric, Community detection in networks: A user guide. *Physics Reports* **659**, 1–44 (2016), URL https://linkinghub.elsevier.com/retrieve/pii/S0370157316302964.

[78] M. E. J. Newman, Equivalence between modularity optimization and maximum likelihood methods for community detection. *Physical Review E* **94**(5), 052315 (2016), URL https://link.aps.org/doi/10.1103/PhysRevE.94.052315.

[79] H. S. Migon, D. Gamerman, and F. Louzada, Statistical Inference: an Integrated Approach. Chapman and Hall/CRC, 2nd edition (2014), URL https://www.taylorfrancis.com/books/9780429066849.

[80] M. A. Riolo, G. T. Cantwell, G. Reinert, and M. E. J. Newman, Efficient method for estimating the number of communities in a network. *Physical Review E* **96**(3), 032310 (2017), URL https://link.aps.org/doi/10.1103/PhysRevE.96.032310.

[81] T. P. Peixoto, Hierarchical block structures and high-resolution model selection in large networks. *Physical Review X* **4**(1), 011047 (2014), URL https://link.aps.org/doi/10.1103/PhysRevX.4.011047.

[82] F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová, Typology of phase transitions in Bayesian inference problems. *Physical Review E* **99**(4), 042109 (2019), URL https://link.aps.org/doi/10.1103/PhysRevE.99.042109.

[83] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters* **107**(6), 065701 (2011), URL https://link.aps.org/doi/10.1103/PhysRevLett.107.065701.

[84] D. Hric, T. P. Peixoto, and S. Fortunato, Network structure, metadata, and the prediction of missing nodes and annotations. *Physical Review X* **6**(3), 031038 (2016), URL https://link.aps.org/doi/10.1103/PhysRevX.6.031038.

[85] C. De Bacco, D. B. Larremore, and C. Moore, A physical model for efficient ranking in networks. *Science Advances* **4**(7), eaar8260 (2018), URL https://advances.sciencemag.org/lookup/doi/10.1126/sciadv.aar8260.

[86] P. D. Hoff, A. E. Raftery, and M. S. Handcock, Latent space approaches to social network analysis. *Journal of the American Statistical Association* **97**(460), 1090–1098 (2002), URL http://www.tandfonline.com/doi/abs/10.1198/016214502388618906.

[87] D. Liben-Nowell and J. Kleinberg, The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology* **58**(7), 1019–1031 (2007), URL http://doi.wiley.com/10.1002/asi.20591.

[88] A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová, Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Physical Review E* **90**(1), 012801 (2014), URL https://link.aps.org/doi/10.1103/PhysRevE.90.012801.

[89] J.-G. Young, G. St-Onge, E. Laurence, C. Murphy, L. Hébert-Dufresne, and P. Desrosiers, Phase transition in the recoverability of network history. *Physical Review X* **9**(4), 041056 (2019), URL https://link.aps.org/doi/10.1103/PhysRevX.9.041056.

[90] M. E. J. Newman, Network structure from rich but noisy data. *Nature Physics* **14**(6), 542–545 (2018), URL http://www.nature.com/articles/s41567-018-0076-1.

[91] J. K. Harris, An Introduction to Exponential Random Graph Modeling. Number 173 in Quantitative Applications in the Social Sciences, SAGE, Los Angeles (2014).

[92] S. Wasserman and P. Pattison, Logit models and logistic regressions for social networks: I. An introduction to Markov graphs and p*. *Psychometrika* **61**(3), 401–425 (1996), URL http://link.springer.com/10.1007/BF02294547.

[93] C. R. Shalizi and A. Rinaldo, Consistency under sampling of exponential random graph models. *The Annals of Statistics* **41**(2), 508–535 (2013), URL http://projecteuclid.org/euclid.aos/1366980556.

[94] S. Bhamidi, G. Bresler, and A. Sly, Mixing time of Exponential Random Graphs. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pp. 803–812, IEEE, Philadelphia, PA (2008), URL https://ieeexplore.ieee.org/document/4691012/.

[95] J. Park and M. E. J. Newman, Solution of the two-star model of a network. *Physical Review E* **70**(6), 066146 (2004), URL https://link.aps.org/doi/10.1103/PhysRevE.70.066146.

