Theoretical tools for network analysis: Game theory, graph centrality, and statistical inference

by

Travis Bennett Martin

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2016

Doctoral Committee:

Professor Mark E. Newman, Co-Chair
Professor Michael P. Wellman, Co-Chair
Assistant Professor Raj Rao Nadakuditi
Associate Professor Seth Pettie
Assistant Professor Grant Schoenebeck

For my parents, and Hannah

ACKNOWLEDGMENTS

I am immensely grateful to so many people who have supported and encouraged me on my academic journey. First, and foremost, I owe everything to my co-advisors Mark Newman and Michael Wellman. I'm lucky to have *two* role-models to teach me volumes about how to research and learn and, most importantly, how to think about the world. I couldn't ask for a more understanding, supportive, or admirable pair. Mark's kindness and encouragement made graduate school much less like slave labor than the rumors make it out to be, and his passion for elegant science will always be an inspiration. Mike's guidance through graduate school and beyond has been the best salve for my self-doubt, and I've learned much from his intellectual integrity and ability to distill messy questions to their core.

I am deeply thankful for the invaluable help and feedback of my dissertation committee, Raj Rao Nadakuditi, Seth Pettie, and Grant Schoenebeck. I've learned much from them throughout my dissertation process, and even more from them over the last five years. I'm forever indebted to my fellow graduate students for their kindness and friendship. I'm grateful to SRG, for giving me a community of friends where I always felt understood, and especially grateful to Elaine Wah and Erik Brinkman for their friendship, support, and many good memories, academic and otherwise. I'm also grateful to the Newman Group, for helping me talk through ideas, always teaching me new things, and your unwavering friendships. I'm especially grateful to Brian Ball for his guidance in my early days, from which I learned much, and to Thomas Zhang for keeping me smiling and for being the perfect sounding board.

I am fortunate to have had numerous fantastic friends and roommates throughout my graduate career, my Michigan family. Their friendships have kept me sane and given me great joy, and we've formed memories that I will remember forever. I am thankful to all the friendly administrators who have made my path easier, and to many inspiring collaborators, especially those at Microsoft Research. I owe much to Joe Grogan, a dear friend who entrusted me with my first Computer Science job, and to Luay Nakhleh, who inspired my love of academic computer science and suggested I work with Mark.

I am blessed to have endlessly generous and supportive parents who have encouraged me no matter what. Enormous thanks also to the rest of my family, especially my siblings Kelsey and Tyler Martin, on whom I can always depend and who always make me feel local, no matter the distance. Lastly, to Hannah, the love of my life, thank you for your patience, wisdom, kindness, and heart.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Theoretical tools for network analysis: Game theory, graph centrality, and statistical inference

by

Travis Bennett Martin

Chairs: Mark E. Newman and Michael P. Wellman

A computer-driven data explosion has made the difficulty of interpreting large data sets of interconnected entities ever more salient. My work focuses on theoretical tools for summarizing, analyzing, and understanding *network* data sets, or data sets of things and their pairwise connections. I address four network science issues, improving our ability to analyze networks from a variety of domains.

I first show that the sophistication of game-theoretic agent decision making can crucially effect network cascades: differing decision making assumptions can lead to dramatically different cascade outcomes. This highlights the importance of diligence when making assumptions about agent behavior on networks and in general. I next analytically demonstrate a significant irregularity in the popular eigenvector centrality, and propose a new spectral centrality measure, *nonbacktracking centrality*, showing that it avoids this irregularity. This tool contributes a more robust way of ranking nodes, as well as an additional mathematical understanding of the effects of network localization. I next give a new model for *uncertain networks*, networks in which one has no access to true network data but instead observes only probabilistic information about edge existence. I give a fast maximum-likelihood algorithm for recovering edges and communities in this model, and show that it outperforms a typical approach of thresholding to an unweighted network. This model gives a better tool for understanding and analyzing real-world uncertain networks such as those arising in the experimental sciences. Lastly, I give a new lens for understanding scientific literature, specifically as a hybrid coauthorship and citation network. I use this for exploratory analysis of the Physical Review journals over a hundred-year period, and I make new observations about the interplay between these two networks and how this relationship has changed over time.

CHAPTER I

Introduction

A network, at its simplest, is a collection of things and their pairwise connections. We call these things *nodes* and call these connections *edges*. A near infinitude of real-world systems can be and are fruitfully modeled as networks. To name a few data sources, social networks model people and their relationships; the World Wide Web consists of web pages and their hypertext link connections; and protein-protein interaction networks model the chemical relationships between proteins in organisms.

Network science studies not only the above but all complex networks, with the ultimate goal of better modeling, describing, and understanding networks in the real world from a variety of disciplines. One major undertaking of network science is the development of general techniques for understanding networks which can then be applied broadly across domains. Originally, networks were small and analyzable by hand, see for example the network of grade school friendships in Figure 1.1, and these techniques were simple manual calculations such as counting the edges in the network. As computers and algorithms have improved and proliferated, network data has grown to a scale which makes simple network understanding more challenging. How can one begin to make sense of the network of information sharing on Facebook shown in Figure 1.2? But this proliferation has also enabled the creation of more powerful computerized techniques.

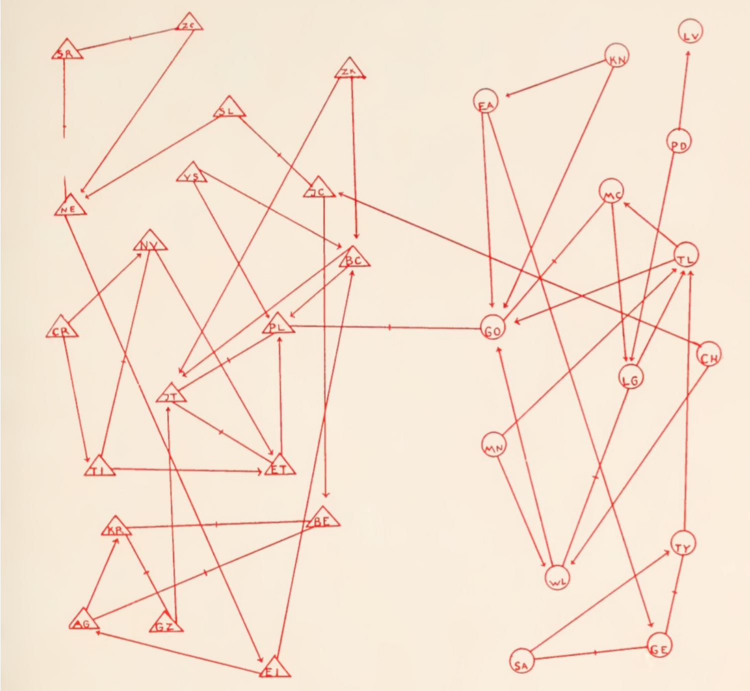


Figure 1.1: A network of friendships between third graders, from 1934 [130].

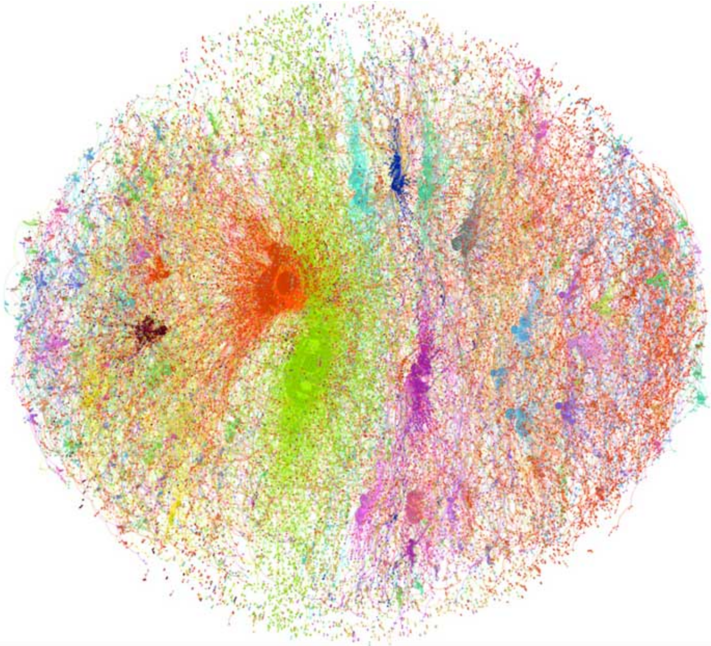


Figure 1.2: A network of a single meme spreading through Facebook, from 2016 [34].

These days, we need computerized algorithms and methods for summarizing and aggregating large-scale network data in productive ways. My work in network science gives new theories for making sense of the deluge of network data available today. In this thesis I discuss four techniques for summarizing and analyzing network data. I show that processes on networks can depend crucially on whether the agents in a network behave strategically. I give a new algorithm for finding important nodes in a network, and a new model for reasoning about networks about which we only have uncertain information. Lastly, I give a new lens for analyzing an academic network.

1.1 History and overview of network science

In the 18th century, mathematicians asked a simple question about the amply bridged Prussian city of Königsberg: can one walk between the four landmasses of the city while crossing each bridge exactly once? In 1735 Leonhard Euler presented a general solution to this problem [64], and his representation of landmasses and bridges using dots and lines is commonly regarded as the beginning of the field of graph theory and also as the beginning of network science.

While graph theory has a long history, the first examples of modern network science came from the study of social networks. In 1933 Moreno gave early examples of social networks in his book about *sociograms*, or "The problem of human interrelations" [130]. In 1969 Travers and Milgram ran an experiment to measure how closely arbitrary people in the United States were connected, finding famously that their participants were connected on average by "six degrees of separation" [173]. In the 1970s Granovetter [74, 75] contributed the seminal sociological theories that weak connections between individuals are essential for widespread influence, and that human collective behavior, such as rioting, can be very sensitive to a group's individual makeup. In a 1977 field study [189], Zachary tracked the classic Karate Club Network as an internal conflict caused one karate club to split into two clubs, and showed that clusters of friends in the club's original social network almost exactly predicted the makeup of the split.

More recently, the impact of network science has diversified to an encyclopedic scale. A selection of major network science results follows. In Section 1.2 I define and discuss several of these concepts in more detail.

Early mathematical developments such as the study of random graph models by Rapoport and Erdős and Rényi [59, 60, 61, 157] laid the groundwork for theoretical analysis of networks. Producing tractable-yet-useful models for explaining networks has remained a core pursuit of network science. Many models have *communities*, or groups of similar nodes with similar connection behaviors [1, 10, 27, 86, 95, 108, 183, 184, 192]. Most networks feature a highly skewed connectivity distribution, with a small number of nodes having substantially more connections than the rest. Numerous graph models give possible mechanisms for this commonality [56, 106, 154]. For example, Barabási and Albert [14] gave the *preferential attachment* model, in which nodes are more likely to connect to nodes which already have more neighbors. Other models aim to replicate properties of specific varieties of networks such as directed edges [179], bipartite structure [109], hierarchies [182], and arbitrary degree distributions [37, 144].

A network *centrality* measure gives a way of assigning numeric importance scores to the nodes of a network, and is often used as a way to quantitatively find the important nodes in a network. PageRank [29] is likely the most famous centrality measure, and was Google's primary original method of ranking web pages on the World Wide Web. A host of other centrality measures exist [25, 67, 97, 103, 118], and the relevant measure depends on what one values as important [28].

Many applications, from epidemiology to advertising, can be modeled as a spreading process on a network. Granovetter's sociological contributions included seminal work on *network cascades* [74], in which a small local event, such as the adoption of a fad, spreads over the network edges. Cascade data is becoming increasingly accessible on the Internet, and many cascade variants have been discussed in the literature [32, 34, 71, 99, 101]. Kempe et al. [99] considered a model of an advertiser maximizing the spread of a cascade, and many such strategic cascade maximization problems have been considered since [35, 73]. Mollison [128] extended simple fully mixed epidemiological models, where all individuals interact homogeneously, to models with heterogeneous contact probabilities based on the distance between individuals. This laid the groundwork for network-based epidemiology, and many network-based epidemiological studies have followed [76, 96, 137].

Computer scientists and statisticians have long been interested in clustering data points [114] and dividing graphs [100, 123], but network science has developed many targeted techniques for finding communities or other empirically inspired structures in networks. Structure can be found by fitting graph models such as those above to data [10, 105], or by taking a more metric-based algorithmic approach to finding communities. Newman and Girvan [142] define *modularity*, which states that a division of a network into communities is good to the extent that there are more edges within communities than one would expect according to chance. Many community finding algorithms aim to optimize modularity [22, 80] or other criteria [6, 39, 66, 85, 133, 148, 162].

Graphs are a convenient representation for many economic models of interacting agents [98, 91], and network science has contributed to a better understanding of these models. In some models the network governs how the economic agents interact [68, 73, 92, 163], while others give possible rationalizations for the network structures we commonly observe [8, 90, 171].

A more complete landscape of Network Science can be found in books by Wasserman and Faust [180], Newman [140], and Easley and Kleinberg [57].

1.2 Basic network science techniques

Here I describe a variety of basic network science concepts that are used throughout the thesis. Refer to Wasserman and Faust [180], Newman [140] or Easley and Kleinberg [57] for a complete introduction.

1.2.1 Graph notation

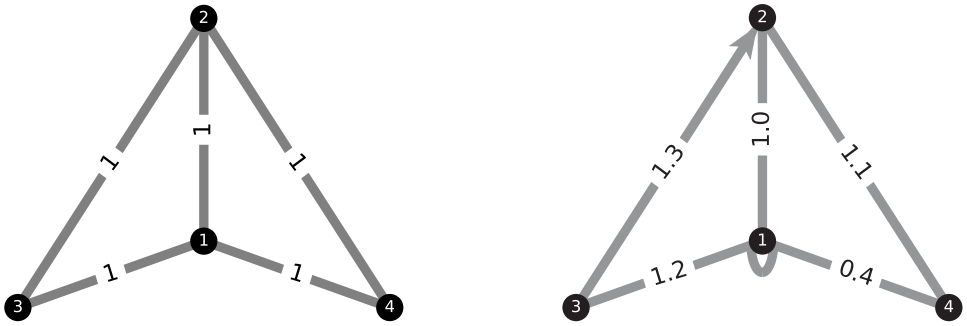
A network can be represented as a graph $G = (V, E)$, where V are the nodes of the graph and $E \subseteq V \times V$ are the edges of the graph. Two nodes are *neighbors* if they are connected by an edge $(i, j) \in E$. I denote the set of neighbors of i by $nb(i) \subset V$. I commonly let $n = |V|$ and $m = |E|$.

Networks may also be represented in matrix form using an *adjacency matrix*, $\mathbf{A}$, with n rows and n columns, where A_{ij} is an element of the matrix having value one if there is an edge between nodes i and j and zero otherwise.

Adjacency matrices are often a computationally efficient method of storing and manipulating *dense* networks, those with many (m is $\Omega(n^2)$) edges between nodes. But for *sparse* networks, where m is $o(n^2)$, it can be more efficient to use an *adjacency list* representation, which lists all connections for each node.

1.2.2 Types of networks

In an *undirected* network edges are symmetric, so $A_{ij} = A_{ji}$, while a *directed* network allows the possibility of edges in only one direction. In an *unweighted* network all edges are equivalent, while a *weighted* network can have edges of varying numeric weights, represented by $A_{ij} = w$ for $w \in \mathbb{R}$. A network with *self-loops* allows a node to be connected to itself, and a network with *multi-edges* allows a pair of nodes to have multiple edges between it. A network without self-loops or multi-edges is *simple.* Examples of these types of graphs can be seen in Figure 1.3. In this thesis I primarily



$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 & 1.2 & 0.4 \\ 1 & 0 & 0 & 1.1 \\ 1.2 & 1.3 & 0 & 0 \\ 0.4 & 1.1 & 0 & 0 \end{pmatrix}$$

Figure 1.3: An undirected, unweighted network without self-loops or multi-edges (left) and a directed, weighted network with self loops and multi-edges (right), and their adjacency matrices.

consider undirected, unweighted, simple graphs.[1]

1.2.3 Degree distribution

The number of neighbors a node has is called its *degree*, d. For node i, $d_i = |nb(i)|$. One granular description of a network is its *degree distribution*, which is a vector $\boldsymbol{p}$ of the fraction of nodes with each possible degree. Many networks, especially large social networks, have *power-law* degree distributions. That is, the fraction of nodes in the graph with degree d, p_d, is approximately proportional to $d^{-\alpha}$ for some $\alpha > 0$, termed the power-law exponent. While the degree distribution is rarely enough information to uniquely identify a network, it characterizes an often-useful aggregate network property.

1.2.4 Clustering coefficient

Edge relationships in graphs are frequently *transitive*, which is the property that if i is connected to j and j is connected to k, i is also connected to k. The only fully

[1] However, I also employ self-loops or multi-edges when they make my calculations cleaner.

transitive graphs are collections of *cliques*, a graph structure with an edge between every pair of nodes, but networks may also be *partially transitive.* That is, for many edge pairs (i, j) and (j, k), i is also connected to k. This is true for a vanishingly small fraction of edge pairs in most large random graphs, but is quite common in most real networks. Many explanations for this partial transitivity exist. In social networks this is equivalent to saying: I'm often friends with my friend's friends. This could be because a shared friend is likely to introduce us, or simply because our shared friend increases the chances that we run in the same circles and thus would be friends with or without our mutual friend.

The extent to which edges in a network obey transitivity is captured by the *global clustering coefficient* [182], which is defined for a given graph as,

$$\text{global clustering coefficient} = \frac{\text{closed node triplets}}{\text{connected node triplets}}, \tag{1.1}$$

where a connected node triplet is any three nodes with at least two edges, and a closed node triplet is any three nodes with three edges among them. For example, the clustering coefficient in the unweighted graph in Figure 1.3 is $\frac{2}{4}$.

1.2.5 Centrality measures

As discussed before, network centrality is a way of measuring how important, or central, each node in a network is. The simplest of centrality measures, *degree centrality*, is the number of connections a node has to other nodes. If one assumed that importance in the world depended exclusively on how many friends one had, one could find important people simply by calculating and ranking by degree centrality.

Eigenvector centrality [25] is a more sophisticated variant which recognizes that not all acquaintances are equal. I am more important, or influential, if the people I know are themselves influential. Eigenvector centrality defines a centrality score v_i

for each node i in an undirected network. v_i is proportional to the sum of the scores of the node's network neighbors, $v_i = \lambda^{-1} \sum_j A_{ij} v_j$, where λ is a proportionality constant and the sum is over all nodes. Defining a vector $\boldsymbol{v}$ whose elements are the v_i, we then have

$$\mathbf{A}\boldsymbol{v} = \lambda \boldsymbol{v}, \tag{1.2}$$

meaning that the vector of centralities is an eigenvector of the adjacency matrix. If we further stipulate that the centralities should all be nonnegative, the Perron-Frobenius theorem [170] states that $\boldsymbol{v}$ must be the leading eigenvector (the vector corresponding to the greatest positive eigenvalue λ). There are many ways of computing this centrality. One can directly use any number of matrix-based algorithms, or one can initialize $\boldsymbol{v}$ arbitrarily and iterate Eq. (1.2) to convergence. We further characterize behavior of this centrality measure, and propose an extension, in Chapter III.

1.2.6 Random graph models and Erdős-Rényi random graphs

A random graph model gives a method for generating network instances to emulate real network data sets. These models can provide useful controlled data for testing new methods or tractable systems for deriving mathematical results. The models can also be used in reverse—instead of generating a network, one can infer structure by fitting a random graph model to a data set.

Erdős and Rényi defined the simplest random graph model in use [59]. In an *Erdős-Rényi graph*, every distinct pair of nodes is connected by an undirected edge with independent probability $p = c/(n-1)$, where c is the expected mean degree of a node. While its simplicity makes it an attractive starting point for random graph models, its homogeneous structure and narrowly peaked binomial[2] degree distribution make it a poor approximation of most real-world networks.

[2] Often approximated with a Poisson distribution for large graphs.

1.2.7 Preferential attachment

Many large networks of a wide variety of types, from Twitter follower networks to gene regulation networks, have degree distributions which obey a power-law in the tail. While the Erdős-Rényi model and stochastic block model (Section 1.2.8) are mathematically elegant models for some simpler networks, a richer random graph model is necessary for generating networks with power-law degree distributions.

The *preferential attachment model* of Barabási and Albert [14] is a simple undirected graph model for generating networks with power-law degree distributions which also simulates the growth of networks over time. This model is a special case of the cumulative advantage model of Price [154].

The preferential attachment model begins with n_0 nodes, each with c arbitrarily distributed edges.[3] In each time step t one node appears and adds c undirected edges to the nodes in the existing network. These edges are randomly distributed in proportion to the existing node degrees, so the probability of adding an edge to existing node i is $P(d_i) = \frac{d_i}{\sum_{j=1}^{n_t-1} d_j}$. Thus the number of nodes at time t is $n_t = n_0 + t$ and the number of edges is exactly $m_t = cn_o + ct$, so that the average network degree is always $2c$.

The rate that a node receives edges over time, in the limit of large t and with a continuous approximation, is

$$\frac{\partial d_i}{\partial t} = cP(d_i) = c\frac{d_i}{\sum_{j=1}^{n_t-1} d_j} = c\frac{d_i}{2tc} = \frac{d_i}{2t}.$$

Integrating with respect to t gives the degree of a node that joined at time t_i at some time t in the future,

$$d_i(t) = c\left(\frac{t}{t_i}\right)^{1/2}.$$

[3] The original source [14] doesn't specify the initial configuration of edges, which have no influence on the properties of the network in the limit of long time. Here I assume a uniform initial degree of $2c$ for each node, for simplicity.

Thus, this model produces networks in which we expect the degree of a node to asymptotically depend exclusively on its age. The full derivation [14] also shows that we expect an overall degree distribution with $p_d \sim d^{-3}$.

1.2.8 Block models

The *stochastic block model* (SBM) is a random graph model widely used for modeling communities in networks [86, 95, 146]. In the conventional definition of the stochastic block model, n nodes are distributed at random among k groups, each with a probability γ_r of being assigned to group r, where $\sum_{r=1}^{k} \gamma_r = 1$. Then undirected edges are placed independently at random between node pairs with probabilities ω_{rs} that depend only on the groups r, s that a pair belongs to. Thus each node in group r has a degree drawn from a binomial (approximately Poisson) distribution with mean $\sum_s n\gamma_s\omega_{rs}$. The *ordinary block model* is a simpler deterministic special case of the stochastic block model, where probabilities of connection are either 0 or 1.

If the diagonal elements ω_{rr} of the probability matrix are larger than the off-diagonal entries then one has traditional *assortative* community structure, which means the network has a higher density of connections within groups than between them. But one can also make the diagonal entries smaller than the off-diagonal entries to generate *disassortative* structure or mixed structure types. Methods for detecting assortative, disassortative, and other structures in networks are discussed below.