[96] S. Chen, A. Mira, and J.-P. Onnela, Flexible model selection for mechanistic network models. *arXiv:1804.00237 [stat]* (2019), URL http://arxiv.org/abs/1804.00237.

[97] T. Rogers, I. P. Castillo, R. Kühn, and K. Takeda, Cavity approach to the spectral density of sparse symmetric random matrices. *Physical Review E* **78**(3), 031116 (2008), URL https://link.aps.org/doi/10.1103/PhysRevE.78.031116.

[98] G. T. Cantwell and M. E. J. Newman, Mixing patterns and individual differences in networks. *Physical Review E* **99**(4), 042306 (2019), URL https://link.aps.org/doi/10.1103/PhysRevE.99.042306.

[99] G. T. Cantwell, Y. Liu, B. F. Maier, A. C. Schwarze, C. A. Serván, J. Snyder, and G. St-Onge, Thresholding normally distributed data creates complex networks. *Physical Review E* **101**(6), 062302 (2020), URL https://link.aps.org/doi/10.1103/PhysRevE.101.062302.

[100] G. T. Cantwell and M. E. J. Newman, Message passing on networks with loops. *Proceedings of the National Academy of Sciences* **116**(47), 23398–23403 (2019), URL http://www.pnas.org/lookup/doi/10.1073/pnas.1914893116.

[101] P. Block and T. Grund, Multidimensional homophily in friendship networks. *Network Science* **2**(2), 189–212 (2014), URL https://www.cambridge.org/core/product/identifier/S2050124214000174/type/journal_article.

[102] R. Noldus and P. Van Mieghem, Assortativity in complex networks. *Journal of Complex Networks* **3**(4), 507–542 (2015), URL https://academic.oup.com/comnet/article-lookup/doi/10.1093/comnet/cnv005.

[103] A. Condon and R. M. Karp, Algorithms for graph partitioning on the planted partition model. *Random Structures & Algorithms* **18**(2), 116–140 (2001), URL https://onlinelibrary.wiley.com/doi/abs/10.1002/1098-2418%28200103%2918%3A2%3C116%3A%3AAID-RSA1001%3E3.0.CO%3B2-2.

[104] F. McSherry, Spectral partitioning of random graphs. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pp. 529–537, IEEE, Newport Beach, CA, USA (2001), URL https://ieeexplore.ieee.org/document/959929/.

[105] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, The anatomy of the Facebook social graph. *arXiv:1111.4503* (2011), URL http://arxiv.org/abs/1111.4503.

[106] K. M. Altenburger and J. Ugander, Monophily in social networks introduces similarity among friends-of-friends. *Nature Human Behaviour* **2**(4), 284–290 (2018), URL http://www.nature.com/articles/s41562-018-0321-8.

[107] L. Peel, J.-C. Delvenne, and R. Lambiotte, Multiscale mixing patterns in networks. *Proceedings of the National Academy of Sciences* **115**(16), 4057–4062 (2018), URL http://www.pnas.org/lookup/doi/10.1073/pnas.1713019115.

[108] M. Piraveenan, M. Prokopenko, and A. Y. Zomaya, Local assortativeness in scale-free networks. *EPL (Europhysics Letters)* **84**(2), 28002 (2008), URL https://iopscience.iop.org/article/10.1209/0295-5075/84/28002.

[109] M. Piraveenan, M. Prokopenko, and A. Y. Zomaya, On congruity of nodes and assortative information content in complex networks. *Networks and Heterogeneous Media* **7**(3), 441–461 (2012), URL http://www.aimsciences.org/journals/displayArticlesnew.jsp?paperID=7797.

[110] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, Mixed membership stochastic blockmodels. *Journal of Machine Learning Research* **9**, 34 (2008).

[111] P. Latouche, E. Birmelé, and C. Ambroise, Overlapping stochastic block models with application to the French political blogosphere. *The Annals of Applied Statistics* **5**(1), 309–336 (2011), URL http://projecteuclid.org/euclid.aoas/1300715192.

[112] F. Chung and L. Lu, Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics* **6**(2), 125–145 (2002), URL http://link.springer.com/10.1007/PL00012580.

[113] W. W. Zachary, An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* **33**(4), 452–473 (1977), URL https://www.journals.uchicago.edu/doi/10.1086/jar.33.4.3629752.