1.2.9 Community structure and detection

Most networks have groups, or communities, of nodes which have similar connection behaviors. For example, within the network of all college students, students at University of Michigan (UM) display assortative structure, as they are much more likely than average college students to be friends with other UM students. As another example, the national college dating network may display disassortative structure, in that students more frequently date students of a differing gender. In the *community detection problem*, one is given only the nodes and edges of a network and must infer the underlying node affiliations. Community detection has proven to be a robust way of simplifying and understanding networks.

A division of a network into k communities is given in the form of a vector $\boldsymbol{g} \in \{1, ..., k\}^n$, where the community assignment for each node is represented by an integer. A central question of community detection is, “What makes a particular community division good?”

Computer science has many metrics for evaluating graph partitions, such as minimum cut or conductance. One popular metric in network science is *modularity* [142], which informally is the extent to which a network has more edges within communities than one would expect by chance. That is, if one were to re-wire edges completely at random while preserving the expected degrees of nodes, one would expect there to be an edge between nodes i and j with probability $\frac{d_i d_j}{2m}$. However, in an actual network with adjacency matrix $\mathbf{A}$, the relative probability is deterministic, given by A_{ij}. Formally, the modularity is defined as

$$\text{modularity} = \frac{1}{2m} \sum_{(i,j) \in E} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(g_i, g_j),$$

where δ is the Kronecker delta: $\delta(a, b) = 1 \iff a = b$. There exist many procedures for maximizing this criterion in order to find communities, for example via greedy [22] or spectral techniques [141].

There exist several problems with modularity maximization, one of which is that it fails to account for communities of nodes which are disassortative, as in the dating example mentioned above. A popular technique for handling more general types of community structure is to assume that the network came from a stochastic block model, with blocks corresponding to communities. Community detection then becomes a problem of inferring the unknown block model from the network data.

1.2.10 Maximum likelihood estimation

Maximum likelihood estimation (MLE) is a general statistical technique in which one estimates unknown model parameters to be the ones that maximize the likelihood of observed data. This allows one to infer the parameters of a generative model, such as the SBM, from an observed instance of that model, such as a network.

For the SBM, we can write down the probability of generating a particular SBM network $\mathbf{A}$ as a function of the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$ and can marginalize over the latent group variables $\boldsymbol{g}$,

$$\begin{aligned} P(\mathbf{A}|\boldsymbol{\gamma},\boldsymbol{\omega}) &= \sum_{\boldsymbol{g}} P(\mathbf{A}|\boldsymbol{g},\boldsymbol{\omega})P(\boldsymbol{g}|\boldsymbol{\gamma}) \\ &= \sum_{\boldsymbol{g}} \left[\prod_{i<j} \omega_{g_i g_j}^{A_{ij}} (1-\omega_{g_i g_j})^{1-A_{ij}} \prod_i \gamma_{g_i} \right]. \end{aligned} \tag{1.3}$$

Then we say that the MLE block model parameters are those that maximize Equation (1.3). This could be computed exactly in exponential time by summing over all k^n possible values of $\boldsymbol{g}$, but in practice this maximization is approximated by methods such as the expectation-maximization algorithm or simulated annealing.

1.2.11 Expectation-maximization algorithm

As noted above, straightforward computation of MLE requires the maximization of a generally intractable sum. The expectation-maximization (EM) algorithm [53, 122] is an iterative procedure for finding a local maximum, useful as a heuristic for global maximization.

For example, to apply the EM algorithm to the SBM setting, we maximize $P(\mathbf{A}|\boldsymbol{\gamma},\boldsymbol{\omega})$ by iteratively estimating values for $\boldsymbol{\gamma}, \boldsymbol{\omega}$, and $\boldsymbol{g}$. First, it is easier to maximize the log-probability,

$$\log P(\mathbf{A}|\boldsymbol{\gamma},\boldsymbol{\omega}) = \log \sum_{\boldsymbol{g}} P(\mathbf{A}|\boldsymbol{g},\boldsymbol{\omega})P(\boldsymbol{g}|\boldsymbol{\gamma}). \tag{1.4}$$

We simplify our log-of-a-sum expression using *Jensen's inequality*, which states that for any set of positive-definite quantities x_i, the log of their sum satisfies

$$\log \sum_i x_i \geq \sum_i q_i \log \frac{x_i}{q_i}, \tag{1.5}$$

where q_i is any probability distribution over i satisfying the normalization condition $\sum_i q_i = 1$. One can easily verify by substitution that the exact equality is achieved by choosing

$$q_i = \frac{x_i}{\sum_i x_i}. \tag{1.6}$$

Thus choosing q_i according to Eq. (1.6) effectively maximizes Eq. (1.5) with respect to $\boldsymbol{q}$.

Returning to Eq. (1.4) and applying Jensen's inequality gives

$$\log P(\mathbf{A}|\boldsymbol{\gamma},\boldsymbol{\omega}) \geq \sum_{\boldsymbol{g}} q(\boldsymbol{g}) \log \frac{P(\mathbf{A}|\boldsymbol{g},\boldsymbol{\omega})P(\boldsymbol{g}|\boldsymbol{\gamma})}{q(\boldsymbol{g})}. \tag{1.7}$$

We choose $q(\boldsymbol{g})$ according to Eq. (1.6),

$$q(\boldsymbol{g}) = \frac{P(\mathbf{A},\boldsymbol{g}|\boldsymbol{\gamma},\boldsymbol{\omega})}{\sum_{\boldsymbol{g}} P(\mathbf{A},\boldsymbol{g}|\boldsymbol{\gamma},\boldsymbol{\omega})} = \frac{P(\mathbf{A},\boldsymbol{g}|\boldsymbol{\gamma},\boldsymbol{\omega})}{P(\mathbf{A}|\boldsymbol{\gamma},\boldsymbol{\omega})}. \tag{1.8}$$

While Eq. (1.7) appears more complicated than our original Eq. (1.4), it has the desirable property that, given $q(\boldsymbol{g})$, it can be directly maximized by differentiation with respect to parameters $\boldsymbol{\gamma},\boldsymbol{\omega}$. This is called the M-step. The EM algorithm performs maximization by iteratively maximizing Eq. (1.7) and then performing the so-called E-step, updating $q(\boldsymbol{g})$ according to Eq. (1.8).

A series of simplifications yields

$$q(\boldsymbol{g}) = \frac{P(\mathbf{A},\boldsymbol{g}|\gamma,\boldsymbol{\omega})}{\sum_{\boldsymbol{g}} P(\mathbf{A},\boldsymbol{g}|\gamma,\boldsymbol{\omega})} = \frac{P(\mathbf{A},\boldsymbol{g}|\gamma,\boldsymbol{\omega})}{P(\mathbf{A}|\gamma,\boldsymbol{\omega})} = P(\boldsymbol{g}|\mathbf{A},\boldsymbol{\gamma},\boldsymbol{\omega}). \tag{1.9}$$

So the EM algorithm has the added benefit that, upon convergence, Eq. (1.8) gives an estimate of our latent group variables, giving us a way of measuring community structure. I give more details of the EM algorithm applied to the SBM and extensions of the SBM in Chapter IV.

1.2.12 Belief propagation

Belief propagation [150] gives an exact and efficient method of calculating a joint probability distribution over discrete random variables whose dependencies can be represented by a tree, and is used for example in studying graphical models. It works by iteratively passing messages between nodes, and has been shown to be approximately correct in practice, though inexact, in real-world non-tree graphs.

The message-passing approach has recently proved useful in network science for efficiently calculating network properties [50, 105, 165]. In particular, it allows us to more quickly calculate the values in Eq. (1.8), as I show in detail in Chapter IV. In the SBM, the messages that nodes pass to each other are estimates of the nodes' community memberships.

1.2.13 Random matrix theory for networks

An adjacency matrix, like all square matrices, has n not necessarily distinct eigenvalues. These eigenvalues together are called the *spectrum* of the matrix. The restricted form of the adjacency matrix allows us to say useful things about its spectrum. For example, the adjacency matrix of an undirected graph is always symmetric, and therefore has an all-real spectrum. The mathematics of random matrix theory allows us to further characterize the usual behavior of simple-enough random graph models, giving us further insight into the structure of networks [134, 172, 191].

In the limit of large n, the spectra of many random graph models, including the Erdős-Rényi and SBM, obey general trends. They often consist of a continuous band of eigenvalues, with only a few *outlying eigenvalues* below or above this band. These outlying eigenvalues and their corresponding eigenvectors frequently contain useful information about network structure.

For example, as mentioned above, every network has a greatest eigenvalue which is positive and has a corresponding all-positive eigenvector which gives the eigenvector centrality. In the SBM, the number of outlying eigenvalues above the continuous spectral band is equal to the number of communities in the model [191].

Many network science techniques use the spectral properties of other graph matrices, such as the modularity matrix [190] or the graph Laplacian [19].

1.2.14 Game theory on networks

Game theory is the study of systems of interacting, self-interested, and intelligent agents. Such a system is termed a *game*. A game has n players, and each player i has her own *strategy set* $\boldsymbol{S_i}$ of possible strategies, from which i chooses a (possibly randomized) strategy $s_i \in \boldsymbol{S_i}$. The outcome of the game is determined by the strategy choices of all players, $\boldsymbol{s} = (s_1, ..., s_n)$. Each player has a utility function $u_i(s_i, \boldsymbol{s_{-i}})$ which depends on the strategy of player i and the strategies of all other players, $\boldsymbol{s_{-i}}$. An agent's goal is to maximize her expected utility.

How does one determine what will happen for a given game? How will agents reason about other agents and choose their own strategies? Game theory has many *equilibrium concepts* which attempt to answer these questions. The *Nash equilibrium* defines an outcome $\boldsymbol{s}$ to be an equilibrium if and only if no agent could strictly increase her expected utility by changing strategies while all other agents stayed the same. That is, for all i,

$$u_i(s_i, \boldsymbol{s_{-i}}) \geq u_i(s_i', \boldsymbol{s_{-i}}) \text{ for all } s_i' \in \boldsymbol{S_i}. \tag{1.10}$$

A Nash equilibrium represents one of many potentially justifiable rational outcomes for a game, and thus could be a reasonable guess for what might happen in a game. Game theory also gives extensive theory for more sophisticated situations, including games in which people make decisions in some order after observing the actions of others, and games in which agents have uncertainty about the details of the game.

There are rich connections between game theory and network science. Many natural games involve networks, and networks also give a convenient way for expressing relationships between agents, for example by representing agents as nodes and their influences by edges. As has been repeatedly shown, including in Chapter II and a study of network cascades [120], the structure of the network can have dramatic impact on the game result, so tools for understanding networks are vital for understanding these types of games.

1.3 Overview of chapters

In this chapter I've given an introduction to network science, an overview of the landscape of related work and major results, and some basic notation and concepts that are used throughout the chapters to come. In the next four chapters I describe new theoretical tools for network analysis, which together contribute to the growing body of techniques for understanding and analyzing networks in the real world.

In Chapter II I ask, "What is the impact of the sophistication of agent (game-theoretic) decision making in network cascades?" I show that differing assumptions about agent decision making can lead to dramatically different cascade outcomes, highlighting the importance of care when making assumptions about agent behavior on networks and in general. For this project I wrote the majority of the manuscript, and formulated and proved the majority of the results. Michael Wellman originally suggested this model. Grant Schoenebeck derived the cloud network result and the result limiting performance for non-adaptive schedules. This chapter is based on work published in the proceedings of the *ACM Conference on Economics and Computation* [117].

In Chapter III I analytically demonstrate a significant irregularity in the popular eigenvector centrality, and propose a new spectral centrality measure, *nonbacktracking centrality*, showing that it avoids the irregularity. This tool contributes a more robust way of ranking nodes, as well as an additional mathematical understanding of the effects of localization. I calculated all numerical results for this project, and wrote the first manuscript draft. Xiao Zhang, Mark Newman, and I jointly discussed analytical results and edited the manuscript. This chapter is based on work published in *Physical Review E* [118].

In Chapter IV I create a new model for *uncertain networks*: networks in which one has no access to true network data but instead observes only probabilities of edge existence. I give a fast maximum-likelihood algorithm for recovering edges and communities in this model, and show that it outperforms a typical approach of thresholding to an unweighted network. My model gives a better method of understanding and analyzing these real-world uncertain networks such as those arising in the experimental sciences. I calculated all numerical results for this project and derived the final definition of an uncertain network. I collaborated with Brian Ball and Mark Newman in brainstorming possible uncertain network models, finding analytical solutions, and deriving the BP equations. Brian Ball gave the first, Kullback-Leibler-based, formulation of an uncertain network. This chapter is based on work published in *Physical Review E* [119].

Lastly, in Chapter V I give a unique way of understanding scientific literature, specifically as a hybrid coauthorship and citation network. I use this for exploratory analysis of the Physical Review journals over a hundred-year period, and I make new observations about the interplay between these two networks and how this relationship has changed over time. For this project I was the primary author, and collaborated with Brian Ball to decide which measurements to perform, program the network analysis scripts, interpret the results, and write the manuscript. Brian Karrer preprocessed and disambiguated the data, and Mark Newman helped in selecting which results to include and with manuscript revisions. This chapter is based on work published in *Physical Review E* [116].

CHAPTER II

Strategic cascades

2.1 Introduction

A common topic in the study of network behavior is that of contagious or *cascading* processes, in which a number of nodes, or *agents*, start with some property they then spread to their neighbors according to some specified propagation rules. This naturally represents phenomena such as the spread of trends, technologies, or influence among people or groups, or cascading failures in structures such as power grids or banks. Scientists have, for many years, observed that processes can be heavily influenced by the network on which they occur [167, 75, 74, 42]. This influence has been confirmed in the real world by experiments from a wide array of fields [41, 43, 110, 11] including the study of product adoption [16, 30, 115, 71].

Various models with simple spreading rules have been proposed [7, 131, 181] to explain, for example, how breaking news spreads over the Internet or how a new technology spreads in popularity. Such models can be roughly classified in two categories, according to whether the spread is defined directly as a stochastic process, or in terms of decisions by self-interested agents who derive utility based on their choices and the choices of others in the network. In the latter case, the cascade scenario can be framed as a *game*, and agent strategies cast as equilibria in the game. Due to the complexity of such games, however, typical agent-based cascade models assume that agents make decisions *myopically*, evaluating utility of alternative choices in the current state, without explicitly considering the future choices of others, nor their own potential impact on those choices.

Our goal in this research is to investigate the implications of more forward-looking, or *strategic* agent behavior. In what way do cascade patterns differ if agents behave strategically rather than myopically? How does the sophistication of agent decision making affect one's ability to influence a cascade process through scheduling of agent decisions?

2.1.1 Approach

To address these questions, we employ the framework of Chierichetti et al. [35], described in Section 2.2. This prior work presents many interesting results about cascade behavior of myopic agents, and demonstrates the striking power of a scheduler to influence myopic cascades. Under our new assumption of strategic behavior, we find that even many simple cases of this game, such as pairwise agent interactions on a line, seem intractable to analyze. Thus instead of solving the game generally, we take the approach of bounding the difference in cascade outcomes between myopic and strategic agent types. We find that cascade outcomes can be markedly different, as can the potential influence of a scheduler, depending on the particular network setting. We are able to obtain tight bounds through two easily analyzed graph families.

2.1.2 Results of Chierichetti et al. for myopic agents

Chierichetti et al. investigate a network of agents making choices between two options with positive externalities, Y and N, under the influence of a scheduler. (We adopt their model and describe it in detail in Section 2.2.) The primary contribution of these authors is in analyzing the impact of the *schedule*: the order in which agents make choices. They show that for any network there is some schedule which gets an expected constant fraction of the agents to choose Y. They also give networks and schedules which cause all but a constant number of agents to choose Y, in expectation. Lastly, they show that nonadaptive (fixed-sequence) schedules can obtain 50% expected Y-adoption at best.

2.1.3 Related work

Sequential voting and information cascades are two facets of a vast literature attempting to explain herd behavior [36]. Sequential voting models [3, 52] consider strategic agents aiming to choose the majority decision, but with an additional private preference. Information cascades [12, 21] consider strategic agents with a noisy signal, attempting to determine the correct choice. Both of these models tend to simplify network effects by placing agents on a complete graph.

Granovetter [74] introduces the *threshold model*, a foundational theory of network cascades which has since been studied and extended by many others [54, 160, 99]. In the threshold model, agents take an action if a certain number of their neighbors have taken the same action. Altman et al. [4] give an example of self-interested agents behaving in accordance with the threshold model, but in general self-interested behavior may not align with set thresholds.

Some agent-based cascade models [158, 131] allow agents to revise their decisions over multiple rounds of play. In each round, an agent myopically adopts its best choice in the current state. Some research on such models [24, 58] also introduces an element of noise in agent choice, and investigates the convergence of cascades over time.

Galeotti et al. [68] introduce a model with strategic agents which have access to incomplete information about the network outside their direct neighbors, thus making strategic agent behavior tractable. Lastly, Chierichetti et al. [35] introduce a cascade scheduling problem on networks based on a model studied by Arthur [7], which also assumes simple myopic agent decision making. Their model has been further extended by Cao et al. [31] and Hajiaghayi et al. [81].

2.2 Model

We model a *game* $F = (G, p, \pi)$ in which a collection of agents choose between two actions, Y ("yes") and N ("no"). Agents make their choices one at a time in a sequence determined by the *scheduler*. Once an agent has decided, it cannot change its action. We refer to the *collection of agents* as V, the total number of agents as $n = |V|$, an *individual agent* as $i \in V$, and the *choice* agent i makes as $c_i \in \{Y, N\}$. Agents are vertices on the finite simple graph $G = (V, E)$, and we say that two agents are *neighbors* if they are connected by an edge $e \in E$. We denote the set of neighbors of i by $nb(i) \subset V$.

Each agent i has a *preference type*, $t_i \in \{Y, N\}$, which is independently randomly assigned at the beginning of the game. An agent is assigned type Y with probability p (a game parameter) and type N with probability $1 - p$. We assume that Y is the less likely preference, so $p < .5$. Types are private: only i knows the value of t_i (until it is possibly revealed by i's choice).

Agents make their choices to maximize individual *utility*. An agent obtains utility π (a game parameter) for choosing its type ($c_i = t_i$), and a unit of utility for each neighboring node making the same decision that it does. Thus a node faces tension between choosing its type and the type it expects the majority of its neighbors to choose (when these types disagree). Formally, agent i's total utility is:

$$u_i(t_i, c_i, \mathbf{c}_{-i}) = \pi \mathbb{1}(c_i = t_i) + |\{j \in nb(i) : c_j = c_i\}|,$$

where $\mathbb{1}$ is the indicator function and vector $\mathbf{c}_{-i}$ represents the choice of all other nodes.

We differentiate between two modes of agent decision making: *myopic* and *strategic*. At the time agent i makes its decision, some nodes have already chosen and the remainder are undecided. A myopic agent makes its decision based on only the choices of decided nodes. It does not look into the future to consider the likely actions of undecided nodes, hence the term "myopic". Let $m_Y(i)$ and $m_N(i)$ denote the number of neighbors of i who have chosen Y and N, respectively, at the time i is scheduled to decide. Then a myopic i chooses t_i if $|m_Y(i) - m_N(i)| \leq \pi$, and the majority type among its decided neighbors otherwise.

A strategic agent aims to maximize its expected utility at the end of the game. We assume it knows the details of the game (G, p, and π), the schedule S (discussed below), and the decisions of already-decided agents. The agent reasons about the likely choices of undecided agents, assuming they all are strategic and play according to a *perfect Bayesian equilibrium* (PBE). In our setting, each agent moves exactly once and types are independent, so there is no relevant updating. Under these conditions, each node of the game tree is essentially a singleton information set, treatable as a *subgame*. Thus, the PBE concept here corresponds exactly to game solution by backward induction. To determine an agent's utility-maximizing action in some game, one can first solve for the choice of the last agent to move in all possible subgames with only one agent left to move. Knowing the choice of the last agent, one can solve for the choice of the penultimate agent in all subgames where all agents but two have moved. This reasoning can be repeated until the behavior of all agents in all subgames is known, yielding a PBE.

We make the additional assumption that an agent chooses its preference type, t_i, if it would otherwise be indifferent between options. We show that any game and schedule combination correspond to exactly one PBE consistent with this assumption (see Theorem 2.12). Thus the behavior of all strategic nodes is well defined.

Following Chierichetti et al. [35], our analysis includes a scheduler whose goal is to determine a schedule S that maximizes the expected number of agents choosing Y. A schedule determines the order in which agents make their decisions. We consider two classes of schedule: *nonadaptive* and *adaptive*. A nonadaptive schedule is simply a fixed ordering of nodes, that is, a permutation of V. An adaptive schedule, in contrast, can select the next agent to choose based on previous agent decisions. Formally, adaptive schedule S is a function of agent choices, $S : \{Y, N, U\}^n \rightarrow V$, where U indicates that the corresponding agent is as yet *undecided*.

We evaluate schedules by their *performance*, which is the expected number of nodes choosing Y once all have decided. An *optimal* schedule has the greatest performance among all schedules, or *optimal performance*. We use *strategic* and *myopic* to qualify performance. For example, a schedule's strategic performance is the performance of the schedule for strategic agents. A state of a game in progress, in which some but not necessarily all agents have decided, is a *situation*.

We say that a situation is a *Y-cascade* if every future agent chooses Y regardless of type. We similarly define an *N-cascade*. A situation is a *total cascade* if the first agent necessarily initiates a cascade of its type. A game is a *predetermined Y-cascade* if the starting situation is a Y-cascade. We similarly define *predetermined N-cascade*.

2.3 Roadmap of results

The main result of this chapter is a demonstration that cascade outcomes can vary drastically depending on the assumption of myopic or strategic agents. Specifically, we show that the difference in performance between myopic and strategic agents can be arbitrarily close to the maximum possible difference of 100% in either direction. In addition, we solve for equilibrium agent behavior in several particular game classes, provide miscellaneous results characterizing the behavior of cascade games with strategic agents, and show a result demonstrating the importance of the capabilities of the scheduler:

- In Section 2.4, we analyze the clique—both as a first example and as a way of introducing intuition, techniques, and results useful for subsequent sections. We present instances in which strategic performance is 0%: strictly worse than the constant expected adoption guaranteed for myopic agents. We conversely present instances for which strategic performance is greater than myopic performance.

- In Section 2.5, we show that myopic performance can be much larger than strategic performance: the difference can be arbitrarily close to 100%. We prove this by analyzing a specific class of games which occur on a graph we call a council graph.

- In Section 2.6, we show the converse: strategic performance can be arbitrarily close to 100% greater than myopic performance. We show this by analyzing a class of games which occur on a graph we call a cloud graph.

- In Section 2.7, we give several results. In particular we show that performance in the nonadaptive setting is bounded by p for both myopic and strategic agents. This improves upon the results of Chierichetti et al. [35] showing a myopic agent upper bound of $\frac{1}{2}$. We also demonstrate a family of graphs in which myopic performance is always at least as great as strategic performance, no matter the parameter settings.

- In Section 2.8, we investigate the commitment power of the scheduler and show that, in some cases, an ability to make non-credible threats can strictly enhance performance.

- Finally, in Section 2.9, we present an algorithm to compute the performance of a graph with strategic agents that is efficient on a certain class of highly symmetric graphs.

The complete analysis of some results and several lemma and theorem proofs are relegated to Appendix A.

2.4 Clique analysis

A *clique* is a complete graph where every two agents are connected. We begin our study of the difference between strategic and myopic performance with a description of behavior on the clique because it is illustrative of the difference between the myopic and strategic settings, and is used in subsequent proofs. The clique is also easier to analyze as nodes occupy indistinguishable positions in the network, rendering all schedules identical.

2.4.1 An example

Let $\pi = 1.1$, $p = 0.09$, and our graph be a clique of size 3. We name the nodes in the order that they are scheduled: 1, 2, and 3. Note that on a clique all nodes have the same neighbors, so all schedules are identical. We reason about the behavior of strategic agents in this game by backward induction.

First consider the behavior of the last node to choose, agent 3. If agents 1 and 2 have both chosen N or have both chosen Y, 3 will match with them. Otherwise, $c_3 = t_3$.

Next consider the behavior of agent 2. If $t_2 = N$, $c_2 = N$ no matter what. Even if $c_1 = Y$, agent 2 can expect to get a match from 3 with probability 0.91 if $c_2 = N$. Its expected payoff would be 2.01 for $c_2 = N$ versus 2 for $c_2 = Y$. If $t_2 = Y$, $c_2 = N$ if $c_1 = N$. This is because agent 2's expected payoff for $c_2 = N$ is 2, versus 1.19 for $c_2 = Y$.

Knowing this behavior, $c_1 = N$ regardless of t_1. Suppose $t_1 = Y$. Then agent 1 gets payoff 2 for $c_1 = N$, or payoff $1.1 + 0.91(0 + 0.09) + (0.09)2 = 1.3619$ for $c_1 = Y$, so is best off choosing N.

Even this very simple graph demonstrates a qualitative difference between strategic and myopic behavior. As Chierichetti et al. [35] show, the optimal schedule for any graph with myopic agents achieves at least a constant fraction of Y-adoption, in expectation. In this example, myopic agents achieve over 6.7% expected adoption. Yet for strategic agents, the example scenario yields zero adoption. We further characterize the behavior of the clique graph in Section 2.4.2.

2.4.2 Asymptotically large clique

We characterize the behavior of games on cliques in the limit of large clique size. For the remainder of this section, we assume p and π to be fixed and represent a game $F = (G, p, \pi)$ solely by its graph G. When G is a complete graph (clique) of size n, we use K_n. Games on cliques can be divided into two classes of asymptotic behavior.

Theorem 2.1. *For any fixed $0 < p < 1/2$ and $\pi > 0$, there exists an M such that for all $n \geq M$, K_n gives either:*

1. *A predetermined N-cascade (all agents always choose N), or*
2. *A total cascade (first agent chooses its type t and the remaining agents match t, starting a t-cascade).*

We denote these two classes of behavior by $\mathbf{C}_{PNC}$ and $\mathbf{C}_{TC}$. The class a particular game belongs to depends on p, π, and n. The proof of this theorem, in Appendix A.1, follows from the fact that, as cliques become very large, a node prefers any guaranteed cascade over a chance of being left out of a cascade.

We find cliques in both $\mathbf{C}_{PNC}$ (see Section 2.4.1) and $\mathbf{C}_{TC}$ (see below). $\mathbf{C}_{PNC}$ corresponds to cases where myopic agents give higher performance than strategic agents, and $\mathbf{C}_{TC}$ corresponds to the opposite. A clique transitions from $\mathbf{C}_{TC}$ to $\mathbf{C}_{PNC}$ as p decreases and π increases. On the boundary of this transition we find cases where a clique alternates, depending on the parity of n, between $\mathbf{C}_{PNC}$ and $\mathbf{C}_{TC}$. Computational confirmation of these results can be found in Section 2.4.3.

When Strategic Outperforms Myopic on a Clique Section 2.4.1 presents an example where strategic agents yield zero performance but myopic agents give positive performance. One might expect the clique to always favor myopic performance, as strategic agents are aware that Y-preference is less likely, and thus might be more likely to choose N than their myopic counterparts. We show that this is not the case. When $1 \leq \pi < 1 + p$, myopic agents underperform strategic agents because two Y decisions are required to start a myopic Y-cascade and only one is required to start a strategic Y-cascade. Thus the probability of a Y-cascade is $\frac{p^2}{p^2+(1-p)^2} \approx p^2$ for myopic agents and p for strategic agents.

Theorem 2.2. *For any $1 \leq \pi < 1 + p$, the probability of a Y-cascade with strategic users on a clique graph is p.*

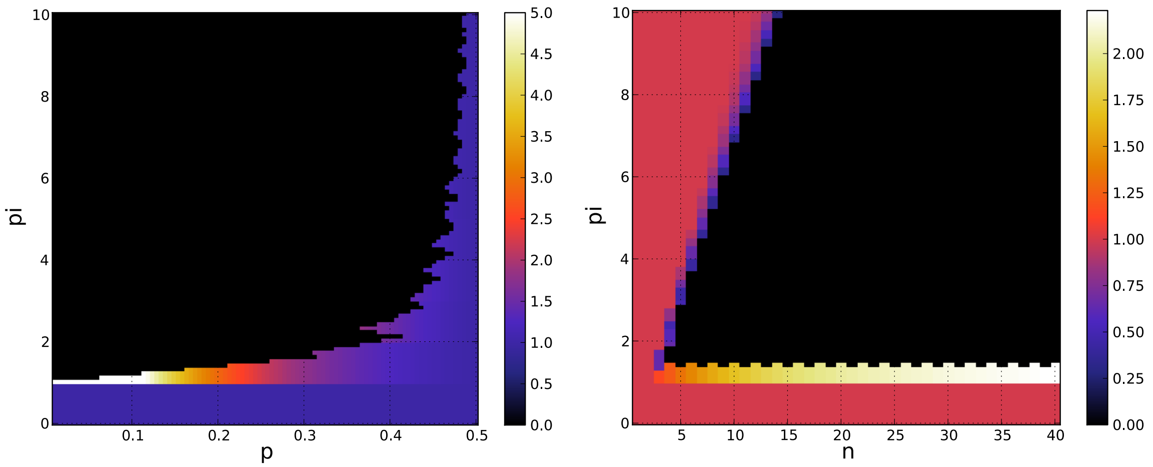
Note that when $\pi < 1$, strategic performance is equal to myopic performance by Lemma A.1.

Computational results in Section 2.4.3 suggest that for any $\pi \geq 1$, there exists settings of p such that strategic performance is greater than myopic performance.

2.4.3 Clique computational solution

In Section 2.4.1 we prove that, under some parameter settings for the clique, strategic agents result in a performance of zero. In Section 2.4.2 we prove that other parameter settings result in strategic agents outperforming myopic agents. In this section we provide computational verification for these two scenarios. For the specifics of our algorithm, please refer to Section 2.9.

Figure 2.1(a) displays results of a program which simulates a clique of 40 agents, each making the optimal strategic or myopic decision. We calculate the strategic and



(a) $n = 40$

(b) $p = 0.25$

Figure 2.1: Strategic-to-myopic performance ratio on clique graph.

myopic performance for a variety of p and π combinations and plot their ratio in the figure. The black region corresponds to the class $\mathbf{C}_{PNC}$ and the lighter regions correspond to the class $\mathbf{C}_{TC}$. The band at the bottom for $\pi < 1$ results from the immediate total cascade (Lemma A.1). The band just above $\pi = 1$ in Fig. 2.1(a) corresponds to the region partially described by Theorem 2.2, where one agent can start a strategic cascade but two agents are necessary for a myopic cascade. Figure 2.1(b) displays results of the same program, but with fixed p to examine the effect of varying n. The resulting black area is governed by two simple bounds. The lower pink area results from $\pi < 1$ according to Lemma A.1 as described above. The left pink wedge appears when π is large relative to n and all agents choose their preference. At the pink-black border we see non-trivial behavior.

2.5 Myopic outperforms strategic

We exhibit a setting where myopic performance is $(100 - \epsilon)\%$ but strategic performance is zero. Thus, unlike in the myopic case, where performance is always bounded above some constant [35], it is possible to get zero strategic performance while simul-

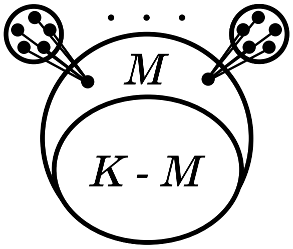


Figure 2.2: A council graph, as described in the text, with intra-clique connections excluded. The large circle is the council and the small circles are subcliques.

taneously having arbitrarily high myopic performance. This constitutes the first half of our core result. We prove this bound constructively, by characterizing the behavior of a family of graphs which have optimal strategic performance of 0% and an optimal myopic performance which approaches 100% in the limit of large graph size.

Our graph is a modified version of a clique graph, which we call a *council* graph (see Figure 2.2). It consists of a large clique, the council, of size K and M smaller subcliques of size 5.[1] M is $o(K)$, for example, $M = \sqrt{K}$. Each of the subcliques is completely connected to a unique node, its *representative*, from the council. This gives $K - M$ council nodes of degree $K - 1$, M council nodes of degree $K + 4$, and $5M$ subclique nodes of degree 5.

Near 100% Myopic Performance We demonstrate a schedule giving performance tending to 100% in the limit of large graph size. We do not prove this schedule's optimality, but it gives a lower bound sufficient for our purposes. We say a subclique

[1] Any subclique of constant size ≥ 5 will work.

is *fresh* if none of its nodes have been scheduled. Our schedule, S, is the following:

```
1 Choose any fresh subclique, $j$.
        Schedule nodes from $j$ until one chooses $N$ or all have chosen $Y$.
        If all nodes in $j$ have chosen $Y$, schedule $j$'s council representative, $r_j$.
2 Repeat 1 until three representatives have been scheduled or no fresh subcliques
  remain.
3 Schedule all council nodes without $N$-decided neighbors.
4 Schedule all remaining council nodes in any order.
5 Schedule all remaining subclique nodes in any order.
```

Theorem 2.3. *For any $p < .5$, $2 < \pi < 3$, the myopic performance of S approaches 100% as $K \to \infty$.*

Proof. Our proof proceeds by a careful description of behavior at each point in the schedule. First note that a myopic agent will choose its type if ≤ 2 of its neighbors have been scheduled, and Y if ≥ 3 more of its neighbors have chosen Y than N.

So with probability p^3 a fresh subclique from Step 1 will be a Y-cascade and its representative will also choose Y. With probability $1 - p^3$ a fresh subclique from Step 1 will not be a Y-cascade and its representative will not be scheduled.

We can bound the probability of (the undesirable event of) not having 3 subclique Y-cascades by:

$$\binom{M}{2}(p^3)^2(1 - p^3)^{M-2} + Mp^3(1 - p^3)^{M-1} + (1 - p^3)^M \ < \ M^2(1 - p^3)^{M-2}.$$

Once there are 3 subclique Y-cascades, the entire council chooses Y. Thus, the expected fraction of Y is at least:

$$K(1 - M^2(1 - p^3)^{M-2})/(K + M),$$

which tends to 1 as $K \to \infty$. □

0% Strategic Performance We characterize behavior of the council graph with strategic agents and prove that certain choices of p and π give 0% performance. This, together with the results from above, gives a tight bound on the extent to which myopic performance can be greater than strategic performance.

Theorem 2.4. *For some $p < .5$, $2 < \pi < 3$, any schedule on a council graph with strategic agents has 0% performance.*

The following lemma is used in the proof below.

Lemma 2.5. *For some $p < .5$, $2 < \pi < 3$, a clique of $k \geq 5$ undecided nodes and one node guaranteed to choose Y will result in k N-decisions and 1 Y-decision.*

We prove this lemma in Appendix A.2 by direct comparison of expected utilities.

Proof of Theorem 2.4. The council graph was chosen to facilitate analysis by simplification to more easily understood cliques. As such, we invoke Lemma 2.5, which proves the existence of cliques and settings of p and π which strongly favor N, in the sense that even if a common neighbor of the clique is guaranteed to choose Y, the bias towards N-preference results in a predetermined N-cascade.

Thus, even if a clever scheduler convinces the whole council to choose Y, we see by Lemma 2.5 that, for some p and π, the subclique chooses N.

The nodes in the council, being fully strategic, know any subclique neighbors they have are guaranteed N-neighbors. By another application of Lemma 2.5, we see that all council nodes choose N and thus, for some p and π, any schedule is doomed to 0% performance. □

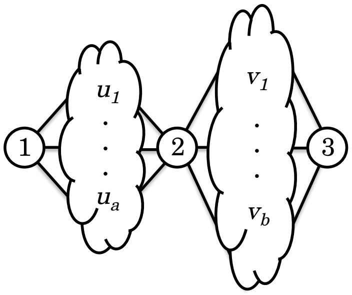


Figure 2.3: An example cloud graph.

2.6 Strategic outperforms myopic

By now it is natural to see how strategic agents' expectations of future N-preference lead to lower strategic performance than myopic performance. As seen in Section 2.4.2, there are also games where strategic agents give higher performance than myopic agents. In this section we show the second half of our core result, that this difference can be as large as $(100-\epsilon)\%$.

We prove this bound constructively by analyzing a special graph we call a *cloud* graph. We first show that for certain parameters on the cloud graph it is possible to obtain strategic performance of 100%. Recall that no graph can achieve exactly 0% or 100% myopic performance, because the first myopic node always chooses its type. We do, however, show that for some settings on the cloud, myopic performance approaches 0% while strategic performance remains 100%, giving our desired bound.

A cloud graph (Figure 2.3) consists of two singular *outer* vertices of degree a and b respectively, one singular *inner* vertex of degree $a+b$, and two *clouds* of vertices of respective size a and b, with each vertex of degree two. Each of the outer vertices is connected to every vertex in a distinct cloud. The inner vertex is connected to every vertex in both clouds. We call the cloud with a vertices A and the one with b vertices B.

Theorem 2.6. *Fix arbitrary $\epsilon > 0$. Then there exist parameters a, b, p, and $\pi < 2$ such that strategic performance on the cloud graph is 100% whilst the myopic perfor-*

mance is at most ϵ.

This follows from Lemma 2.7 and Lemma 2.11, proved in the remainder of this section.

100% Strategic Performance We give sufficient conditions for obtaining 100% performance with strategic agents in a cloud graph. We refer to the singular vertices, from left to right, as $1, 2$, and 3, and assume that $a < b$.

An optimal schedule, S_{opt}, is the following:

```
Schedule 1.
if 1 chooses Y then
    Schedule 2 followed by 3.
else
    Schedule 3 followed by 2.
Schedule all cloud vertices in any order.
```

An overview of the proof of optimality is as follows. Scheduling agent 2 before agent 3 guarantees that 2, and thus all nodes in A, will match the choice of 1 (Lemma 2.10). On the other hand, scheduling agent 3 before agent 2 gives some positive probability that the nodes in A choose their type (Lemma 2.9). This outcome results in lower utility for 1. Thus the adaptive schedule can be used to incentivize 1 to choose Y through threat of punishment for choosing N. We are able to show that, for large enough cloud sizes, threat of punishment to 1 for choosing N is enough to convince it to choose Y, giving 100% performance. This is given formally below.

Lemma 2.7. *If cloud sizes satisfy $a(1-p) + \pi < bp$ and $ap^2 > \pi$, then, under the schedule S_{opt}, 1, and thus all agents, will always choose Y.*

Proof. We are able to punish 1 for choosing N because scheduling 3 first gives a p^2 chance of the nodes in A choosing their type, while scheduling 2 first guarantees all nodes in A choose c_1, a more desirable outcome to 1. We use several lemmas outlining the behavior of agents 2 and 3, proved below.

Assume 1 is N-type. Being Y-type only increases 1's payoff for choosing Y. We show below that 1's utility for choosing Y is higher than its utility for choosing N when $a(1-p)+\pi < bp$ and $ap^2 > \pi$.

If 1 chooses N, then we schedule 3, followed by 2. In this case, Lemma 2.9 shows that $c_2 = c_3 = t_3$. With probability $1-p$ all nodes in A choose N, and with probability p A is split. 1's expected utility is $\pi + a(1-p) + a(1-p)p$.

If 1 chooses Y, then we schedule 2, followed by 3. In this case, Lemma 2.10 shows that $c_2 = c_1$. 1's expected utility is a.

Then 1 will choose Y as long as $a > \pi + a(1-p)(1+p)$. Or, equivalently, $ap^2 > \pi$, which is true by assumption. Once 1 chooses Y, the schedule leads to all remaining nodes choosing Y. □

We must pick appropriate cloud sizes (depending on π and p) and have $\pi < 2$ for the theorem to be true. This is possible for any p by selecting large enough a and even larger b. The following lemmas detail the behaviors of the clouds and agents 2 and 3 used in the proof of Lemma 2.7.

Lemma 2.8. *The behavior of an unscheduled cloud neighbored by two decided singular agents is completely determined by the singular agents' choices. If their choices are different, then every cloud agent will choose its type and an expected* p *fraction of the cloud agents will choose* Y*. In this case we say that the cloud has been* split*. If they make the same choice* c*, all cloud agents will choose* c*.*

Knowing the cloud behavior, we can characterize the behavior of the case where agent 3 is scheduled and then agent 2 is scheduled.

Lemma 2.9. *When* $a(1-p)+\pi < bp$*, if agent 3 is scheduled to choose and 2 has not been scheduled yet,* $c_3 = t_3$*. Then, when 2 is scheduled next,* $c_2 = c_3$*.*

Lemma 2.10. *If agent 2 is scheduled to choose after 1 but before 3, it will choose* c_1 *if* $bp > ap > \pi$*. When 3 is scheduled, it will match 2.*

Here, the scheduler persuades cloud agents to adopt the minority preference through the threat of unfavorable adaptive sequencing. All nonadaptive schedules have performance bounded by p (Theorem 2.17), and thus the above result clearly requires adaptivity. In fact, the difference in performance for adaptive and nonadaptive scheduling can be arbitrarily large for strategic agents (Corollary 2.21).

Near 0% Myopic Performance By Lemma 2.7, we can obtain 100% adoption with strategic agents for any p if we pick cloud sizes a and b large enough. The proportion of myopic adoption, however, is some polynomial of p, and thus can be made arbitrarily small. The combination of these two results gives us a tight bound on the extent to which strategic performance can be greater than myopic performance.

Lemma 2.11. *Fixing $\pi < 2$, the myopic performance is bounded by $p(1-p)^3(1-\frac{p}{1-p})^2 + [1-(1-p)^3(1-\frac{p}{1-p})^2]$. In the limit of $p \to 0$, the proportion of myopic adoption in the cloud graph also goes to 0.*

Proof. We bound the myopic performance by a polynomial in p for $\pi < 2$.

Denote the current difference between the number of agents in cloud A (B) who have chosen Y and those who have chosen N by d_A (d_B). With probability $(1-p)^3$, all singular agents are type N. A singular N-type agent will choose Y only if $d_A \geq 1$ or $d_B \geq 1$. A cloud agent will choose its type or N unless at least one of the singular agents has already chosen Y.

We bound the probability of a singular agent choosing Y by noticing that the probability of a cloud ever achieving a Y majority by agents choosing their type is no greater than $\frac{p}{1-p}$, a result from the mathematics of biased random walks. Thus, the probability that all cloud nodes choose their type (or N) is at least $(1-p)^3(1-\frac{p}{1-p})^2$. This probability, which we denote q, tends to 1 as $p \to 0$. The expected proportion of Y-adoptions is no greater than $pq + (1-q)$, which goes to 0 as $p \to 0$. □

2.7 Miscellaneous results

Having analyzed behavior of cascades on specific classes of graphs, we aim to give properties of cascade behavior for arbitrary games, regardless of G, p, or π.

We first address the issue of the multiplicity of PBE. The existence or uniqueness of PBE is not guaranteed for all classes of games. The possibility of zero or multiple PBE would render some of our key concepts, such as the performance of a schedule, unclear. Fortunately, as we show below, our assumption that agents consistently choose their type when indifferent between options always results in the selection of a unique PBE. This follows from a simple backward induction argument. We also give a technique for relaxing this behavioral assumption but keeping the unique PBE property:

Theorem 2.12. *If agents are never indifferent between choices, or always resolve any indifference in a consistent way—by choosing the same option whenever they are in the same situation—then their PBE behavior is uniquely defined.*

The following theorem shows that our assumption of indifference can be avoided, while keeping the same cascade outcome, by slightly adjusting π.

Theorem 2.13. *Given a game $F = (G, p, \pi)$, let P be the performance under an adaptive schedule S with the assumption that a node always chooses its type if it is indifferent between Y and N. Then there exists an ϵ such that $F' = (G, p, \pi' = \pi + \epsilon)$ also achieves performance P under S, and under S no node is indifferent between choices.*

The proof, in Appendix A.3, simply chooses ϵ less than the smallest utility difference.

We also prove that increasing p alone can never decrease the performance of the optimal schedule. The main ingredient in the proof is a coupling argument. This monotonicity is not observed for π or n.[2]

Theorem 2.14. *For any two games, $F = (G, p, \pi)$ and $F' = (G, p', \pi)$, with $0 < p < p' < .5$, the performance of any nonadaptive schedule S for F is weakly worse than the performance of S for F'.*

Star Graph It seems that behavior on every graph we study varies unpredictably as game parameters change. Even a graph as simple as a clique can exhibit drastically different cascade outcomes from small changes in p or π. However, this is not always the case. Games on the *star* graph—a graph with one *interior* agent (of degree $n-1$) connected to $n-1$ *exterior* agents (of degree 1)—have notably regular behavior.

Theorem 2.15. *For any parameters on any star graph, the optimal performance in the myopic setting is at least the optimal performance in the strategic setting for both adaptive and nonadaptive schedules.*

The proof, in Appendix A.4, establishes the optimality of threshold strategies for nodes and then shows that the myopic thresholds always beat the strategic thresholds.

Knowing that myopic performance exceeds strategic on the star, we next explore the degree of this advantage. We find that, for adaptive schedules, myopic performance can be arbitrarily close to an additive factor of 50% greater than strategic performance.

In the limit of large star graphs, adaptive myopic performance is $\frac{p}{1-p}$ for any $\pi < 1$. Thus, for p arbitrarily close to .5, myopic performance approaches 100%. This result does not hold for strategic agents: a backward induction argument shows that for small enough π, strategic performance is bounded by p.

Theorem 2.16. *For any star graph with $\pi < 1 - p$ and any adaptive schedule, strategic Y-type nodes choose N when a majority of nodes have chosen N, upper bounding strategic performance by p.*

[2] An example of non-monotonicity can be found in Figure 2.1(b).