[114] L. A. Adamic and N. Glance, The political blogosphere and the 2004 U.S. election: Divided they blog. In *Proceedings of the 3rd international workshop on link discovery - LinkKDD '05*, pp. 36–43, ACM Press, Chicago, Illinois (2005), URL http://portal.acm.org/citation.cfm?doid=1134271.1134277.

[115] K. M. Harris and J. R. Udry, National Longitudinal Study of Adolescent to Adult Health (Add Health), 1994-2008 [Public Use]: Version 21 (2008), URL https://www.icpsr.umich.edu/icpsrweb/DSDR/studies/21600/versions/V21. Type: dataset.

[116] P. Gill, J. Lee, K. R. Rethemeyer, J. Horgan, and V. Asal, Lethal connections: The determinants of network connections in the Provisional Irish Republican Army, 1970–1998. *International Interactions* **40**(1), 52–78 (2014), URL http://www.tandfonline.com/doi/abs/10.1080/03050629.2013.863190.

[117] W. N. Francis and H. Kučera, A standard corpus of present-day edited American English (1979).

[118] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E* **84**(6), 066106 (2011), URL https://link.aps.org/doi/10.1103/PhysRevE.84.066106.

[119] E. Mossel, J. Neeman, and A. Sly, A proof of the block model threshold conjecture. *Combinatorica* **38**(3), 665–708 (2018), URL http://link.springer.com/10.1007/s00493-016-3238-8.

[120] C. M. Bishop, Pattern Recognition and Machine Learning. Information science and statistics, Springer, New York (2006).

[121] M. R. Gupta and Y. Chen, Theory and use of the EM algorithm. *Foundations and Trends in Signal Processing* **4**(3), 223–296 (2010), URL http://www.nowpublishers.com/article/Details/SIG-034.

[122] D. P. Landau and K. Binder, A Guide to Monte Carlo Simulations in Statistical Physics. Cambridge University Press, Cambridge, 4th edition (2014), URL http://ebooks.cambridge.org/ref/id/CBO9781139696463.

[123] M. E. J. Newman and G. T. Barkema, Monte Carlo Methods in Statistical Physics. Clarendon Press; Oxford University Press, Oxford, U.K. (1999).

[124] S. v. Buuren, Flexible Imputation of Missing Data. Chapman & Hall/CRC interdisciplinary statistics series, CRC Press, Boca Raton, FL (2012).

[125] E. Lazega, The Collegial Phenomenon: the Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership. Oxford University Press, Oxford, U.K. (2001).

[126] A. L. Traud, P. J. Mucha, and M. A. Porter, Social structure of Facebook networks. *Physica A: Statistical Mechanics and its Applications* **391**(16), 4165–4180 (2012), URL https://linkinghub.elsevier.com/retrieve/pii/S0378437111009186.

[127] M. Mezard and A. Montanari, Information, Physics, and Computation. Oxford Graduate Texts, Oxford University Press, Oxford (2009).

[128] D. J. C. MacKay, Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge, U.K. (2003).

[129] K. F. Riley, M. P. Hobson, and S. J. Bence, Mathematical Methods For Physics and Engineering. Cambridge University Press, Cambridge, 3rd edition (2006).

[130] J. W. Cooley and J. W. Tukey, An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation* **19**(90), 297–297 (1965), URL http://www.ams.org/jourcgi/jour-getitem?pii=S0025-5718-1965-0178586-1.

[131] N. Goldenfeld, Lectures on Phase Transitions and the Renormalization Group. Frontiers in physics; v. 85, Addison-Wesley, Advanced Book Program, Reading, Mass. (1992).

[132] S. M. Ross, Stochastic Processes. Wiley series in probability and statistics, Wiley, New York, 2nd edition (1996).

[133] M. P. H. Stumpf and M. A. Porter, Critical truths about power laws. *Science* **335**(6069), 665–666 (2012), URL https://www.sciencemag.org/lookup/doi/10.1126/science.1216142.

[134] P. L. Krapivsky and S. Redner, Network growth by copying. *Physical Review E* **71**(3), 036118 (2005), URL https://link.aps.org/doi/10.1103/PhysRevE.71.036118.