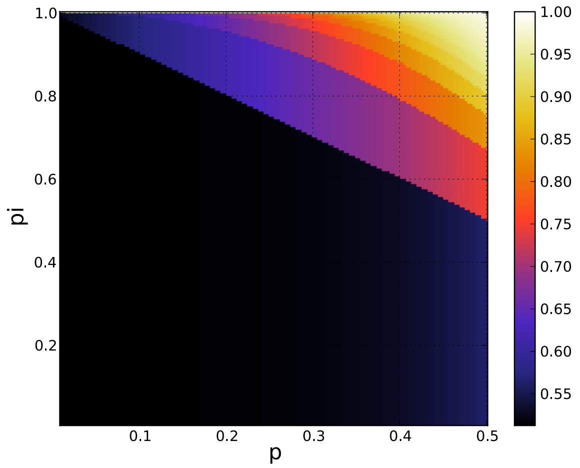


Figure 2.4: Strategic-to-myopic performance ratio on star, $n = 41$.

Proof. We characterize behavior on the star with strategic agents when $\pi < 1 - p$ by backward induction. Let d be the difference between the number of exterior nodes who have chosen Y and the number who have chosen N: $d = m_Y - m_N$. We show that any node chooses N when $d < 0$.

Consider the behavior of the last node, i. Assume the best case for a Y choice, that $t_i = Y$. If $d = -1$, i receives $p + \pi$ utility for $c_i = Y$ and 1 utility for $c_i = N$. By assumption, i prefers N. Any $d < -1$ gives i π utility for $c_i = Y$ and 1 utility for $c_i = N$. This completes the base case.

Next we prove the inductive step. Assuming the theorem holds when $k - 1$ agents remain, we show the theorem holds when k agents remain. Denote the agent choosing with k agents remaining by j. Assume the best case for a Y choice, that $t_i = Y$. Using the inductive hypothesis, we find that utilities for j are exactly as above for i. □

We proved that optimal strategic performance is never greater than optimal myopic performance for a star graph. We also present computational verification. For details of the algorithm used for computing solutions, see Section 2.9.

Figure 2.4 displays results of a computational solution of the optimal schedule for a star of 41 agents. We calculate the strategic-to-myopic performance ratio for a variety of p and π combinations.

Nonadaptive Schedules Lastly, we prove that no nonadaptive schedule for strategic or myopic agents can achieve more than a p fraction performance, on any graph. A similar bound of p was proved independently by Hajiaghayi et al. [81, Theorem 1], using different techniques, restricted to myopic agents on the clique graph. Their theorem generalizes to the setting where agents can have heterogeneous π thresholds. Whereas we do not explicitly address heterogeneity in π here, we note that the proof of Theorem 2.17 immediately extends to this more general model.

Both results improve on the 50% bound of Chierichetti et al. [35], which covers myopic agents on arbitrary graphs.

Bounding nonadaptive schedule performance for strategic agents entails that the very high performance of Section 2.6 is not possible when the scheduler cannot react to decisions made by nodes (Corollary 2.21). Moreover, it rules out the possibility of predetermined Y-cascades with nonadaptive schedules. Our proof combines a careful inductive argument with the repeated application of a result from the analysis of Boolean functions.

Theorem 2.17. *No nonadaptive schedule can achieve more than p fraction performance for any $p \leq .5, \pi > 0$, in the myopic or strategic setting.*

To prove Theorem 2.17 we use a lemma from Mossel et al. [132, Lemma 5.1]:

Lemma 2.18. *Let $f : \{Y, N\}^n \to \{Y, N\}$ be a monotone function (so that flipping input bits from N to Y cannot change the output from Y to N and vice versa) and $P_p(f = Y)$ be the probability that f outputs Y when applied to random inputs each Y with probability p. Also, let $P_{1/2}(f = Y) = 1/2$. Then $P_p(f = Y) \leq p$ for all $0 \leq p < 1/2$.*

Proof of Theorem 2.17. We first prove the theorem for the myopic setting. We apply Lemma 2.18 separately to each node, in combination with Lemma 2.19, to show that each agent chooses Y with probability at most p. From linearity of expectations, we know the myopic performance is at most a p fraction of the nodes.

Fix a game and schedule, and adopt the following notation.

Let $c_i : \{Y, N\}^i \to \{Y, N\}$ be the function which takes as input the types of the first i agents and outputs the selection of agent i.

Let $\mathring{c}_i : \{Y, N\}^i \to \{Y, N\}^i$ be the function which takes as input the types of the first i agents and outputs the selection of the first i agents.

Let $\hat{c}_i : \{Y, N\}^{i-1} \times \{Y, N\} \to \{Y, N\}$ be the function which takes as input the selections of the first $i - 1$ agents and the type of the ith agent and outputs the selection of the ith agent.

We denote the types of the first i agents as $t^{(i)} = t_1, \ldots, t_i \in \{Y, N\}^i$ and denote by $\neg w \in \{Y, N\}^i$ the string with each coordinate the opposite as in $w \in \{Y, N\}$.

Lemma 2.19 shows that $c_i(\neg t^{(i)}) = \neg c_i(t^{(i)})$, from which we see that $P_{1/2}(c_i = Y) = 1/2$ because for each string, exactly one of w and $\neg w$ evaluates to Y. Thus we can employ Lemma 2.18 to see that $P_p(c_i = Y) \leq p$, which proves the theorem in the myopic case.

Lemma 2.19. $c_i(\neg t^{(i)}) = \neg c_i(t^{(i)})$.

Proof of Lemma 2.19. We can do this by induction on i to show that both c_i and $\mathring{c}_i$ have this property. Note that because Y and N are treated symmetrically in the myopic setting, we know that $\hat{c}_i(\neg w) = \neg \hat{c}_i(w)$ for all i.

The base case follows because $c_1(t_1) = \mathring{c}_1(t_1) = \hat{c}_1(t_1)$, and we know that $\hat{c}_1$ has the property.

Assume that the statement is true for all $j < i$. Note that

$$\begin{aligned} c_i(\neg t^{(i)}) &= \hat{c}_i(\mathring{c}_{i-1}(\neg t^{(i-1)}), \neg t_i) \\ &= \hat{c}_i(\neg \mathring{c}_{i-1}(t^{(i-1)}), \neg t_i) \\ &= \neg \hat{c}_i(\mathring{c}_{i-1}(t^{(i-1)}), t_i) = \neg c_i(t^{(i)}). \end{aligned}$$

The first line follows from the definition of c_i and $\hat{c}_i$ and second line follows from induction. Similarly, for $\mathring{c}_i$:

$$\begin{aligned} \mathring{c}_i(\neg t^{(i)}) &= \mathring{c}_{i-1}(\neg t^{(i-1)}) \circ \hat{c}_i(\mathring{c}_{i-1}(\neg t^{(i-1)}), \neg t_i) \\ &= \neg \mathring{c}_{i-1}(t^{(i-1)}) \circ \hat{c}_i(\neg \mathring{c}_{i-1}(t^{(i-1)}), \neg t_i) \\ &= \neg \mathring{c}_{i-1}(t^{(i-1)}) \circ \neg \hat{c}_i(\mathring{c}_{i-1}(t^{(i-1)}), t_i) = \neg \mathring{c}_i(t^{(i)}). \end{aligned}$$

□

We next prove the strategic case of Theorem 2.17. The intuition is straightforward. If a node imagines that all future nodes are equally likely to prefer Y and N, then again Y and N are treated symmetrically, as in the myopic setting, and Lemma 2.18 applies. So given that this node's type and the types of agents that have already chosen are Y independently with probability p, the probability that each node chooses Y is at most p. This probability only decreases when this node expects future nodes to be Y-type less often.

We define $c_i^p : \{Y, N\}^i \to \{Y, N\}$, $\mathring{c}_i^p : \{Y, N\}^i \to \{Y, N\}^i$, and $\hat{c}_i^p : \{Y, N\}^{i-1} \times \{Y, N\} \to \{Y, N\}$ analogously to above, except here we assume that all agents play strategically according to the case where each node is Y-type with probability p.

The outline of the proof of Lemma 2.20, given in full in Appendix A.3, is as follows. We again see that $P_{1/2}(c_i^{1/2}(t^{(i)}) = Y) = 1/2$ by the same reasoning, and applying Lemma 2.18 we see that $P_p(c_i^{1/2}(t^{(i)}) = Y) \leq p$. We would like to show that $P_p(c_i^p(t^{(i)}) = Y) \leq p$. To complete the lemma it is enough to show that c_i^p is monotone in p. That is, increasing p only makes a Y outcome more likely.

By induction we will show that c_i^p is also monotone with respect to p. This completes the proof of the theorem because then $P_p(c_i^p(t^{(i)}) = Y) \leq P_p(c_i^{1/2}(t^{(i)}) = Y) \leq p$.

Lemma 2.20. *c_i^p and $\mathring{c}_i^p$ are also monotone in their inputs and in p.*

This completes the proof of Theorem 2.17. □

Theorems 2.17 and 2.6 imply that adaptive schedules can be arbitrarily more powerful than nonadaptive ones in the strategic setting.

Corollary 2.21. *For any $\epsilon > 0$, there exists a game F with strategic agents for which an adaptive scheduler can achieve 100% adoption and a nonadaptive scheduler achieves $\leq \epsilon$%.*

Similarly we see that adaptive schedules can be arbitrarily more powerful than nonadaptive ones in the myopic setting by combining Theorem 2.17 and a lemma from Chierichetti et al. [35, Lemma 3.1], which gives a graph with adaptive performance of $(100 - O(\frac{1}{pn}))\%$.

Corollary 2.22. *For any $\epsilon > 0$, there exists a game F with myopic agents for which an adaptive scheduler can achieve $\geq (100 - \epsilon)\%$ adoption and a nonadaptive scheduler achieves $\leq \epsilon$%.*

2.8 Scheduler commitment power

Our model dictates that the scheduler chooses and publishes its (possibly adaptive) schedule in advance. This publication is a commitment to follow the schedule even in situations where, once reached, it is suboptimal. We refer to a scheduler who can commit in advance as *Stackelberg*, after the classic economic model of imperfect

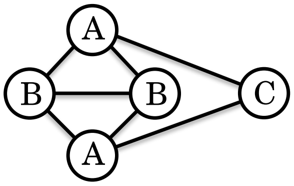


Figure 2.5: A graph in which a Stackelberg scheduler achieves greater performance.

competition in which a first-moving player is notably advantaged by an ability to make *non-credible* threats [175]. Such ability contrasts with a scheduler who is restricted to schedules that make the performance-maximizing decision in every subgame.

Whereas the power to make non-credible threats allows players to obtain strictly greater utility in some games, there are many natural games for which this power yields no advantage. Our question is whether in this context Stackelberg scheduling ability is strictly more powerful than the ability to only make credible threats. A priori, it is unclear how non-credible threats could aid the scheduler. It seems that the only way to convince a node *not* to choose N is to threaten to surround it with an abundance of Ys in the case where it does choose N. Maximizing Ys, however, aligns with the scheduler's goal, and can only be non-credible if somehow concentrating these Ys lowers overall expected performance.

We have, however, found a game instance, illustrated in Figure 2.5, for which commitment power provides an advantage. For this graph, with parameters $p = 0.18$ and $\pi = 1.85$, a Stackelberg scheduler can achieve performance of 0.573 whereas the best subgame-optimal schedule yields 0.371.

Our five node graph has three types of nodes which are in indistinguishable positions. We call the groups A, B, and C and don't distinguish between nodes within each group. We give the Stackelberg schedule in Figure 2.6(a) and the subgame-optimal schedule, which corresponds to a Perfect Bayesian Equilibrium (PBE), in Figure 2.6(b).

To see how the Stackelberg scheduler outperforms the PBE scheduler, note that

```
Schedule $A$:
if $A$ chooses $Y$ then
    Schedule $A$, then remaining nodes in any order
else
    Schedule $C$:
    if $C$ chooses $Y$ then
        Schedule $A$, then $B$, then $B$
    else
        Schedule $B$:
        if $B$ chooses $Y$ then
            Schedule $B$, then $A$
        else Schedule $A$, then $B$
```

(a) Stackelberg schedule

```
Schedule $C$:
if $C$ chooses $Y$ then
    Schedule $B$:
    if $B$ chooses $Y$ then
        Schedule $A$. All remaining nodes choose $Y$
    else
        Schedule $A$, then $A$, then $B$
else
    Schedule remaining nodes in any order. All choose $N$
```

(b) PBE schedule

Figure 2.6: Schedules demonstrating the increased power of non-credible threats by the scheduler.

the Stackelberg scheduler schedules A first and it chooses its type, whereas if the PBE scheduler scheduled A first it would choose N. Both schedulers agree on what to do if A chooses N. If A chooses Y, the Stackelberg scheduler schedules A next even though scheduling B next would yield higher expected performance. The PBE scheduler must pick B next in this case. The higher performance of picking B next comes at the cost of giving fewer expected Y-matches to A, and thus makes A, if scheduled first, less inclined to play Y. In this instance, the result is that a first-moving A would play N if faced with a PBE schedule, and its type if faced with the optimal Stackelberg schedule.

Since it cannot threaten A, the PBE scheduler does not schedule A first, and instead starts with C, which gives fairly similar cascade behavior but results in fewer nodes choosing Y, in expectation. This completes our example of a graph with higher Stackelberg than PBE performance.

Assuming commitment power in the foregoing analysis simplifies our arguments by avoiding the necessity of verifying optimal scheduling in all subgames. Results for myopic outperforming strategic hold *a fortiori* if we relax the assumption of commitment power, as the ability to make threats is useful only for strategic agents. Our derivation (Section 2.6) of the bound for strategic outperforming myopic exploits commitment power, however, we have verified that a more complicated demonstration can be constructed supporting the same bound under the weaker assumption of subgame-optimal PBE schedules.

2.9 Computational solutions to strategic cascades

It is straightforward to write a program which computes, by brute force, the optimal[3] schedule for an arbitrary graph. Node behavior can be solved by backward induction. Logically, the exponential number of possible schedules and agent type configurations makes this approach infeasible. In this section we describe an approach to efficiently find solutions for strategic cascade problems on a subclass of highly symmetric graphs.

Our code[4] finds the optimal schedule for arbitrary *block model*[5] graphs with strategic or myopic agents. Block models have been studied extensively in the past [169, 179] as a natural framing of networks in which nodes can be divided into classes or types with shared characteristics. We define them in Section 1.2.8. For example, a block model describing a social network at a high school could have a type for each grade. Students would be more likely to have edges to students of their same grade, and less likely to have edges to students of other grades. More abstractly, the star graph is easily described as a block model in which the two classes are "interior agent" and

[3] Our algorithm calculates the optimal schedule under the assumption that the scheduler is acting according to a PBE and *cannot* make empty threats. The main theoretical analyses of this chapter assume the scheduler *can* make empty threats and is acting according to a Stackelberg equilibrium. On the star and clique these two equilibrium concepts give identical optimal schedules. See Section 2.8 for in-depth discussion.

[4] The work in this section was performed in collaboration with Erik Brinkman. Code can be found at `https://github.com/tbmbob/block-scheduling`. `block_dp.py` is the file containing the solver. Code has not been prepared for public release. Please contact `travisbm@umich.edu` with any questions.

[5] Any graph can be expressed as a block model graph with n blocks. Our algorithm works efficiently only for graphs with a small number of blocks.

“exterior agent”.

Our code finds optimal schedules efficiently for graphs with a small (constant) number of types. The star, clique, and cloud graphs all fit this requirement. Running time is polynomial in the number of agents and exponential in the number of blocks. The code solves for the optimal performance through a combination of dynamic programming and backward induction. It first solves all possible scenarios with one node left to choose and stores the results. Then, by using these results, the program solves optimal behavior when there are two nodes left to choose. It continues this process until it solves for the optimal behavior with all nodes left to choose. It avoids the exponential running time of a naive backward induction by treating all agents within a block the same. It is then able to consider only which *block* to schedule next, not which node to schedule next. By reasoning over blocks instead of node types, the scheduler needs only to compare between $O(b)$ choices at each step, where b is the number of blocks, instead of $O(n)$ choices.

Examples of data gathered from our code can be viewed in Figures 2.1 and 2.4. This computational method is far from a panacea. Very few real-world graphs follow strict block models, and even idealized graphs often have too many types to permit efficient simulation. However, this code has been useful in verifying results for simple star, clique, and cloud graphs and in suggesting further results. For example, simulation on the clique suggested the possibility of certain parameter spaces resulting in higher strategic performance than myopic performance.

2.10 Conclusions

We have demonstrated that the common assumption of myopic decision making by agents participating in cascades can have significant consequences. For the specific model of Chierichetti et al., we find that assuming strategic instead of myopic agent decision making leads to markedly different cascade behavior. We show, by counterexample, that their result of linear performance for any graph does not apply when agents are strategic. We have identified graphs for which the performance difference between myopic and strategic agents is (asymptotically) as large as possible, in either direction. More broadly, we illustrate methods for reasoning about strategic cascade behavior and characterize the contrasting behavior of strategic and myopic agents in a range of qualitatively distinct settings. Lastly, we prove some results for strategic agents on general graphs, and demonstrate the power of scheduler commitment.

Modeling cascades with perfectly strategic agents is not necessarily more realistic than modeling agents with limited rationality. Thus, I do not argue for the strategic behavior we characterize as a definitive predictive model. Rather, my point is to demonstrate the potential impact of alternative assumptions about agent decision making on networks. It is likely that typical network decision making lies somewhere between myopic and strategic, and by characterizing the behavioral poles I hope to provide guidance for understanding the range within. Of course, substantial work remains to achieve a full understanding of behavior between these poles.

I consider cascades to be representative of a broader class of scenarios involving dynamic decision on networks. For these too one should expect the spectrum of behaviors, myopic to strategic, to exhibit qualitative variety in generated outcomes.

CHAPTER III

Nonbacktracking centrality

3.1 Introduction

I introduce network centrality measures in Section 1.2.5. Eigenvector centrality and its variants are some of the most widely used of all centrality measures. They are commonly used in social network analysis [180] and form the basis for ranking algorithms such as the HITS algorithm [102] and the eigenfactor metric [20].

As we argue in this chapter, however, eigenvector centrality also has serious flaws. In particular, we show that, depending on the details of the network structure, the leading eigenvector of the adjacency matrix can undergo a localization transition in which most of the weight of the vector concentrates around one or a few nodes in the network. While there may be situations, such as the solution of certain physical models on networks or searching for network hubs, in which localization of this kind is useful or at least has some scientific interest, in the present case it is undesirable, significantly diminishing the effectiveness of the centrality as a tool for quantifying the importance of nodes. Moreover, as we will show, localization can happen under common real-world conditions, for instance in networks with power-law degree distributions.

As a solution to these problems, we propose a new centrality measure based on the leading eigenvector of the Hashimoto or nonbacktracking matrix [83, 105]. This measure has the desirable properties of (1) being closely equal to the standard eigenvector centrality in dense networks, where the latter is well behaved, while also (2) avoiding localization, and hence giving useful results, in cases where the standard centrality fails. Overall, we contribute to network science knowledge by explaining a irregularity in a popular centrality measure and providing a new related measure fixing this irregularity.

3.2 Localization of eigenvector centrality

A number of numerical studies of real-world networks have shown evidence of localization phenomena in the past [65, 69, 62, 44, 72]. In this chapter we formally demonstrate the existence of a localization phase transition in the eigenvector centrality and calculate its properties using techniques of random matrix theory.

The fundamental cause of the localization phenomenon we study is the presence of *hubs* within networks, nodes of unusually high degree, which are a common occurrence in many real-world networks [14]. Consider the following simple undirected network model consisting of a random graph plus a single hub node, which is a special case of a model introduced previously in [134]. In a network of n nodes, $n-1$ of them form a random graph in which every distinct pair of nodes is connected by an undirected edge with independent probability $c/(n-2)$, where c is the mean degree. The nth node is the hub and is connected to every other node with independent probability $d/(n-1)$, so that the expected degree of the hub is d. In the regime where $c \gg 1$ it is known that (with high probability) the spectrum of the random graph alone has the classic Wigner semicircle form, centered around zero, plus a single leading eigenvalue with value $c+1$ and corresponding leading eigenvector equal to the uniform vector $(1,1,1,\ldots)/\sqrt{n}$ plus random Gaussian noise of width $\mathrm{O}(1/\sqrt{n})$ [172]. Thus the eigenvector centralities of all vertices are $\mathrm{O}(1/\sqrt{n})$ with only modest fluctuations. No single node dominates the picture and the eigenvector centrality is well behaved.

If we add the hub to the picture, however, things change. The addition of an extra vertex naturally adds one more eigenvalue and eigenvector to the spectrum, whose values we can calculate as follows. Let $\mathbf{X}$ denote the $(n-1)\times(n-1)$ adjacency matrix of the random graph alone and let the vector $\boldsymbol{a}$ be the first $n-1$ elements of the final row and column, representing the hub. (The last element is zero.) Thus the full adjacency matrix has the form

$$\mathbf{A} = \begin{pmatrix} \boxed{\mathbf{X}} & \boxed{\boldsymbol{a}} \\ \boxed{\boldsymbol{a}^T} & 0 \end{pmatrix}. \tag{3.1}$$

Let z be an eigenvalue of $\mathbf{A}$ and let $\boldsymbol{v} = (\boldsymbol{v}_1|v_n)$ be the corresponding eigenvector, where $\boldsymbol{v}_1$ represents the first $n-1$ elements and v_n is the last element. Then, multiplying out the eigenvector equation $\mathbf{A}\boldsymbol{v} = z\boldsymbol{v}$, we find

$$\mathbf{X}\boldsymbol{v}_1 + v_n\boldsymbol{a} = z\boldsymbol{v}_1, \qquad \boldsymbol{a}^T\boldsymbol{v}_1 = zv_n. \tag{3.2}$$

Rearranging the first of these, we get

$$\boldsymbol{v}_1 = v_n(z\mathbf{I}-\mathbf{X})^{-1}\boldsymbol{a}, \tag{3.3}$$

and substituting into the second we get

$$\boldsymbol{a}^T(z\mathbf{I}-\mathbf{X})^{-1}\boldsymbol{a} = z, \tag{3.4}$$

where $\mathbf{I}$ is the identity. Writing the matrix inverse in terms of its eigendecomposition $(z\mathbf{I}-\mathbf{X})^{-1} = \sum_i \boldsymbol{x}_i(z-\chi_i)^{-1}\boldsymbol{x}_i^T$, where $\boldsymbol{x}_i$ is the ith eigenvector of $\mathbf{X}$ and χ_i is the corresponding eigenvalue, Eq. (3.4) becomes

$$\frac{(\boldsymbol{a}^T\boldsymbol{x}_1)^2}{z-(c+1)} + \sum_{i=2}^{n-1} \frac{(\boldsymbol{a}^T\boldsymbol{x}_i)^2}{z-\chi_i} = z, \tag{3.5}$$

where we have explicitly separated the largest eigenvalue $\chi_1 = c+1$ and the remaining $n-2$ eigenvalues, which follow the semicircle law.