[135] E. T. Jaynes and G. L. Bretthorst, Probability Theory: the Logic of Science. Cambridge University Press, Cambridge, U.K. (2003).

[136] J. Berg and M. Lässig, Correlated random networks. *Physical Review Letters* **89**(22), 228701 (2002), URL https://link.aps.org/doi/10.1103/PhysRevLett.89.228701.

[137] L. Apeltsin, J. H. Morris, P. C. Babbitt, and T. E. Ferrin, Improving the quality of protein similarity network clustering algorithms using the network edge weight distribution. *Bioinformatics* **27**(3), 326–333 (2011), URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3031030/.

[138] E. Bullmore and O. Sporns, Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience* **10**(3), 186–198 (2009).

[139] N. Langer, A. Pedroni, and L. Jäncke, The problem of thresholding in small-world network analysis. *PLOS ONE* **8**(1) (2013), URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3536769/.

[140] M. Rubinov and O. Sporns, Complex network measures of brain connectivity: uses and interpretations. *Neuroimage* **52**(3), 1059–1069 (2010).

[141] W.-Q. Huang, X.-T. Zhuang, and S. Yao, A network analysis of the Chinese stock market. *Physica A: Statistical Mechanics and its Applications* **388**(14), 2956–2964 (2009).

[142] V. Sekara and S. Lehmann, The strength of friendship ties in proximity sensor data. *PLOS ONE* **9**(7), e100915 (2014), URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0100915.

[143] A. Stopczynski, V. Sekara, P. Sapiezynski, A. Cuttone, M. M. Madsen, J. E. Larsen, and S. Lehmann, Measuring large-scale social networks with high resolution. *PLOS ONE* **9**(4), e95978 (2014), URL http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0095978.

[144] M. Á. Serrano, M. Boguñá, and A. Vespignani, Extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences* **106**(16), 6483–6488 (2009).

[145] F. Radicchi, J. J. Ramasco, and S. Fortunato, Information filtering in complex weighted networks. *Physical Review E* **83**(4), 046101 (2011).

[146] N. Dianati, Unwinding the hairball graph: Pruning algorithms for weighted complex networks. *Physical Review E* **93**(1), 012304 (2016).

[147] J. Vaynberg and J. Qin, Weak protein–protein interactions as probed by NMR spectroscopy. *Trends in Biotechnology* **24**(1), 22 – 27 (2006).

[148] B. F. Maier, benmaier/ThredgeCorr (2019), URL https://github.com/benmaier/ThredgeCorr. Original-date: 2018-06-19T23:29:11Z.

[149] P. Diaconis and S. Janson, Graph limits and exchangeable random graphs. *Rendiconti di Matematica* **Serie VII**(28), 33–61 (2008).

[150] P. Orbanz and D. M. Roy, Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE transactions on pattern analysis and machine intelligence* **37**(2), 437–461 (2015).

[151] D. N. Hoover, Relations on probability spaces and arrays of random variables. *Institute for Advanced Study, Princeton, NJ* (1979).

[152] B. Harris and A. P. Soms, The use of the tetrachoric series for evaluating multivariate normal probabilities. *Journal of Multivariate Analysis* **10**(2), 252–267 (1980), URL http://www.sciencedirect.com/science/article/pii/0047259X80900172.

[153] P. D. Miller, Applied Asymptotic Analysis. American Mathematical Soc. (2006).

[154] M. E. J. Newman, The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences* **98**, 404–409 (2001).

[155] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein, Reactome: A knowledgebase of biological pathways. *Nucleic Acids Research* **33**, D428–432 (2005).

[156] B. Bollobás, Random Graphs. Cambridge Studies in Advanced Mathematics, Cambridge University Press, 2nd edition (2001).

[157] D. Fernholz and V. Ramachandran, The diameter of sparse random graphs. *Random Structures and Algorithms* **31**, 482–516 (2007).

[158] R. Cohen and S. Havlin, Scale-free networks are ultrasmall. *Physical Review Letters* **90**(5) (2003), URL https://link.aps.org/doi/10.1103/PhysRevLett.90.058701.