Although we don't know the values of the quantities $\boldsymbol{a}^T\boldsymbol{x}_i$ appearing in Eq. (3.5), the left-hand side as a function of z clearly has poles at each of the eigenvalues χ_i and a tail that grows as $1/z$ for large z. Moreover, for properly normalized $\boldsymbol{x}_1$ the numerator of the first term in the equation is $\mathrm{O}(1/n)$ and hence this term diverges significantly only when $z-(c+1)$ is also $\mathrm{O}(1/n)$, i.e., when z is very close to the leading eigenvalue $c+1$. Hence the qualitative form of the function must be as depicted in Fig. 3.1 and solutions to the full equation correspond to the points where this form crosses the diagonal line representing the right-hand side of the equation. These points are marked with dots in the figure.

As the geometry of the figure makes clear, the solutions for z, which are the eigenvalues of the full adjacency matrix of our model including the hub vertex, must fall in between the eigenvalues χ_i of the matrix $\mathbf{X}$, and hence satisfy an interlacing condition of the form $z_1 > \chi_1 > z_2 > \chi_2 > \ldots > \chi_{n-1} > z_n$, where we have numbered both sets of eigenvalues in order from largest to smallest. In the limit where the network becomes large and the eigenvalues $\chi_2 \ldots \chi_{n-1}$ form a continuous semicircular band, this interlacing imposes tight bounds on the solutions z_3 to z_{n-1}, such that they must follow the same semicircle distribution. Moreover, the leading eigenvalue z_1 has to fall within $\mathrm{O}(1/n)$ of $\chi_1 = c+1$, and hence $z_1 \to c+1$ in the large size limit.

This leaves just two unknown eigenvalues, z_2 lying above the semicircular band and z_n lying below it. In the context of the eigenvector centrality it is the one at the top that we care about. In Fig. 3.1 this eigenvalue is depicted as lying below the

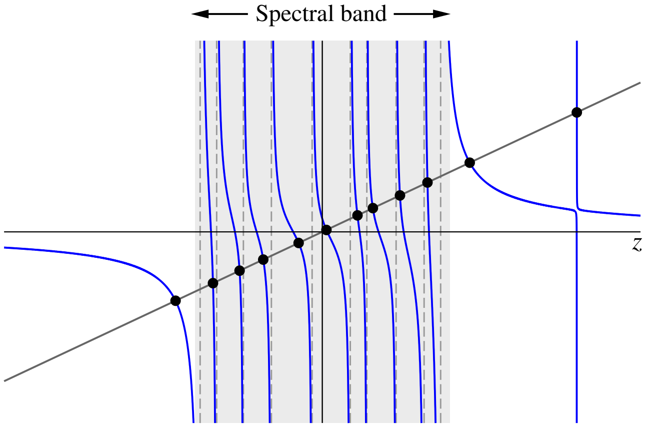


Figure 3.1: Graphical representation of the solution of Eq. (3.5). The curves represent the left-hand side of the equation, which has poles at the positions of the eigenvalues χ_i (marked by the vertical dashed lines). The diagonal line represents the right-hand side and the points where the two cross, marked by dots, are the solutions of the equation for z.

leading eigenvalue z_1, but it turns out that this is not always the case, as we now show.

Consider Eq. (3.5) for any value of z well away from $c+1$, so that the first term on the left can be neglected (meaning that z is not within $\mathrm{O}(1/n)$ of $c+1$). The vector $\boldsymbol{x}_i$ for $i \geq 2$ is uncorrelated with $\boldsymbol{a}$ and hence the product $\boldsymbol{a}^T\boldsymbol{x}_i$ is a Gaussian random variable with variance d/n and, averaging over the randomness, the equation then simplifies to

$$\frac{d}{n}\operatorname{Tr}(z\mathbf{I}-\mathbf{X})^{-1} = z. \tag{3.6}$$

The quantity $g(z) = n^{-1}\operatorname{Tr}(z\mathbf{I}-\mathbf{X})^{-1}$ is a standard one in the theory of random matrices—it is the so-called Stieltjes transform of $\mathbf{X}$, whose value for a symmetric matrix with iid elements such as this one is known to be [172]

$$g(z) = \frac{z-\sqrt{z^2-4c}}{2c}. \tag{3.7}$$

Combining Eqs. (3.6) and (3.7) and solving for z we find the eigenvalue we are looking for:

$$z_2 = \frac{d}{\sqrt{d-c}}. \tag{3.8}$$

Depending on the degree d of the hub, this eigenvalue may be either smaller or larger than the other high-lying eigenvalue $z_1 = c+1$. Writing $d/\sqrt{d-c} > c+1$ and rearranging, we see that the hub eigenvalue becomes the leading eigenvalue when

$$d > c(c+1), \tag{3.9}$$

i.e., when the hub degree is roughly the square of the mean degree. Below this point, the leading eigenvalue is the same as that of the random graph without the hub and the eigenvector centrality is given by the corresponding eigenvector, which is well behaved, so the centrality has no problems. Above this point, however, the leading eigenvector is the one introduced by the hub, and this eigenvector, as we now show, has severe problems.

If the eigenvector $\boldsymbol{v} = (\boldsymbol{v}_1 | v_n)$ is normalized to unity then Eq. (3.3) implies that

$$1 = |\boldsymbol{v}_1|^2 + v_n^2 = v_n^2 \big[\boldsymbol{a}^T (z\mathbf{I} - \mathbf{X})^{-2} \boldsymbol{a} + 1\big], \tag{3.10}$$

and hence

$$\begin{aligned} v_n^2 &= \frac{1}{\boldsymbol{a}^T (z\mathbf{I} - \mathbf{X})^{-2} \boldsymbol{a} + 1} = \frac{1}{(d/n)\,\mathrm{Tr}(z\mathbf{I} - \mathbf{X})^{-2} + 1} \\ &= \frac{1}{-dg'(z) + 1}, \end{aligned}$$

where $g(z)$ is again the Stieltjes transform, Eq. (3.7), and $g'(z)$ is its derivative. Performing the derivative and setting $z = d/\sqrt{d-c}$, we find that

$$v_n^2 = \frac{d-2c}{2d-2c}, \tag{3.11}$$

which is constant and does not vanish as $n \to \infty$. In other words, a finite fraction of the weight of the vector is concentrated on the hub vertex.

The neighbors of the hub also receive significant weight: the average of their values is given by

$$\frac{\boldsymbol{a}^T \boldsymbol{v}_1}{d} = \frac{v_n}{d} \boldsymbol{a}^T (z\mathbf{I} - \mathbf{X})^{-1} \boldsymbol{a} = v_n g(z) = \frac{v_n}{\sqrt{d-c}}. \tag{3.12}$$

Thus they are smaller than the hub centrality v_n, but still constant for large n. Finally, defining the $(n-1)$-element uniform vector $\mathbf{1} = (1, 1, 1, \ldots)$, the average of all $n-1$ non-hub vector elements is

$$\langle v_i \rangle = \frac{\mathbf{1}^T \boldsymbol{v}_1}{n-1} = \frac{v_n}{n-1} \mathbf{1}^T (z\mathbf{I} - \mathbf{X})^{-1} \boldsymbol{a}, \tag{3.13}$$

where we have used Eq. (3.3) again. Averaging over the randomness and noting that $\mathbf{X}$ and $\boldsymbol{a}$ are independent and that the average of $\boldsymbol{a}$ is $d\mathbf{1}/(n-1)$, we then get

$$\langle v_i \rangle = \frac{d v_n}{n-1} g(z) = \frac{1}{n-1} \frac{d v_n}{\sqrt{d-c}}, \tag{3.14}$$

which falls off as $1/n$ for large n.

Thus, in the regime above the transition defined by (3.9), where the eigenvector generated by adding the hub is the leading eigenvector, a non-vanishing fraction of the eigenvector centrality falls on the hub vertex and its neighbors, while the average vertex in the network gets only an $\mathrm{O}(1/n)$ vanishing centrality in the limit of large n, much less than the $\mathrm{O}(1/\sqrt{n})$ centrality received by the average vertex below the transition. This is the phenomenon we refer to as localization: the abrupt focusing of essentially all of the centrality on just a few vertices as the degree of the hub passes above the critical value $c(c+1)$. In the localized regime the eigenvector centrality picks out the hub and its neighbors clearly, but assigns vanishing weight to the average node. Thus, in this regime the dynamic range of centrality is dramatically reduced,

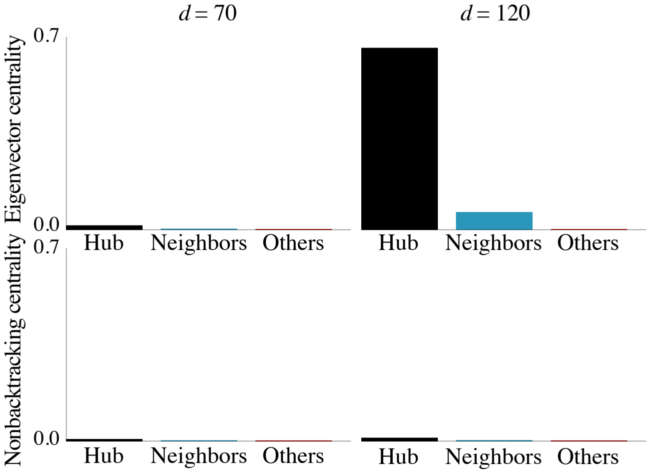


Figure 3.2: Bar charts of centralities for three categories of node for four examples of the model network studied here, as described in the text. All plots share the same scale. Error bars are small enough to be invisible on this scale.

reducing our ability to distinguish between nodes.

3.2.1 Numerical results

As a demonstration of the localization phenomenon, we show in Fig. 3.2 plots of the centralities of nodes in networks generated using our model. Each plot shows the average centrality of the hub, its neighbors, and all other nodes for a one-million-node network with $c = 10$. The top two plots show the situation for the standard eigenvector centrality for two different values of the hub degree—$d = 70$ and $d = 120$. The former lies well within the regime where there is no localization, while the latter is in the localized regime. The difference between the two is striking—in the first the hub and its neighbors get higher centrality, as they should, but only modestly so, while in the second the centrality of the hub vertex becomes so large as to dominate the figure.

The extent of the localization can be quantified by calculating an inverse participation ratio $S = \sum_{i=1}^{n} v_i^4$. In the regime below the transition where there is no localization and all elements v_i are $\mathrm{O}(1/\sqrt{n})$ we have $S = \mathrm{O}(1/n)$. But if one or more elements are O(1), then $S = \mathrm{O}(1)$ also. Hence if there is a localization transition in the network then, in the limit of large n, S will go from being zero to nonzero at the transition in the classic manner of an order parameter. Fig. 3.3 shows a set of such transitions in our model, each falling precisely at the expected position of the localization transition.

3.2.2 Power-law networks

So far we have looked only at the localization process in a simple model network, but localization occurs in more realistic networks as well. In general, we expect it to be a problem in networks with high-degree hubs or in very sparse networks, those with low average degree c, where it is relatively easy for the degree of a typical vertex to exceed the localization threshold. Many real-world networks fall into these categories. Consider, for example, the common case of a network with a power-law degree distribution, such that the fraction p_k of nodes with degree k goes as $k^{-\alpha}$ for some constant exponent α [14]. We can mimic such a network using the so-called configuration model [129, 144], a random graph with specified degree distribution. There are again two different ways a leading eigenvalue can be generated, one due to the average behavior of the entire network and one due to hub vertices of particularly high degree. In the first case the highest eigenvalue for the configuration model is known to be equal to the ratio of the second and first moments of the degree distribution $\langle k^2\rangle/\langle k\rangle$ in the limit of large network size and large average degree [38, 134]. At the same time, the leading eigenvalue must satisfy the Rayleigh bound $z \geq \boldsymbol{x}^T\mathbf{A}\boldsymbol{x}/\boldsymbol{x}^T\boldsymbol{x}$ for any real vector $\boldsymbol{x}$, with better bounds achieved when $\boldsymbol{x}$ better approximates the true leading eigenvector. If d denotes the highest degree of any hub in the network and we choose an approximate eigenvector of form similar to the one in our earlier model network, having elements $x_i = 1$ for the hub, $1/\sqrt{d}$ for neighbors of the hub, and zero otherwise, then the Rayleigh bound implies $z \geq \sqrt{d}$.

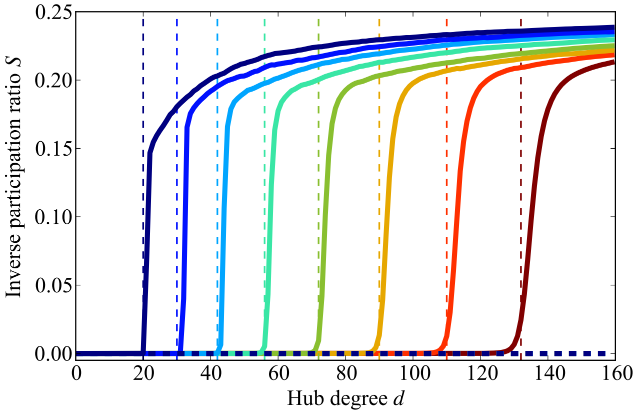


Figure 3.3: Numerical results for the inverse participation ratio S as a function of hub degree d for networks generated using the model described in the text with $n = 1\,000\,000$ vertices and average degree c ranging from 4 to 11. The solid curves are eigenvector centrality; the horizontal dashed curves are the nonbacktracking centrality. The vertical dashed lines are the expected positions of the localization transition for each curve, from Eq. (3.9).

Thus the eigenvector generated by the hub will be the leading eigenvector whenever $\sqrt{d} > \langle k^2 \rangle / \langle k \rangle$ (possibly sooner, but not later).

In a power-law network with n vertices and exponent α, the highest degree goes as $d \sim n^{1/(\alpha-1)}$ [55] and hence increases with increasing n, while $\langle k^2 \rangle \sim d^{3-\alpha}$ and $\langle k \rangle \sim$ constant for the common case of $\alpha < 3$. Thus we will have $\sqrt{d} > \langle k^2 \rangle / \langle k \rangle$ for large n provided $\frac{1}{2} > 3 - \alpha$. So we expect the hub eigenvector to dominate and the eigenvector centrality to fail due to localization when $\alpha > \frac{5}{2}$,[1] something that happens in many real-world networks. (Similar arguments have also been made by Chung *et al.* [38] and by Goltsev *et al.* [72].) We give empirical measurements of localization in a number of real-world networks in Table 3.1 below.

[1] Intuitively, as α increases, the distribution of degrees becomes more uneven. Though the average hub size decreases with increasing α, the average degree decreases even more quickly, and thus increasing α increases localization.

3.3 Nonbacktracking centrality

So if eigenvector centrality fails to do its job, what can we do to fix it? Qualitatively, the localization effect arises because a hub with high eigenvector centrality gives high centrality to its neighbors, which in turn reflect it back again and inflate the hub's centrality. We can make the centrality well behaved again by preventing this reflection. To achieve this we propose a modified eigenvector centrality, similar in many ways to the standard one, but with an important change. We define the centrality of node j to be the sum of the centralities of its neighbors as before, but the neighbor centralities are now calculated *in the absence of node j*. This is a natural definition in many ways—when I ask my neighbors what their centralities are in order to calculate my own, I want to know their centrality due to their other neighbors, not myself. This modified eigenvector centrality has the desirable property that when typical degrees are large, so that the exclusion or not of any one node makes little difference, its value will tend to that of the standard eigenvector centrality. But in sparser networks of the kind that can give problems, it will be different from the standard measure and, as we will see, better behaved.

Our centrality measure can be calculated using the Hashimoto or nonbacktracking matrix [83, 105], which is defined as follows. Starting with an undirected network with m edges, one first converts it to a directed one with $2m$ edges by replacing each undirected edge with two directed ones pointing in opposite directions. The nonbacktracking matrix $\mathbf{B}$ is then the $2m \times 2m$ non-symmetric matrix with one row and one column for each directed edge $i \to j$ and elements

$$B_{k\to l,i\to j} = \delta_{jk}(1 - \delta_{il}), \tag{3.15}$$

where δ_{ij} is the Kronecker delta. Thus a matrix element is equal to one if edge $i \to j$ points into the same vertex that edge $k \to l$ points out of and edges $i \to j$ and

$k \to l$ are not pointing in opposite directions between the same pair of vertices, and zero otherwise. Note that, since the nonbacktracking matrix is not symmetric, its eigenvalues are in general complex, but the largest eigenvalue is always real, as is the corresponding eigenvector.

The element $v_{i\to j}$ of the leading eigenvector of the nonbacktracking matrix now gives us the centrality of vertex i ignoring any contribution from j, and the full nonbacktracking centrality x_j of vertex j is defined to be the sum of these centralities over the neighbors of j:

$$x_j = \sum_i A_{ij} v_{i\to j}. \tag{3.16}$$

In principle one can calculate this centrality directly by calculating the leading eigenvector of $\mathbf{B}$ and then applying Eq. (3.16). In practice, however, one can perform the calculation faster by making use of the so-called Ihara (or Ihara–Bass) determinant formula, from which it can be shown [105] that the vector $\boldsymbol{x}$ of centralities is equal to the first n elements of the leading eigenvector of the $2n \times 2n$ matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{I} - \mathbf{D} \\ \mathbf{I} & \mathbf{0} \end{pmatrix}, \tag{3.17}$$

where $\mathbf{A}$ is the adjacency matrix as previously, $\mathbf{I}$ is the $n \times n$ identity matrix, and $\mathbf{D}$ is the diagonal matrix with the degrees of the vertices along the diagonal. Since $\mathbf{M}$ only has marginally more nonzero elements than the adjacency matrix itself ($2m + 2n$ for a network with m edges and n vertices, versus $2m$ for the adjacency matrix), finding its leading eigenvector takes only slightly longer than the calculation of the ordinary eigenvector centrality.

To see that the nonbacktracking centrality can indeed eliminate the localization transition, consider again our random-graph-plus-hub model and, as before, let us first consider the random graph on its own, without the hub. Our goal will be to calculate the leading eigenvalue of the nonbacktracking matrix for this random graph and then demonstrate that no other eigenvalue ever surpasses it even when the hub is added into the picture, and hence that there is no transition of the kind that occurs with the standard eigenvector centrality.

Since all elements of the nonbacktracking matrix are real and nonnegative, the leading eigenvalue and eigenvector satisfy the Perron–Frobenius theorem, meaning the eigenvalue is itself real and nonnegative as are all elements of the eigenvector for appropriate choice of normalization. Note moreover that at least one element of the eigenvector must be nonzero, so the average of the elements is strictly positive.

Making use of the definition of the nonbacktracking matrix in Eq. (3.15), the eigenvector equation $z\boldsymbol{v} = \mathbf{B}\boldsymbol{v}$ takes the form

$$\begin{aligned} zv_{k\to l} &= \sum_{i\to j} B_{k\to l,i\to j} v_{i\to j} = \sum_{i\to j} \delta_{jk}(1-\delta_{il}) v_{i\to j} \\ &= \sum_{ij} A_{ij}\delta_{jk}(1-\delta_{il}) v_{i\to j} = \sum_{i} A_{ik}(1-\delta_{il}) v_{i\to k} \end{aligned} \tag{3.18}$$

or

$$zv_{j\to l} = \sum_{i(\neq l)} A_{ij} v_{i\to j}, \tag{3.19}$$

where we have changed variables from k to j for future convenience. Expressed in words, this equation says that z times the centrality of an edge emerging from vertex j is equal to the sum of the centralities of the other edges feeding into j. For an uncorrelated, locally tree-like random graph of the kind we are considering here, i.e., a network where the source and target of a directed edge are chosen independently and there is a vanishing density of short loops, the centralities on the incoming edges are drawn at random from the distribution over all edges—the fact that they all point to vertex j has no influence on their values in the limit of large graph size. Bearing this in mind, let us calculate the average $\langle v\rangle$ of the centralities $v_{j\to l}$ over all edges in the network, which we do in two stages. First, making use of Eq. (3.19), we calculate the sum over all edges originating at vertices j whose degree k_j takes a particular value k:

$$\begin{aligned} z \sum_{\substack{j\to l:\\ k_j=k}} v_{j\to l} &= z \sum_{jl:k_j=k} A_{jl} v_{j\to l} = \sum_{jl:k_j=k} A_{jl} \sum_{i(\neq l)} A_{ij} v_{i\to j} \\ &= \sum_{ij:k_j=k} A_{ij} v_{i\to j} \sum_{l(\neq i)} A_{jl} = (k-1) \sum_{ij:k_j=k} A_{ij} v_{i\to j} \\ &= \langle v\rangle (k-1) \sum_{ij:k_j=k} A_{ij} = \langle v\rangle (k-1) k n_k, \end{aligned} \tag{3.20}$$

where n_k is the number of vertices with degree k and we have in the third line made use of the fact that $v_{i\to j}$ has the same distribution as values in the graph as whole to make the replacement $v_{i\to j} \to \langle v\rangle$ in the limit of large graph size.

Now we sum this expression over all values of k and divide by the total number of edges $2m$ to get the value of the average vector element $\langle v\rangle$:

$$z\langle v\rangle = \frac{\langle v\rangle}{2m} \sum_{k=0}^{\infty} (k-1) k n_k = \langle v\rangle \frac{\langle k^2\rangle - \langle k\rangle}{\langle k\rangle}. \tag{3.21}$$

Thus for any vector $\boldsymbol{v}$ we must either have $\langle v\rangle = 0$, which as we have said cannot happen for the leading eigenvector, or

$$z = \frac{\langle k^2\rangle - \langle k\rangle}{\langle k\rangle}. \tag{3.22}$$

For the particular case of the Poisson random graph under consideration here, this gives a leading eigenvalue of $z = c$, the average degree.