[159] R. Cohen, S. Havlin, and D. ben Avraham, Structural properties of scale-free networks. In S. Bornholdt and H. G. Schuster (eds.), *Handbook of Graphs and Networks*, pp. 85–110, Wiley-VCH Verlag, Weinheim, FRG (2004), URL http://doi.wiley.com/10.1002/3527602755.ch4.

[160] H. A. Bethe, Statistical theory of superlattices. *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences* **150**(871), 552–575 (1935), URL https://royalsocietypublishing.org/doi/10.1098/rspa.1935.0122.

[161] J. Pearl, Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the Second AAAI Conference on Artificial Intelligence*, AAAI'82, pp. 133–136, AAAI Press, Pittsburgh, Pennsylvania (1982).

[162] M. Mezard, Analytic and algorithmic solution of random satisfiability problems. *Science* **297**(5582), 812–815 (2002), URL https://www.sciencemag.org/lookup/doi/10.1126/science.1073287.

[163] S. Yoon, A. V. Goltsev, S. N. Dorogovtsev, and J. F. F. Mendes, Belief-propagation algorithm and the Ising model on networks with arbitrary distributions of motifs. *Physical Review E* **84**(4), 041144 (2011), URL https://link.aps.org/doi/10.1103/PhysRevE.84.041144.

[164] B. Karrer and M. E. J. Newman, Message passing approach for general epidemic models. *Physical Review E* **82**(1), 016101 (2010), URL https://link.aps.org/doi/10.1103/PhysRevE.82.016101.

[165] R. Gallager, Low-density parity-check codes. *IEEE Transactions on Information Theory* **8**(1), 21–28 (1962), URL http://ieeexplore.ieee.org/document/1057683/.

[166] B. J. Frey and D. J. C. MacKay, A revolution: Belief propagation in graphs with cycles. In *Proceedings of the 10th International Conference on Neural Information Processing Systems*, NIPS'97, pp. 479–485, MIT Press, Denver, CO (1997).

[167] M. Mezard, G. Parisi, and M. A. Virasoro, Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications. World Scientific Publishing Company (1987).

[168] M. Chertkov and V. Y. Chernyak, Loop calculus in statistical physics and information science. *Physical Review E* **73**(6), 065102 (2006), URL https://link.aps.org/doi/10.1103/PhysRevE.73.065102.

[169] M. E. J. Newman, Random graphs with clustering. *Physical Review Letters* **103**(5), 058701 (2009), URL https://link.aps.org/doi/10.1103/PhysRevLett.103.058701.

[170] J. C. Miller, Percolation and epidemics in random clustered networks. *Physical Review E* **80**(2), 020901 (2009), URL https://link.aps.org/doi/10.1103/PhysRevE.80.020901.

[171] B. Karrer and M. E. J. Newman, Random graphs containing arbitrary distributions of subgraphs. *Physical Review E* **82**(6), 066118 (2010), URL https://link.aps.org/doi/10.1103/PhysRevE.82.066118.

[172] M. E. J. Newman, Spectra of networks containing short loops. *Physical Review E* **100**(1), 012314 (2019), URL https://link.aps.org/doi/10.1103/PhysRevE.100.012314.

[173] J. S. Yedidia, W. T. Freeman, and Y. Weiss, Generalized belief propagation. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, pp. 668–674, MIT Press, Denver, CO (2000).

[174] J. Yedidia, W. Freeman, and Y. Weiss, Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory* **51**(7), 2282–2312 (2005), URL http://ieeexplore.ieee.org/document/1459044/.

[175] H. L. Frisch and J. M. Hammersley, Percolation processes and related topics. *Journal of the Society for Industrial and Applied Mathematics* **11**(4), 894–918 (1963), URL http://epubs.siam.org/doi/10.1137/0111066.

[176] D. Stauffer and A. Aharony, Introduction to Percolation Theory. Routledge, London (2003). OCLC: 249097091.

[177] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, Models of social networks based on social distance attachment. *Physical Review E* **70**(5), 056122 (2004), URL https://link.aps.org/doi/10.1103/PhysRevE.70.056122.

[178] R. R. Nadakuditi and M. E. J. Newman, Spectra of random graphs with arbitrary expected degrees. *Physical Review E* **87**(1), 012803 (2013), URL https://link.aps.org/doi/10.1103/PhysRevE.87.012803.