This result has been derived previously by other means [105] but the derivation given here has the advantage that it is easy to adapt to the case where we add a hub vertex to the network. Doing so adds just a single term to Eq. (3.21) thus:

$$z\langle v\rangle = \frac{\langle v\rangle}{2m}\left[\sum_{k=0}^{\infty}(k-1)kn_k + (d-1)d\right], \tag{3.23}$$

where d is the degree of the hub, as previously. Hence the leading eigenvalue is

$$z = \frac{(n-1)\big(\langle k^2\rangle - \langle k\rangle\big) + (d-1)d}{2m}. \tag{3.24}$$

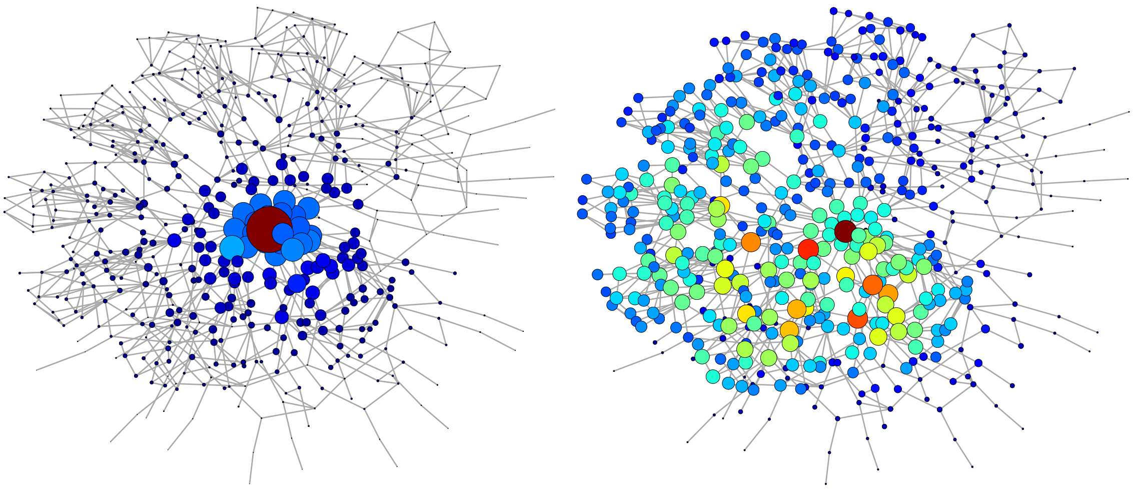
For constant d and constant (or growing) average degree, however, the term in d becomes negligible in the limit of large n and we recover the same result as before $z = c$.

Thus no new leading eigenvalue is introduced by the hub in the case of the non-backtracking matrix, and there is no phase transition as eigenvalues cross for any value of d.

It is worth noting, however, that there are other mechanisms by which high-lying eigenvalues can be generated. For instance, if a network contains a large clique (a complete subgraph in which every node is connected to every other) it can generate an outlying eigenvalue of arbitrary size, as we can see by making use of the so-called Collatz–Wielandt formula, a corollary of the Perron–Frobenius theorem that says that for any vector $\boldsymbol{v}$ the leading eigenvalue satisfies

$$z \geq \min_{i:v_i\neq 0}\frac{[\mathbf{B}\boldsymbol{v}]_i}{v_i}. \tag{3.25}$$

Choosing a $\boldsymbol{v}$ whose elements are one for edges within the clique and zero elsewhere, we find that a clique of size k implies $z \geq k-2$, which can supersede any other leading eigenvalue for sufficiently large k. The corresponding eigenvector is localized on the clique vertices, potentially causing trouble once again for the eigenvector centrality. This localization on cliques would be an interesting topic for further investigation.



(a) Eigenvector centrality (b) Nonbacktracking centrality

Figure 3.4: Eigenvector and nonbacktracking centralities for the electronic circuit network from Table 3.1. Node sizes are proportional to centrality (and color also varies with centrality).

3.3.1 Numerical results

As a test of our nonbacktracking centrality, we show in the lower two panels of Fig. 3.2 results for the same networks as the top two panels. As the figure makes clear, the measure now remains well behaved in the regime beyond the former position of the localization transition—there is no longer a large jump in the value of the centrality on the hub or its neighbors as we pass the transition. Similarly, the dashed curves in Fig. 3.3 show the inverse participation ratio for the nonbacktracking centrality and again all evidence of localization has vanished.

The inverse participation ratio also provides a convenient way to test for localization in other networks, both synthetic and real. Table 3.1 summarizes results for eleven networks, for both the traditional eigenvector centrality and the nonbacktracking version. The synthetic networks are generated using the random-graph-plus-hub model of this chapter and the configuration model with power-law degree distribution, and in each case there is evidence of localization in the eigenvector centrality in the regimes where it is expected and not otherwise, but no localization at all, in

				Non-
	Network	Nodes	Eigenvector	backtracking
Synthetic	Planted hub, $d = 70$	1 000 001	2.6×10^{-6}	1.4×10^{-6}
	Planted hub, $d = 120$	1 000 001	0.2567	1.4×10^{-6}
	Power law, $\alpha = 2.1$	1 000 000	0.0089	0.0040
	Power law, $\alpha = 2.9$	1 000 000	0.2548	0.0011
Empirical	Physics collaboration	12 008	0.0039	0.0039
	Word associations	13 356	0.0305	0.0075
	Youtube friendships	1 138 499	0.0479	0.0047
	Company ownership	7 253	0.2504	0.0161
	Ph.D. advising	1 882	0.2511	0.0386
	Electronic circuit	512	0.1792	0.0056
	Amazon	334 863	0.0510	0.0339

Table 3.1: Inverse participation ratio for a variety of networks calculated for traditional eigenvector centrality and the nonbacktracking version. The first four networks are computer-generated, as described in the text. The remainder are, in order: a network of coauthorships of papers in high-energy physics [111], word associations from the Free Online Dictionary of Computing [18], friendships between users of the Youtube online video service [127], a network of which companies own which others [145], academic advisors and advisees in computer science [47], electronic circuit 838 from the ISCAS 89 benchmark set [126], and a product co-purchasing network from the online retailer Amazon.com [111].

any case, for the nonbacktracking centrality. A similar picture is seen in the real-world networks—typically either localization in the eigenvector centrality but not the nonbacktracking version, or localization in neither case. Figure 3.4 illustrates the situation for one of the smaller real-world networks, where the values on the highest-degree vertex and its neighbors are overwhelmingly large for the eigenvector centrality (left panel) but not for the nonbacktracking centrality (right panel).

3.4 Conclusions

In this chapter I have shown that the widely used network measure known as eigenvector centrality fails under commonly occurring conditions because of a localization transition in which most of the weight of the centrality concentrates on a small number of vertices. The phenomenon is particularly visible in networks with high-degree hubs or power-law degree distributions, which includes many important real-world examples. I propose a new spectral centrality measure based on the nonbacktracking matrix which rectifies the problem, giving values similar to the standard eigenvector centrality in cases where the latter is well behaved, but avoiding localization in cases where the standard measure fails. The new measure is found to give significant decreases in localization on both synthetic and real-world networks. Moreover, the new measure can be calculated almost as quickly as the standard one, and hence is practical for the analysis of very large networks of the kind common in recent studies.

The nonbacktracking centrality is not the only possible solution to the problem of localization. For example, in studies of other forms of localization in networks it has been found effective to introduce a regularizing "teleportation" term into the adjacency and similar matrices, i.e., to add a small amount to every matrix element as if there were a weak edge between every pair of vertices [5, 155]. This strategy is reminiscent of Google's PageRank centrality measure [29], a popular variant of eigenvector centrality that includes such a teleportation term, and recent empirical studies suggest that PageRank may be relatively immune to localization [63]. The use of eigenvector centrality or PageRank depends on the network and importance process one is dealing with [26]. PageRank is more appropriate for directed graphs in which the importance contribution should be weighted inversely by a node's degree, while eigenvector centrality is more appropriate for importance processes with parallel duplication on undirected networks, such as the influence of attitudes. Eigenvector centrality is widely used, and thus those using it for network understanding will benefit from the analysis of localization given in this chapter. It would be a worthwhile topic for future research to develop theory similar to that presented here to describe localization (or lack of it) in PageRank and related measures. Ultimately, as I also discussed in Chapter II, the assumptions one makes about how centrality works can have a large impact on the outcome of one's centrality measure.

CHAPTER IV

Uncertain networks

4.1 Introduction

Most current techniques for the analysis of networks begin with the assumption that the network data available to us are reliable, a faithful representation of the true structure of the network. But many real-world data sets, perhaps most of them, in fact contain errors and inaccuracies. Thus, rather than representing a network by a set of nodes joined by binary yes-or-no edges, as is commonly done, one would ideally express this uncertainty by specifying a full distribution over possible networks. This distribution would be unwieldy, but with the simplifying assumption of edgewise-independent errors we need to specify a calibrated probability of connection only between each pair of nodes, which represents our certainty (or uncertainty) about the existence of the corresponding edge. If most of the probabilities are close to zero or one then the data are reliable—for every node pair we are close to being certain that it either is or is not connected by an edge. But if a significant fraction of pairs have a probability that is neither close to zero nor close to one then we are uncertain about the network structure. These probabilities could come from first-principles knowledge of the error process, or from calibration performed according to some ground truth [152]. In recent years an increasing number of network studies have started to provide probabilistic estimates of uncertainty in this way, particularly in the biological sciences.

One simple method for dealing with uncertain networks is *thresholding*: we assume that edges exist whenever their probability exceeds a certain threshold that we choose. In work on protein-protein interaction networks, for example, Krogan *et al.* [104] assembled a sophisticated interaction data set that includes explicit estimates of the likelihood of interaction between every pair of proteins studied. To analyze their data set, however, they then converted it into a conventional binary network by thresholding the likelihoods, followed by traditional network analyses. While this technique can certainly reveal useful information, it has some drawbacks. First, there is the issue of the choice of the threshold level. Krogan *et al.* used a value of 0.273 for their threshold, but there is little doubt that their results would be different if they had chosen a different value and little known about how to choose the value correctly. Second, thresholding throws away potentially useful information. There is a substantial difference between an edge with probability 0.3 and an edge with probability 0.9, but the distinction is lost if one applies a threshold at 0.273—both fall above the threshold and so are considered to be edges. Third, thresholded probability values fail to conserve basic network properties such as the expected number of edges, meaning that thresholded networks are essentially guaranteed to be wrong, often by a wide margin. If, for instance, we have 100 node pairs connected with probability 0.5 each, then on average we expect 50 of those pairs to be connected by edges in the true network. If we place a threshold on the probability values at, say, 0.273, however, then all 100 of them will be converted into edges, a result sufficiently far from the expected value of 50 as to have a very low chance of being correct.

In this chapter we develop an alternative and principled approach to the analysis of uncertain network data. We focus in particular on the problem of community detection in networks, one of the best studied analysis tasks. We make use of maximum-likelihood inference techniques, whose application to networks with definite edges is well developed [146, 39, 70, 50]. Here we extend those developments to uncertain networks and show that the resulting analyses give significantly better results in controlled tests than thresholding methods. As a corollary, our methods also allow us to estimate which of the uncertain edges in a data set is mostly likely to be a true edge and hence reconstruct, in a probabilistic fashion, the true structure of the underlying network.

A number of authors have considered related questions in the past. There exists a substantial literature on the analysis of weighted networks, meaning networks in which the positions of the edges are exactly known but the edges carry varying weights, such as strengths, lengths, or volumes of traffic. Such weighted networks are somewhat similar to the uncertain networks studied in this chapter—edges can be either strong or weak in a certain sense—but have importantly different semantics of weight generation. For instance, the data sets we consider include probabilities of connection for every node pair, whereas weighted networks have weights only for node pairs that are known to be connected by an edge. More importantly, in our uncertain networks we imagine that there is a definite underlying network but that it is not observed; all we see are noisy measurements of the underlying truth. In weighted networks the data are considered to be exact and true and the variation of edge weights represents an actual physical variation in the properties of connections.

Methods for analyzing weighted networks include simple mappings to unweighted networks and generalizations of standard methods to the weighted case [138]. Inference methods akin to those we use here have also been applied to the weighted case [1] and to the case of affinity matrices, as used for example in computer vision for image segmentation [161]. A little further afield, Harris and Srinivasan [82] study a noisy model of network failures in which edges are deleted with uniform probability, while Saade *et al.* [165] use spectral techniques to detect node properties, but not community affiliations, when the underlying network is known but the node properties depend on noisy edge labels. Guimer and Sales-Pardo [79] similarly give a framework for network inference in the presence of noise, but their model assumes one can observe only an unweighted network with possibly erroneous edges. Using inference techniques similar to those in this chapter, Xu *et al.* [187] have studied the prediction of edge labels and Kurihara *et al.* [107] have applied inference to a case where the data give the frequency of interaction between nodes. Lastly, Bassett *et al.* [17] have studied correlation matrices, which can be view as a type of weighted network, and give a technique for computing the probability that correlations are the result of chance, though this type of data is quite distinct from the edge probabilities studied in this chapter.

Several intellectual contributions distinguish our work from the existing literature. Our primary contribution is the model itself, which gives a new framework for analyzing uncertain network data using any number of models, not just those with community structure. Our approach might in future also lead to new techniques for adapting existing weighted methods to the analysis of uncertain networks in a rigorous manner, potentially leading to new belief propagation or spectral methods [51]. While our maximum likelihood estimation method is the optimal way of recovering model parameters, tractability demands relaxed methods which may be suboptimal. As we discuss below, we use several heuristics for finding the maximum likelihood, and these heuristics only guarantee convergence to a global optimum in the limit of large sparse graphs, with sufficient random restarts. For any given network instance, it is quite possible that a previously studied algorithm for weighted networks may give an approximately optimal solution, but we leave a broad empirical comparison to further work.

4.2 Methods

We focus on the problem of community detection in networks whose structure is uncertain. We suppose that we have data which, rather than specifying with certainty whether there is an edge between two nodes i and j, gives us only a probability Q_{ij} that there is an edge. We assume that the probabilities are independent, though correlated probabilities are certainly possible.

At the most basic level our goal is to classify the nodes of the network into non-overlapping communities with assortative structure. More generally we may also be interested in disassortative structures or mixed structures in which different groups may be either assortative or disassortative within the same network. Conceptually, we assume that even though our knowledge of the network is uncertain, there is a definite underlying network in which each edge either exists or does not, but we cannot see this network. The underlying network is assumed to be undirected and simple (i.e., it has no multi-edges or self-edges). The edge probabilities we observe are a noisy representation of the true network, but they nonetheless can contain information about structure—enough information, as we will see, to make possible the accurate detection of communities in many situations.

Our approach to the detection problem takes the classic form of a statistical inference algorithm. We propose a generative model for uncertain community-structured networks, then fit that model to our observed data. The parameters of the fit tell us about the community structure.

4.2.1 The model

The model we use is an extension to the case of uncertain networks of the standard stochastic block model, described in Section 1.2.8.

Given the parameters γ_r and ω_{rs}, one can write down the probability that we generate a particular network in which node i is assigned to group g_i and the placement

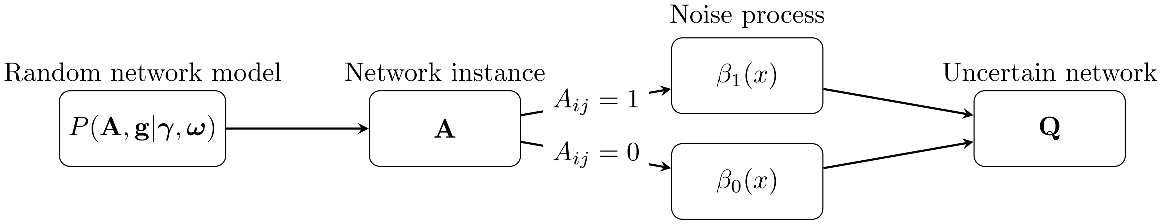


Figure 4.1: The model of uncertain network generation used in our calculations. A community assignment $\boldsymbol{g}$ and network $\mathbf{A}$ are drawn from a random network model such as the stochastic block model. The experimental uncertainty is represented by giving each pair of nodes i, j a probability Q_{ij} of being connected by an edge, drawn from different distributions for edges $A_{ij} = 1$ and nonedges $A_{ij} = 0$.

of the edges is described by an adjacency matrix $\mathbf{A}$ with elements $A_{ij} = 1$ if there is an edge between nodes i and j and 0 otherwise:

$$\begin{aligned} P(\mathbf{A}, \boldsymbol{g}|\boldsymbol{\gamma}, \boldsymbol{\omega}) &= P(\boldsymbol{g}|\boldsymbol{\gamma}) P(\mathbf{A}|\boldsymbol{g}, \boldsymbol{\omega}) \\ &= \prod_i \gamma_{g_i} \prod_{i<j} \omega_{g_i g_j}^{A_{ij}} (1 - \omega_{g_i g_j})^{1-A_{ij}}. \end{aligned} \tag{4.1}$$

Here $\boldsymbol{\gamma}$ represents the vector of group probabilities γ_r and $\boldsymbol{\omega}$ represents the matrix of probabilities ω_{rs}.

In extending the stochastic block model to uncertain networks we imagine a multistep process, illustrated in Figs. 4.1 and 4.2, in which the network is first generated using the standard stochastic block model and then the definite edges and nonedges are replaced by probabilities, reflecting uncertainty or noise in the network data. The exact shape of the noise depends on the detailed effects of the experimental procedure used to measure the network, which we assume to be unknown. We assume only that the edge likelihoods are calibrated [152] probabilities in a sense defined below in Eq. (4.4). However, it still turns out to be possible to perform precise inference on the data.

We represent the noise process by two unknown functions. The function $\beta_1(Q)$ represents the probability density on the interval from 0 to 1 that a true edge between two

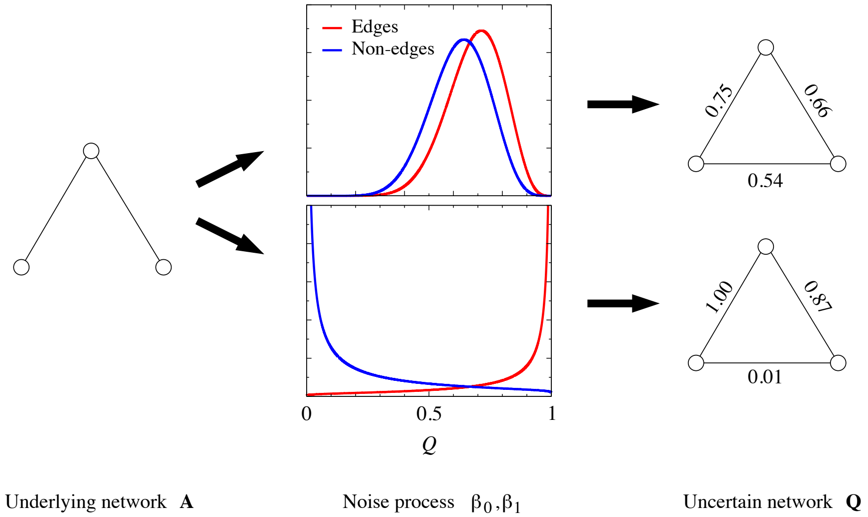


Figure 4.2: Simple example of the generation of two uncertain networks from an initial network with three nodes. The two networks generated (right-hand side) differ only in their noise distributions, $\beta_0(Q)$ and $\beta_1(Q)$, whose probability density functions (PDFs) are shown in the center. The lower pair of distributions corresponds to a low-noise setting in which the PDFs for edges and nonedges are quite distinct and the resulting probability matrix $\mathbf{Q}$ retains most of the information from the original adjacency matrix $\mathbf{A}$. The upper pair of distributions corresponds to a high-noise setting in which the two PDFs are almost the same and the final matrix $\mathbf{Q}$ retains little of the original network structure.

nodes in the original (unobserved) network gives rise to a measured probability Q of connection between the same nodes in the observed (probabilistic) data. Conversely, the function $\beta_0(Q)$ represents the probability density that a nonedge gives rise to probability Q.

Given these two functions and our edge-wise independence assumption, we can write an expression for the probability (technically, probability density) that a true network represented by adjacency matrix $\mathbf{A}$ gives rise to a matrix of observed edge probabilities $\mathbf{Q} = \{Q_{ij}\}$ thus:

$$P(\mathbf{Q}|\mathbf{A}) = \prod_{i<j} \bigl[\beta_1(Q_{ij})\bigr]^{A_{ij}} \bigl[\beta_0(Q_{ij})\bigr]^{1-A_{ij}}. \tag{4.2}$$

The crucial observation that makes our calculations possible is that the functions β_0 and β_1 are not independent, because the numbers Q_{ij} that they generate are not just any edge weights but are specifically probabilities and are assumed to be *calibrated* [152]. The calibration assumption requires that, in expectation over all node pairs, a pair with edge probability $Q_{ij} = Q$ must be connected by an edge ($A_{ij} = 1$) with probability Q. For example, 90% of all node pairs with $Q_{ij} = 0.9$ should, in expectation, be connected by edges.

If there are m edges in total in our true underlying network, then in expectation there are $m\beta_1(Q)$ edges and $[\binom{n}{2} - m]\beta_0(Q)$ nonedges with observed edge probability Q. Hence for every possible value of Q we must have

$$\frac{m\beta_1(Q)}{m\beta_1(Q) + (\binom{n}{2} - m)\beta_0(Q)} = Q. \tag{4.3}$$

Rearranging, we then find that

$$\frac{\beta_1(Q)}{\beta_0(Q)} = \frac{Q/\rho}{(1-Q)/(1-\rho)}, \tag{4.4}$$

where

$$\rho = \frac{m}{\binom{n}{2}} \tag{4.5}$$

is the so-called *density* of the network, the fraction of possible edges that are in fact present.[1] Since we don't know the true network, we don't normally know the value of m, but it can be approximated by the expected number of edges $\sum_{i<j} Q_{ij}$, which becomes an increasingly good estimate as the network gets larger, and from this figure we can calculate ρ.[2]

Note that Eq. (4.4) implies that $\beta_0(1) = 0$ and $\beta_1(0) = 0$, and that fixing any two of β_0, β_1, or ρ determines the third. The equation is also compatible with the choice $\beta_0(Q) = \delta(Q)$, $\beta_1(Q) = \delta(Q-1)$, where $\delta(x)$ is the Dirac delta function, which corresponds to the conventional case of a perfectly certain network with $Q_{ij} = A_{ij}$.

Using Eq. (4.4) we can now write Eq. (4.2) as

$$\begin{aligned} P(\mathbf{Q}|\mathbf{A}) = \prod_{i<j} &\frac{1-\rho}{1-Q_{ij}} \beta_0(Q_{ij}) \\ &\times \prod_{i<j} \left(\frac{Q_{ij}}{\rho}\right)^{A_{ij}} \left(\frac{1-Q_{ij}}{1-\rho}\right)^{1-A_{ij}}. \end{aligned} \tag{4.6}$$

Thus the noise function factors from our maximum-likelihood expression. The first product is a constant with respect to $\mathbf{A}$ and hence will have no effect on our likelihood maximization, because we care only about the position of the likelihood maximum and not its value. Henceforth, we will neglect this factor. Then we combine Eqs. (4.1) and (4.6) to get an expression for the likelihood of the data $\mathbf{Q}$ and the community

[1] For randomized models without a fixed density, ρ is the expected value of the density.

[2] A fully sound algorithm would require a complete MLE estimate of ρ. Here we simply estimate by edge counting.

assignments $\boldsymbol{g}$, neglecting constants and given the model parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$:

$$\begin{aligned}
P(\mathbf{Q},\boldsymbol{g}|\boldsymbol{\gamma},\boldsymbol{\omega}) &= \sum_{\mathbf{A}} P(\mathbf{Q}|\mathbf{A})P(\mathbf{A},\boldsymbol{g}|\boldsymbol{\gamma},\boldsymbol{\omega}) \\
&\propto \prod_i \gamma_{g_i} \prod_{i<j} \sum_{A_{ij}=0,1} \left[\frac{Q_{ij}\omega_{g_ig_j}}{\rho}\right]^{A_{ij}} \left[\frac{(1-Q_{ij})(1-\omega_{g_ig_j})}{1-\rho}\right]^{1-A_{ij}} \\
&= \prod_i \gamma_{g_i} \prod_{i<j} \biggl[\frac{Q_{ij}\omega_{g_ig_j}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{g_ig_j})}{1-\rho}\biggr]. \qquad (4.7)
\end{aligned}$$

Our goal is now, given a particular set of observed data $\mathbf{Q}$, to maximize this likelihood to find the best-fit parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$. In the process we will determine the community assignments $\boldsymbol{g}$ as well (which are frequently the primary objects of interest).

4.2.2 Fitting to empirical data

Fitting the model to an observed but uncertain network, represented by the probabilities Q_{ij}, means determining the values of the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$ that maximize the probability of generating the particular data we see. In other words, we want to maximize the *marginal likelihood* of the data given the parameters:

$$P(\mathbf{Q}|\boldsymbol{\gamma},\boldsymbol{\omega}) = \sum_{\boldsymbol{g}} P(\mathbf{Q},\boldsymbol{g}|\boldsymbol{\gamma},\boldsymbol{\omega}). \qquad (4.8)$$

Equivalently, we can maximize the logarithm of this quantity, which gives the same result (since the logarithm is a monotone function) but is often easier to maximize.

Direct maximization by differentiation gives rise to a set of implicit equations that have no simple solution, so instead we simplify with Jensen's inequality, Eq. (1.5).

Applying Jensen's inequality to (4.8), we get

$$
\begin{aligned}
\log P(\mathbf{Q}|\boldsymbol{\gamma},\boldsymbol{\omega}) &\geq \sum_{\boldsymbol{g}} q(\boldsymbol{g}) \log \frac{P(\mathbf{Q},\boldsymbol{g}|\boldsymbol{\gamma},\boldsymbol{\omega})}{q(\boldsymbol{g})} \\
&= \sum_{\boldsymbol{g}} q(\boldsymbol{g}) \sum_i \log \gamma_{g_i} + \tfrac{1}{2} \sum_{\boldsymbol{g}} q(\boldsymbol{g}) \sum_{ij} \log \left[\frac{Q_{ij}\omega_{g_i g_j}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{g_i g_j})}{1-\rho} \right] \\
&\quad - \sum_{\boldsymbol{g}} q(\boldsymbol{g}) \log q(\boldsymbol{g}) \\
&= \sum_i \sum_r q^i_r \log \gamma_r + \tfrac{1}{2} \sum_{ij} \sum_{rs} q^{ij}_{rs} \log \left[\frac{Q_{ij}\omega_{rs}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{rs})}{1-\rho} \right] \\
&\quad - \sum_{\boldsymbol{g}} q(\boldsymbol{g}) \log q(\boldsymbol{g}), \qquad (4.9)
\end{aligned}
$$

where q^i_r is the marginal probability within the probability distribution $q(\boldsymbol{g})$ that node i belongs to community r:

$$
q^i_r = \sum_{\boldsymbol{g}} q(\boldsymbol{g}) \delta_{g_i,r}, \qquad (4.10)
$$

and q^{ij}_{rs} is the joint marginal probability that nodes i and j belong to communities r and s respectively:

$$
q^{ij}_{rs} = \sum_{\boldsymbol{g}} q(\boldsymbol{g}) \delta_{g_i,r} \delta_{g_j,s}, \qquad (4.11)
$$

with δ_{ij} being the Kronecker delta, $\delta_{ij} = 1 \iff i = j$.

Following Eq. (1.6), the exact equality in (4.9), and hence the maximum of the right-hand side, is achieved when

$$
\begin{aligned}
q(\boldsymbol{g}) &= \frac{P(\mathbf{Q},\boldsymbol{g}|\boldsymbol{\gamma},\boldsymbol{\omega})}{\sum_{\boldsymbol{g}} P(\mathbf{Q},\boldsymbol{g}|\boldsymbol{\gamma},\boldsymbol{\omega})} \\
&= \frac{\prod_i \gamma_{g_i} \prod_{i<j} \left[\frac{Q_{ij}\omega_{g_i g_j}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{g_i g_j})}{1-\rho} \right]}{\sum_{\boldsymbol{g}} \prod_i \gamma_{g_i} \prod_{i<j} \left[\frac{Q_{ij}\omega_{g_i g_j}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{g_i g_j})}{1-\rho} \right]}. \qquad (4.12)
\end{aligned}
$$

Thus, calculating the maximum of the left-hand side of (4.9) with respect to the parameters $\boldsymbol{\gamma}, \boldsymbol{\omega}$ is equivalent to a double maximization of the right-hand side with respect to $q(\boldsymbol{g})$ (by choosing the value above) so as to make the two sides equal, and then with respect to the parameters. At first sight, this seems to make the problem more complex, but numerically it is in fact easier—the double maximization can be achieved in a relatively straightforward manner by alternately maximizing with respect to $q(\boldsymbol{g})$ using Eq. (4.12) and then with respect to the parameters. Such alternate maximizations can trivially be shown always to converge to a local maximum of the log-likelihood. They are not guaranteed to find the global maximum, however, so commonly we repeat the entire calculation several times from different starting points and choose among the results the one which gives the highest value of the likelihood.

Once we have converged to the maximum, the final value of the probability distribution $q(\boldsymbol{g})$ is given by Eq. (4.12) to be

$$q(\boldsymbol{g}) = \frac{P(\mathbf{Q}, \boldsymbol{g}|\gamma, \boldsymbol{\omega})}{P(\mathbf{Q}|\gamma, \boldsymbol{\omega})} = P(\boldsymbol{g}|\mathbf{Q}, \gamma, \boldsymbol{\omega}). \tag{4.13}$$

In other words, $q(\boldsymbol{g})$ is the posterior distribution over community assignments $\boldsymbol{g}$ given the observed data $\mathbf{Q}$ and the model parameters. Thus, in addition to telling us the values of the parameters, our calculation, which we discuss in more detail in Section 4.2.3, tells us the probability of any assignment of nodes to communities. Specifically, the one-node marginal probability q_r^i, Eq. (4.10), tells us the probability that node i belongs to community r and, armed with this information, we can calculate the most probable community that each node belongs to, which is the primary goal of our calculation. These marginals also allow us to assess the strength of our community structure, as when the data poorly support community structure the posterior distribution simply becomes uniform.

We still need to perform the maximization of (4.9) over the parameters. We note first that the final sum is independent of either $\boldsymbol{\gamma}$ or $\boldsymbol{\omega}$ and hence can be neglected. Maximization of the remaining terms with respect to $\boldsymbol{\gamma}$ is straightforward. Differentiating with respect to γ_r, subject to the normalization condition $\sum_r \gamma_r = 1$, gives

$$\gamma_r = \frac{1}{n} \sum_i q_r^i. \tag{4.14}$$

Maximization with respect to $\boldsymbol{\omega}$ is a little more tricky. Only the second term in (4.9) depends on $\boldsymbol{\omega}$, but direct differentiation of this term yields a difficult equation, so instead we apply Jensen's inequality (1.5) again, giving

$$\begin{aligned} &\sum_{ij} \sum_{rs} q_{rs}^{ij} \log\biggl[\frac{Q_{ij}\omega_{rs}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{rs})}{1-\rho}\biggr] \\ &\geq \sum_{ij} \sum_{rs} q_{rs}^{ij} \biggl[t_{rs}^{ij} \log \frac{Q_{ij}\omega_{rs}}{\rho t_{rs}^{ij}} + (1-t_{rs}^{ij}) \log \frac{(1-Q_{ij})(1-\omega_{rs})}{(1-\rho)(1-t_{rs}^{ij})}\biggr], \end{aligned} \tag{4.15}$$

where t_{rs}^{ij} is any number between zero and one. The exact equality, and hence the maximum of the right-hand side, is achieved when

$$t_{rs}^{ij} = \frac{Q_{ij}\omega_{rs}/\rho}{Q_{ij}\omega_{rs}/\rho + (1-Q_{ij})(1-\omega_{rs})/(1-\rho)}. \tag{4.16}$$

Thus, by the same argument as previously, we can maximize the left-hand side of (4.15) by repeatedly maximizing the right-hand side with respect to t_{rs}^{ij} using Eq. (4.16) and with respect to ω_{rs} by differentiation. Performing the derivative and setting the result to zero, we find that the maximum with respect to ω_{rs} falls at

$$\omega_{rs} = \frac{\sum_{ij} q_{rs}^{ij} t_{rs}^{ij}}{\sum_{ij} q_{rs}^{ij}}. \tag{4.17}$$

The optimal values of the ω_{rs} can now be calculated by iterating Eqs. (4.16) and (4.17) alternately to convergence from a suitable initial condition.

The quantity t_{rs}^{ij} has a simple physical interpretation, as we can see by applying

Eq. (4.4) to (4.16), giving

$$t_{rs}^{ij} = \frac{\omega_{rs}\beta_1(Q_{ij})}{\omega_{rs}\beta_1(Q_{ij}) + (1-\omega_{rs})\beta_0(Q_{ij})}. \tag{4.18}$$

But by definition

$$\omega_{rs} = P(A_{ij} = 1 | g_i = r, g_j = s), \tag{4.19}$$

$$\beta_1(Q_{ij}) = P(Q_{ij} | A_{ij} = 1), \tag{4.20}$$

$$\beta_0(Q_{ij}) = P(Q_{ij} | A_{ij} = 0), \tag{4.21}$$

and hence

$$\begin{aligned} t_{rs}^{ij} &= \frac{P(A_{ij} = 1 | g_i = r, g_j = s) P(Q_{ij} | A_{ij} = 1)}{P(Q_{ij} | g_i = r, g_j = s)} \\ &= P(A_{ij} = 1 | Q_{ij}, g_i = r, g_j = s). \end{aligned} \tag{4.22}$$

In other words, t_{rs}^{ij} is the posterior probability that there is an edge between nodes i and j, given that they are in groups r and s respectively. This quantity will be useful shortly when we consider the problem of reconstructing a network from uncertain observations.

We now have a complete algorithm for fitting our model to the observed data. The steps of the algorithm are as follows:

1. Make an initial guess (for instance at random) for the values of the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$.

2. Calculate the distribution $q(\boldsymbol{g})$ from Eq. (4.12).

3. Calculate the one- and two-node marginal probabilities q_r^i and q_{rs}^{ij} from Eqs. (4.10) and (4.11).

4. From these quantities calculate updated values of $\boldsymbol{\gamma}$ from Eq. (4.14) and $\boldsymbol{\omega}$ by iterating Eqs. (4.16) and (4.17) to convergence starting from the current estimate of $\boldsymbol{\omega}$.

5. Repeat from step 2 until $q(\boldsymbol{g})$ and the model parameters converge.

The end result is a maximum likelihood estimate of the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$ along with the posterior distribution over community assignments $q(\boldsymbol{g})$ and the probability t_{rs}^{ij} of an edge between any pair of nodes.

Equation (4.17) can usefully be simplified a little further, in two ways. First, note that Eq. (4.16) implies that $t_{rs}^{ij} = 0$ whenever $Q_{ij} = 0$. All of the real-world data sets we have examined are *sparse*, meaning that a large majority of the probabilities Q_{ij} are zero. This means that most of the terms in the numerator of (4.17) vanish and can be dropped from the sum, which speeds up the calculation considerably. Indeed t_{rs}^{ij} need not be evaluated at all for node pairs i, j such that $Q_{ij} = 0$, since this sum is the only place that t_{rs}^{ij} appears in our calculation. Moreover it turns out that we need not evaluate q_{rs}^{ij} for such node pairs either. The only other place that q_{rs}^{ij} appears is in the denominator of Eq. (4.17), which can be simplified by using Eq. (4.11) to rewrite it thus:

$$\sum_{ij} q_{rs}^{ij} = \sum_{g} q(\boldsymbol{g}) \sum_{i} \delta_{g_i,r} \sum_{j} \delta_{g_j,s} = \langle n_r n_s \rangle, \tag{4.23}$$

where $\langle \ldots \rangle$ indicates an average over $q(\boldsymbol{g})$ and $n_r = \sum_i \delta_{g_i,r}$ is the number of nodes in group r, for community assignment $\boldsymbol{g}$. For large networks the number of nodes in a group becomes tightly peaked about its mean value so that $\langle n_r n_s \rangle \simeq \langle n_r \rangle \langle n_s \rangle$ where $\langle n_r \rangle = \sum_{\boldsymbol{g}} q(\boldsymbol{g}) \sum_i \delta_{g_i,r} = \sum_i q_r^i$. Hence

$$\omega_{rs} = \frac{\sum_{ij} q_{rs}^{ij} t_{rs}^{ij}}{\sum_i q_r^i \sum_j q_s^j}. \tag{4.24}$$

This obviates the need to calculate q_{rs}^{ij} for node pairs such that $Q_{ij} = 0$ (which is most node pairs), and in addition speeds the calculation further because the denominator can now be evaluated in time proportional to the number of nodes in the network, rather than the number of nodes squared, as in Eq. (4.17). (And the numerator can be evaluated in time proportional to the number of nonzero Q_{ij}, which is small.)

4.2.3 Belief propagation

In principle, the methods of the previous section constitute a complete algorithm for fitting our model to observed network data. In practice, however, it is an impractical one because it's unreasonably slow. The bottleneck is the sum in the denominator of Eq. (4.12), which is a sum over all possible assignments $\boldsymbol{g}$ of nodes to communities. If there are n nodes and k communities then there are k^n possible assignments, a number that grows with n so rapidly as to prohibit explicit numerical evaluation of the sum for all but the smallest of networks.

This is not a new problem, it is common to most EM algorithms. The traditional way around it is to approximate the distribution $q(\boldsymbol{g})$ by importance sampling using Markov chain Monte Carlo. In this chapter, however, we use a different method, proposed recently by Decelle *et al.* [50, 51] and specific to networks, namely belief propagation.

Originally developed in physics and computer science for the probabilistic solution of problems on graphs and lattices [151, 125], belief propagation is a message passing method in which the nodes of a network exchange messages or *beliefs*, which are probabilities representing the current best estimate of the solution to the problem of interest. In the present case we define a message $\eta_r^{i\to j}$ expressing the probability that node i belongs to community r if node j is removed from the network. The removal of a node is crucial, since it allows us to write a self-consistent set of equations satisfied by the messages, whose solution gives us the distribution $q(\boldsymbol{g})$ over group assignments. Although the equations can without difficulty be written exactly and in full, we will here approximate them to leading order only in the small quantities ω_{rs}. We find this approximation to give excellent results in our applications and to give considerably simpler equations, as well as giving a faster final algorithm.

Within this approximation, the belief propagation equation for the message $\eta_r^{i\to j}$ is:

$$\eta_r^{i\to j} = \frac{\gamma_r}{Z_{i\to j}} \exp\biggl(-\sum_{k,s} q_s^k \omega_{rs}\biggr) \prod_{\substack{k(\neq j)\\ Q_{ik}\neq 0}} \sum_s \eta_s^{k\to i} \biggl[\frac{Q_{ik}\omega_{rs}}{\rho} + \frac{(1-Q_{ik})(1-\omega_{rs})}{1-\rho}\biggr], \quad (4.25)$$

where $Z_{i\to j}$ is a normalization coefficient that ensures $\sum_r \eta_r^{i\to j} = 1$, having value

$$Z_{i\to j} = \sum_r \gamma_r \exp\biggl(-\sum_{k,s} q_s^k \omega_{rs}\biggr) \prod_{\substack{k(\neq j)\\ Q_{ik}\neq 0}} \sum_s \eta_s^{k\to i} \biggl[\frac{Q_{ik}\omega_{rs}}{\rho} + \frac{(1-Q_{ik})(1-\omega_{rs})}{1-\rho}\biggr]. \quad (4.26)$$

q_r^i is, as before, the one-node marginal probability of Eq. (4.10), which can itself be conveniently calculated directly from the messages $\eta_r^{i\to j}$ via

$$q_r^i = \frac{\gamma_r}{Z_i} \exp\biggl(-\sum_{j,s} q_s^j \omega_{rs}\biggr) \prod_{\substack{j\\ Q_{ij}\neq 0}} \sum_s \eta_s^{j\to i} \biggl[\frac{Q_{ij}\omega_{rs}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{rs})}{1-\rho}\biggr]. \quad (4.27)$$

The normalization coefficient Z_i is given by

$$Z_i = \sum_r \gamma_r \exp\biggl(-\sum_{j,s} q_s^j \omega_{rs}\biggr) \prod_{\substack{j\\ Q_{ij}\neq 0}} \sum_s \eta_s^{j\to i} \biggl[\frac{Q_{ij}\omega_{rs}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{rs})}{1-\rho}\biggr]. \quad (4.28)$$

These equations are exact if the set of node pairs i,j with edge probabilities $Q_{ij} > 0$ forms a tree or is at least locally tree-like (meaning that arbitrarily large local neighborhoods take the form of trees in the limit of large network size). For non-trees, which includes most real-world networks, they are only approximate, but previous results from a number of studies show the approximation to be a good one in practice [125, 50, 51, 188, 192, 124]. Probability data of the kind we consider might further deviate from a strict tree-like form if they include a large number of low-probability edges, but nonetheless we find the belief propagation method to work well.

Solution of the equations is by iteration. Typically we start from the current best estimate of the values of the beliefs and iterate to convergence, then from the converged values we calculate the crucial two-node marginal probability q_{rs}^{ij} by noting that

$$\begin{aligned} q_{rs}^{ij} &= P(g_i = r, g_j = s | Q_{ij}) \\ &= \frac{P(g_i = r, g_j = s) P(Q_{ij} | g_i = r, g_j = s)}{\sum_{rs} P(g_i = r, g_j = s) P(Q_{ij} | g_i = r, g_j = s)}. \end{aligned} \tag{4.29}$$

where all data $\mathbf{Q}$ other than Q_{ij} are assumed given in each probability. The probabilities in these expressions are equal to

$$P(g_i = r, g_j = s) = \eta_r^{i\to j} \eta_s^{j\to i}, \tag{4.30}$$

$$\begin{aligned} P(Q_{ij} | g_i = r, g_j = s) &= \beta_0(Q_{ij}) \frac{1-\rho}{1-Q_{ij}} \\ &\quad \times \left[\frac{Q_{ij}\omega_{rs}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{rs})}{1-\rho} \right]. \end{aligned} \tag{4.31}$$

Substituting these into (4.29), we get

$$q_{rs}^{ij} = \frac{\eta_r^{i\to j} \eta_s^{j\to i} \left[\frac{Q_{ij}\omega_{rs}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{rs})}{1-\rho} \right]}{\sum_{rs} \eta_r^{i\to j} \eta_s^{j\to i} \left[\frac{Q_{ij}\omega_{rs}}{\rho} + \frac{(1-Q_{ij})(1-\omega_{rs})}{1-\rho} \right]}. \tag{4.32}$$

Our final algorithm then consists of alternately (a) iterating the belief propagation equations (4.25) to convergence and using the results to calculate the marginal probabilities q_r^i and q_{rs}^{ij} from Eqs. (4.27) and (4.32), and (b) iterating Eqs. (4.16) and (4.24) to convergence to calculate new values of the ω_{rs} and using Eq. (4.14) to calculate new values of γ_r. In practice the algorithm is efficient—in other tests of belief propagation it has been found fast enough for applications to networks of a million nodes or more.

4.2.4 Degree-corrected model

Our method gives a complete algorithm for fitting the standard stochastic block model to uncertain network data represented by the matrix $\mathbf{Q}$ of edge probabilities. As pointed out previously by Karrer and Newman [95], however, the stochastic block model gives poor performance for community detection on many real-world networks because the model assumes a Poisson degree distribution, which is strongly in conflict with the broad, frequently fat-tailed degree distributions seen in real-world networks. Because of this conflict it is often not possible to find a good fit of the stochastic block model to observed network data, for any parameter values, and in such cases the model can return poor performance on community detection tasks.

The fix for this problem is straightforward. The *degree-corrected stochastic block model* is identical to the standard block model except that the probability of an edge between nodes i, j that fall in groups r, s is $d_i d_j \omega_{rs}$ (instead of just ω_{rs}), where d_i is the actual degree of node i in the network. This modification allows the model to accurately fit arbitrary degree distributions, and community detection algorithms that perform fits to the degree-corrected model are found to return excellent results in real-world applications [95].

We can make the same modification to our methods as well, estimating d_i with $\sum_j Q_{ij}$. The developments follow exactly the same lines as for the ordinary (uncorrected) stochastic block model. The crucial equations (4.16) and (4.24) become

$$t_{rs}^{ij} = \frac{Q_{ij} d_i d_j \omega_{rs}/\rho}{Q_{ij} d_i d_j \omega_{rs}/\rho + (1 - Q_{ij})(1 - d_i d_j \omega_{rs})/(1 - \rho)} \tag{4.33}$$

and

$$\omega_{rs} = \frac{\sum_{ij} q_{rs}^{ij} t_{rs}^{ij}}{\sum_i d_i q_r^i \sum_j d_j q_s^j}, \tag{4.34}$$

while the belief propagation equation (4.25) becomes

$$\begin{aligned}\eta_r^{i\to j} &= \frac{\gamma_r}{Z_{i\to j}} \exp\biggl(-d_i d_j \sum_{k,s} q_s^k \omega_{rs}\biggr)\\ &\quad\times \prod_{\substack{k(\neq j)\\ Q_{ik}\neq 0}} \sum_s \eta_s^{k\to i} \biggl[\frac{Q_{ik} d_i d_j \omega_{rs}}{\rho} + \frac{(1-Q_{ik})(1-d_i d_j \omega_{rs})}{1-\rho}\biggr],\end{aligned} \tag{4.35}$$

with corresponding modifications to Eqs. (4.26) to (4.28) and Eq. (4.32).

In the following sections we describe a number of example applications of our methods. Among these, the tests on synthetic networks (Section 4.3.1) are performed using the standard stochastic block model, without degree-correction, while the tests on real-world networks (Section 4.3.2) use the degree-corrected version.

4.3 Results

We have tested the methods described in the previous sections both on computer-generated benchmark networks with known structure and on real-world examples.

4.3.1 Synthetic networks

Computer-generated or synthetic networks provide a controlled test of the performance of our algorithm. We generate networks with known community structure planted within them and then test whether the algorithm is able accurately to detect that structure.