[179] A. Weiße, G. Wellein, A. Alvermann, and H. Fehske, The kernel polynomial method. *Reviews of Modern Physics* **78**(1), 275–306 (2006), URL https://link.aps.org/doi/10.1103/RevModPhys.78.275.

[180] R. J. Baxter, Exactly Solved Models in Statistical Mechanics. Academic Press, London, U.K. (1982).

[181] S. Salinas, Introduction to Statistical Physics. Graduate Texts in Contemporary Physics, Springer-Verlag, New York (2001), URL https://www.springer.com/gp/book/9780387951195.

[182] N. Friel and J. Wyse, Estimating the evidence - a review. *Statistica Neerlandica* **66**(3), 288–308 (2012), URL http://doi.wiley.com/10.1111/j.1467-9574.2011.00515.x.

[183] C. E. Shannon, Prediction and entropy of printed English. *Bell System Technical Journal* **30**(1), 50–64 (1951), URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6773263.

[184] W. S. Bialek, Biophysics: Searching For Principles. Princeton University Press, Princeton, NJ (2012).

[185] P. Cabral, G. Augusto, M. Tewolde, and Y. Araya, Entropy in urban systems. *Entropy* **15**(12), 5223–5236 (2013), URL http://www.mdpi.com/1099-4300/15/12/5223.

[186] R. Zhou, R. Cai, and G. Tong, Applications of entropy in finance: A review. *Entropy* **15**(12), 4909–4931 (2013), URL http://www.mdpi.com/1099-4300/15/11/4909.

[187] E. Ising, Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik* **31**(1), 253–258 (1925), URL http://link.springer.com/10.1007/BF02980577.

[188] L. Onsager, Crystal statistics. I. A two-dimensional model with an order-disorder transition. *Physical Review* **65**(3-4), 117–149 (1944), URL https://link.aps.org/doi/10.1103/PhysRev.65.117.

[189] M. Mézard and G. Parisi, The Bethe lattice spin glass revisited. *The European Physical Journal B* **20**(2), 217–233 (2001), URL http://link.springer.com/10.1007/PL00011099.

[190] A. K. Hartmann and M. Weigt, Phase Transitions in Combinatorial Optimization Problems: Basics, Algorithms and Statistical Mechanics. Wiley-VCH, Weinheim (2005).

[191] M. E. J. Newman, The structure and function of complex networks. *SIAM Review* **45**(2), 167–256 (2003), URL http://epubs.siam.org/doi/10.1137/S003614450342480.

[192] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Kaufmann, San Francisco, Calif (2008).

[193] J. S. Yedidia, W. T. Freeman, and Y. Weiss, Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, pp. 239–269, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2003).

[194] R. Kikuchi, A theory of cooperative phenomena. *Physical Review* **81**(6), 988–1003 (1951), URL https://link.aps.org/doi/10.1103/PhysRev.81.988.

[195] M. Pulver, pulver/autodiff (2020), URL https://github.com/pulver/autodiff. Original-date: 2018-12-16T16:17:01Z.

[196] G. Robins, P. Pattison, Y. Kalish, and D. Lusher, An introduction to exponential random graph (p*) models for social networks. *Social Networks* **29**(2), 173–191 (2007), URL https://linkinghub.elsevier.com/retrieve/pii/S0378873306000372.

[197] D. Cai, T. Campbell, and T. Broderick, Edge-exchangeable graphs and sparsity. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4249–4257, Curran Associates, Inc. (2016), URL http://papers.nips.cc/paper/6586-edge-exchangeable-graphs-and-sparsity.pdf.

[198] L. Tierney and J. B. Kadane, Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* **81**(393), 82–86 (1986), URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1986.10478240.

[199] G. T. Cantwell, gcant/individual_mixing (2019), URL https://github.com/gcant/individual_mixing. Original-date: 2019-03-11T18:14:08Z.

[200] Y. L. Tong, The Multivariate Normal Distribution. Springer (1990).

[201] M. Abramowitz and I. A. Stegun (eds.), Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables. National Bureau of Standards (1964).

[202] M. E. J. Newman and R. M. Ziff, Efficient Monte Carlo algorithm and high-precision results for percolation. *Physical Review Letters* **85**(19), 4104–4107 (2000), URL https://link.aps.org/doi/10.1103/PhysRevLett.85.4104.