For the tests reported here, we generate networks using the standard (not degree-corrected) stochastic block model and then add noise to them to represent the network uncertainty, using functions β_0 and β_1 as defined in Section 4.2.1. We use networks of size $n = 4000$ nodes, divided into two equally-sized communities, and as the noise function $\beta_1(Q)$ for the edges we use a beta distribution:

$$\beta_1(Q) = \frac{Q^{a_1-1}(1-Q)^{b_1-1}}{\mathrm{B}(a_1, b_1)}, \tag{4.36}$$

where $\mathrm{B}(a, b)$ is Euler's beta function. As the noise function $\beta_0(Q)$ for the nonedges we use a beta distribution plus an additional delta-function spike at zero:

$$\beta_0(Q) = c\frac{Q^{a_0-1}(1-Q)^{b_0-1}}{\mathrm{B}(a_0, b_0)} + (1-c)\delta(Q). \tag{4.37}$$

The delta function makes the matrix $\mathbf{Q}$ of edge probabilities realistically sparse, in keeping with the structure of real-world data sets, with a fraction $1-c$ of nonedges having exactly zero probability in the observed data, on average.

Thus there are a total of five parameters in our noise functions: a_0, b_0, a_1, b_1, and c. Not all of these parameters are independent, however, because our functions still have to satisfy the constraint (4.4). Substituting Eqs. (4.36) and (4.37) into (4.4), we see that for the constraint to be satisfied for all $Q > 0$ we must have $a_0 = a_1 - 1$, $b_0 = b_1 + 1$, and

$$\begin{aligned} c &= \frac{1-\rho}{\rho}\frac{\mathrm{B}(a_1, b_1)}{\mathrm{B}(a_0, b_0)} = \frac{1-\rho}{\rho}\frac{\mathrm{B}(a_1, b_1)}{\mathrm{B}(a_1-1, b_1+1)} \\ &= \frac{1-\rho}{\rho}\frac{a_1-1}{b_1}. \end{aligned} \tag{4.38}$$

Thus there are really just two degrees of freedom in the choice of the noise functions. Once we fix the parameters a_1 and b_1, everything else is fixed also. Alternatively, we can fix the parameter c, thereby fixing the density of the data matrix $\mathbf{Q}$, plus one or other of the parameters a_1 and b_1.

The networks we generate are now analyzed using the non-degree-corrected algorithm of Sections 4.2.1 to 4.2.3. We estimate ρ from $\mathbf{Q}$ and assume we know k. To quantify performance we assign each node i to the community r for which its probability q_r^i of membership, Eq. (4.10), as computed by the algorithm, is greatest, then compare the result to the known true community assignments from which the network was generated. Success (or lack of it) is quantified by computing the fraction of nodes placed by the algorithm in the correct groups. We also compare the results against the naive (but common) thresholding method discussed in the introduction [104], in which edge probabilities Q_{ij} are turned into binary yes-or-no edges by cutting them off at some fixed threshold τ, so that the adjacency matrix element A_{ij} is 1 if and only if $Q_{ij} > \tau$. Community structure in the thresholded network is analyzed using the standard stochastic block model algorithm described in, for example, Refs. [50] and [51].

As we vary the parameters of the underlying network and noise functions the performance of both algorithms varies. When the community structure is strong and the noise is weak both algorithms (not surprisingly) do well, recovering the community structure nearly perfectly, while for weak enough community structure or strong noise neither algorithm does better than chance. But, as shown in Fig. 4.3a, there is a regime of intermediate structure and noise in which our algorithm does significantly better than the naive technique. The figure shows the fraction of correctly classified nodes in the naive algorithm as a function of the threshold τ (data points in the figure) compared against the performance of the algorithm of this chapter (dashed line) and, as we can see, the latter outperforms the former no matter what value of τ is used. Note that the worst possible performance still classifies a half of the nodes correctly—even a random coin toss would get this many right—so this is the minimum value on the plot. For high threshold values τ approaching one, the threshold method throws away essentially all edges, leaving itself no data to work with, and hence does little better than chance. Conversely for low thresholds the threshold method treats any node pair with a nonzero connection probability Q_{ij} as having an edge, even

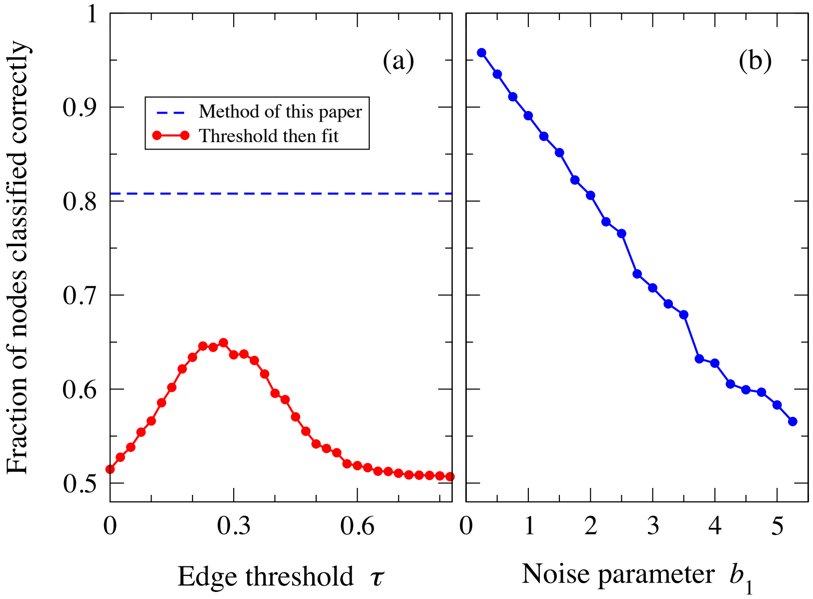


Figure 4.3: Tests of the method described in this chapter on synthetic benchmark networks. (a) Fraction of nodes placed in the correct community for uncertain networks generated using a stochastic block model with $n = 4000$ nodes, two groups of equal size, edge probabilities $\omega_{11} = \omega_{22} = 0.02$, $\omega_{12} = \omega_{21} = 0.014$, and noise parameters $a_1 = 1.4$ and $b_1 = 2$ (see Eq. (4.36)). The horizontal dashed line shows the performance of the algorithm described in this chapter. The points show the performance of a naive algorithm in which the uncertain network is first converted to a binary network by thresholding the edge probabilities and the result then fed into a standard community detection algorithm. The results for each algorithm are averaged over 20 repetitions of the experiment with different networks. Statistical errors are comparable in size to the data points. (b) Fraction of nodes classified into their correct communities for stochastic block model networks with varying amounts of noise in the data. The parameters are the same as for (a) but with the sparsity parameter c fixed at $1/4n$ (see Eq. (4.37) and the ensuing discussion) and varying the parameter b_1, which controls the level of noise in the data.

when an edge is wildly unlikely, thereby introducing large amounts of noise into the calculation that again reduce performance to a level little better than chance. The optimal performance falls somewhere between these two extremes, around $\tau = 0.25$ in this case, but even at this optimal point the thresholding method's performance falls far short of the algorithm of this chapter.

Figure 4.3b shows a different test of the method. Again we use networks generated from a stochastic block model with two groups and calculate the fraction of correctly classified nodes. Now, however, we vary the amount of noise introduced into the network to test the algorithm's ability to recover structure in data of varying quality. The parameters of the underlying network are held constant, as is the parameter c that controls the sparsity of the data matrix $\mathbf{Q}$. This leaves only one degree of freedom, which we take to be the parameter b_1 of the noise process (see Eq. (4.36)).

A network with little noise in the data is one in which true edges in the underlying network are represented by probabilities Q_{ij} close to 1, in other words by a noise distribution $\beta_1(Q)$ with most of its weight close to 1. Such distributions correspond to small values of the parameter b_1. Noisier data are those in which the values of the Q_{ij} are smaller, approaching the values for the nonedges, thereby making it difficult to distinguish between edges and nonedges. These networks are generated by larger values of b_1. Figure 4.3b shows the fraction of correctly classified nodes as a function of b_1, so the noise level is increasing, and the quality of the simulated data decreasing, from left to right in the figure.

As we can see, the algorithm returns close to perfect results when b_1 is small—meaning that the quality of the data is high and the algorithm almost sees the true underlying structure of the network. Performance degrades as the noise level increases, although the algorithm continues to do significantly better than chance even for high levels of noise, indicating that there is still useful information to be extracted even from rather poor data sets.

4.3.2 Protein interaction network

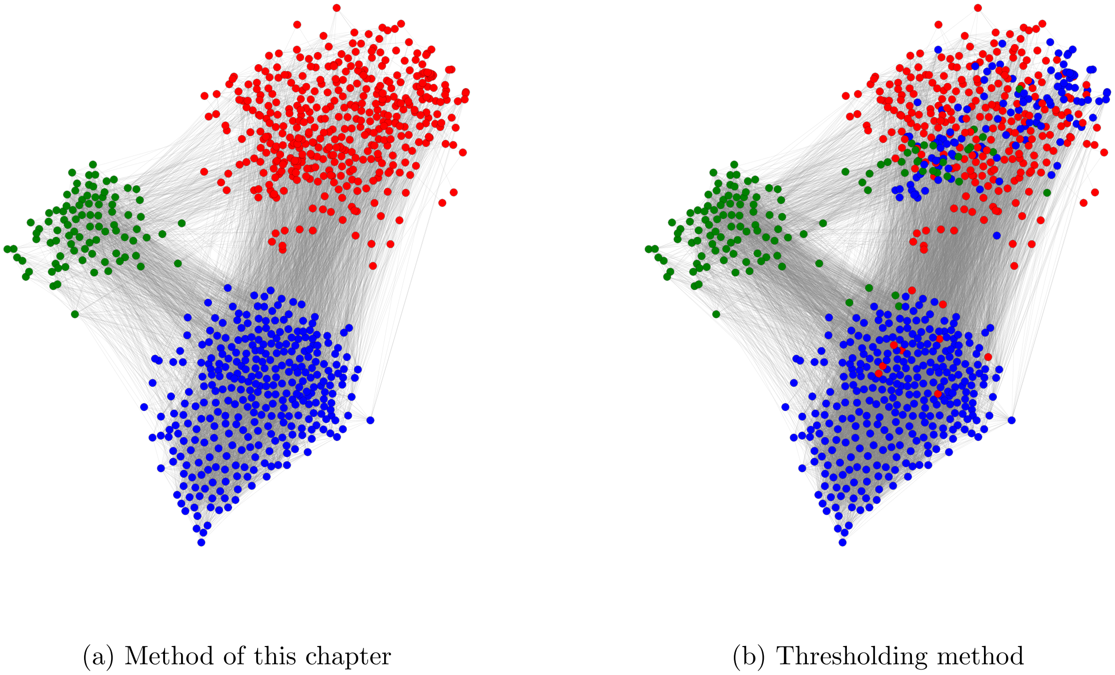
As a real-world example of our methods we have applied them to protein-protein interaction networks from the STRING database [174]. This database contains protein interaction information for 1133 species drawn from a large body of research literature covering a range of different techniques, including direct interaction experiments, genomic information, and cross-species comparisons. The resulting networks are of exactly the form considered in this chapter. For each network there is assumed to be a true underlying network in which every pair of proteins either interacts or doesn't, but, given the uncertainty in the data on which they are based, STRING provides only probabilistic estimates of the presence of each interaction. Thus the data we have for each species consists of a set of proteins—the nodes—plus a likelihood of interaction for each protein pair. A significant majority of protein pairs in each of the networks are recorded as having zero probability of interaction, so the network is sparse in the sense assumed by our analysis and conducive to fast computation.

In the STRING database as well as the work of Krogan *et al.* [104], protein pairs are recorded as having zero interaction probability when they never bind in high throughput experiments. Though a true zero probability of interaction is unlikely due to the possibility of human or equipment error, proteins which do not bind are most likely to have a value of zero. In principle one could add a small estimate of error to every cell of the matrix, but a small enough error would make no difference in the final outcome.

We analyze the data using the degree-corrected version of our algorithm described in Section 4.2.4, which is appropriate because the networks in the STRING database, like most real-world networks, have broad degree distributions.

Figure 4.4a shows the communities found in a three-way split of the protein-protein interaction network of the bacterium *Borrelia hermsii* HS1. Node colors denote the strongest community affiliation for each node, as quantified by the one-node marginal probability q_r^i, with node size being proportional to the probability a node is in its most likely community (so that larger nodes are more certain). In practice, most nodes belong wholly to just one community.

For comparison, we also show in Fig. 4.4b the communities found in the same network by the naive thresholding algorithm discussed earlier in which a node pair i, j is considered connected by an edge if and only if the probability Q_{ij} exceeds a certain



(a) Method of this chapter

(b) Thresholding method

Figure 4.4: Communities found by (a) the algorithm described in this chapter and (b) the thresholding algorithm, in a three-way split of the protein interaction network of the bacterium *Borrelia hermsii* HS1, taken from the STRING database. Nodes are laid out according to the communities in (a) and the layout is the same in both panels.

threshold, which here is set at 0.25, though other thresholds gave similar results. By contrast with the synthetic networks of the previous section, we do not know the true underlying communities for this network and so cannot calculate the fraction of correctly classified nodes, but it is clear from the figures that the new technique gives significantly different results from the thresholding method, particularly for the community that appears in the upper right of the figure.

A closer examination of the data reveals a possible explanation. The communities at the left and bottom in both panels of Fig. 4.4 consist primarily of high-probability edges and are easily identified in the data, so it is perhaps not surprising that both algorithms identify these communities readily and are largely in agreement. However, the third community, in the upper right of the figure, consists largely of edges of relatively low probability and the thresholding method has more difficulty with this case because many edges fall below the threshold value and so are lost, which may explain why the thresholding method divides the nodes of this community among the three groups.

To give a simple picture, imagine a community whose nodes are connected by very many internal edges, but all of those edges have low probability. Because there are so many of them, the total expected number of true internal edges in the underlying network—the number of node pairs times the average probability of connection—could be quite high, high enough to create a cohesive network community. Our algorithm, which takes edge probabilities into account, will allow for this. The thresholding algorithm on the other hand can fail because the edges all have low probability, below the threshold used by the algorithm, and hence are discarded. The result is that the thresholding algorithm sees no edges at all and hence no community. The fundamental problem is that thresholding is just too crude a tool to see subtle patterns in noisy data.

4.4 Edge recovery

A secondary goal in our analysis of uncertain networks is to deduce the structure of the (unobserved) underlying network from the uncertain data. That is, given the matrix **Q** of edge probabilities, can we make an informed guess about the adjacency matrix **A**? We call this the *edge recovery* problem. It is related to, but distinct from, the well studied *link prediction* problem [112], in which one is given a binary network of edges and nonedges but some of the data may be erroneous and the problem is to guess which ones. In the problem we consider, by contrast, the data given are assumed to be correct, but they are incomplete in the sense of being only the probabilities of the edges, rather the edges themselves.

The simplest approach in the present case is simply to use the edge probabilities Q_{ij} themselves to predict the edges—those node pairs i, j with the highest probabilities are assumed most likely to be connected by edges. But if we know, or believe, that our network contains community structure, then we can do a better job. If we know where the communities in the network lie, at least approximately, then given two pairs of nodes with similar values of Q_{ij}, the pair that are in the same community should be more likely to be connected by an edge than the pair that are not (assuming assortative mixing).

It turns out that our EM algorithm gives us precisely the information we need to combine our edge probabilities and network structure to perform edge recovery. The posterior probability (given network parameters $\boldsymbol{\gamma}, \boldsymbol{\omega}$ and final data **Q**) of having an edge between any pair of nodes i, j can be written as

$$\begin{aligned} &P(A_{ij} = 1|\mathbf{Q}, \boldsymbol{\gamma}, \boldsymbol{\omega}) \\ &\quad = \sum_{rs} P(A_{ij} = 1|g_i = r, g_j = s, \mathbf{Q}, \boldsymbol{\gamma}, \boldsymbol{\omega}) P(g_i = r, g_j = s|\mathbf{Q}, \boldsymbol{\gamma}, \boldsymbol{\omega}) \\ &\quad = \sum_{rs} t^{ij}_{rs} q^{ij}_{rs}, \end{aligned} \tag{4.39}$$

where we have made use of Eq. (4.22) and the definition of q_{rs}^{ij}. Both t_{rs}^{ij} and q_{rs}^{ij} are calculated in the course of running the EM algorithm, so we already have these quantities available to us and calculating $P(A_{ij} = 1)$ is a small extra step.

Figure 4.5 shows a test of the accuracy of our edge predictions using synthetic test networks once again. In these tests we generate networks with community structure using the standard stochastic block model, as previously, then run the network through the EM algorithm and calculate the posterior edge probabilities of Eq. (4.39) above. We compare the results against competing predictions based on the prior edge probabilities Q_{ij} alone.

The figure shows *receiver operating characteristic* (ROC) curves of the results. To construct an ROC curve, one asks how many edges we would get right, and how many nonedges we would get wrong, if we were to simply predict that the fraction x of node pairs with the highest probabilities of connection are in fact connected by edges. The ROC curve is the plot of the fraction of true edges correctly predicted (true positive rate, or TPR) against the fraction of nonedges incorrectly predicted (false positive rate, or FPR) for values of x from zero to one. By definition the curve always lies on or above the 45-degree line and the higher the curve the better the results, since a higher curve implies more true positives and fewer false ones.

Figure 4.5 shows the ROC curves both for our method and for the naive method based on the raw probabilities Q_{ij} alone and we can see that, for the particular networks studied here, the additional information revealed by fitting the block model results in a substantial improvement in our ability to identify the edges of the network correctly. One common way to summarize the information contained in an ROC curve is to calculate the area under the curve, where an area of 0.5 corresponds to the poorest possible results—no better than a random guess—and an area of 1 corresponds to perfect edge recovery. For the example shown in Fig. 4.5, the area under the curve for our algorithm is 0.89 while that for the naive algorithm is significantly lower at 0.80.

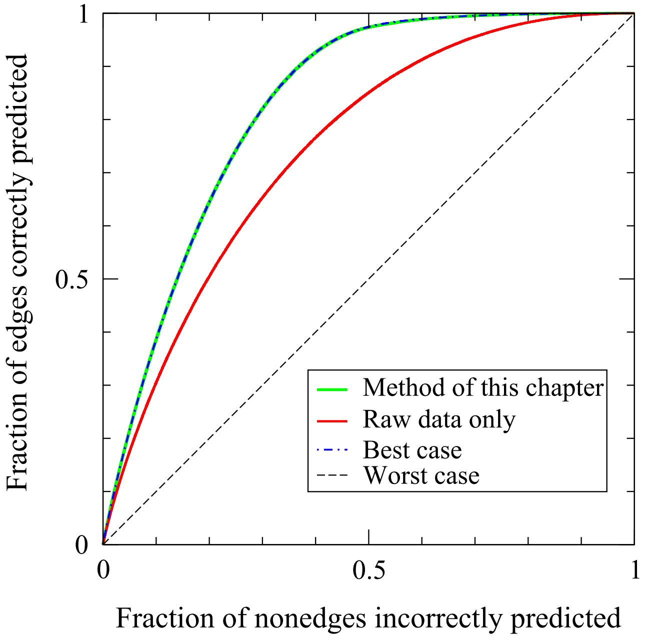


Figure 4.5: Receiver operating characteristic (ROC) curves for the edge recovery problem on a synthetic network generated using a two-group stochastic block model with $n = 4000$ nodes, $\omega_{11} = \omega_{22} = 0.05$, $\omega_{12} = \omega_{21} = 0.001$, and noise parameters $b_1 = 4$ and $c = 1/4n$. The three curves show the performance of the algorithm of this chapter, the naive algorithm based on the raw probabilities Q_{ij} alone, and a hypothetical ideal algorithm that knows the values of the parameters used to generate the model (so that one does not have to run the EM algorithm at all). The diagonal dashed line represents is curve generated by an algorithm that does no better than chance.

Also shown in the figure is a third curve representing performance on the edge recovery task if we assume we know the exact parameters of the stochastic block model that were used to generate the network, i.e., that we don't need to run the EM algorithm to learn the parameter values. This is an unrealistic situation—we very rarely know such parameters in the real world—but it represents the best possible prediction we could hope to make under any circumstances. And, as the figure shows, this best possible performance is in this case indistinguishable from the performance of our EM algorithm, indicating that the EM algorithm is performing the edge recovery task essentially optimally in this case.

4.5 Conclusions

In this chapter I have described methods for the analysis of networks represented by uncertain measurements of their edges. I gave a method for performing the common task of community detection on such networks by fitting a network generative model to the data using a combination of an EM algorithm and belief propagation. I also show how the resulting fit can be used to reconstruct the true underlying network by making predictions of which nodes are connected by edges. Using controlled tests on computer-generated benchmark networks, I show that these methods give better results than previously used techniques that rely on simple thresholding of probabilities to turn indefinite networks into definite ones. And I have given an example application of our methods to a bacterial protein interaction network taken from the STRING database.

The methods described in this chapter could be extended to the detection of other types of structure in networks. If one can define a generative model for a structure of interest then the developments of Section 4.2 can be applied, simply replacing the likelihood $P(\mathbf{A},\boldsymbol{g}|\boldsymbol{\gamma},\boldsymbol{\omega})$ in Eq. (4.7) with the appropriate probability of generation. Generative models have been recently proposed for hierarchical structure in networks [39], overlapping communities [2], ranking or stratified structure [9], and others. In principle, these methods could be extended to any of these structure types in uncertain networks.

CHAPTER V

Coauthorship and citation

5.1 Introduction

Citation networks [153] and coauthorship networks [78, 77, 135] are distinct network representations of bodies of academic literature that have both been the subject of quantitative analysis in recent years. In a citation network the nodes are papers and a directed edge runs from paper A to paper B if A cites B in its bibliography. In a coauthorship network the nodes are authors and an undirected edge connects two authors if they have written a paper together. Both kinds of network can shed light on habits and patterns of academic research. Citation networks, for instance, can give a picture of the topical connections between papers, while coauthorship networks can shed light on patterns of collaboration such as the size of collaborative groups or the frequency of repeated collaboration.

In this chapter we analyze networks of citation and coauthorship derived from a large data set made available by the American Physical Society (APS), which consists of bibliographic and citation data for the Physical Review family of physics journals and spans the entire history of those journals, more than a hundred years, from their inception in 1893 to 2009.[1] The data set is unusual both because of the length of time it spans and also because it contains information on both citation and coauthorship

[1] More details about the data set can be found on the web at https://publish.aps.org/datasets.

for the same body of literature. A number of previous analyses of the data have been published [159, 33, 164] but our work adopts a somewhat different viewpoint from other studies in focusing on the interactions between authorship and citation, as well as on long time-scale patterns in the data. In particular, the simultaneous availability of citation and coauthorship data allow us to associate citations not only with papers but with individual authors, so that we can tell whether or not a particular author cites another. Combining this insight with the temporal aspects of the data we find, for example, that researchers cite their own or coauthors' papers more quickly after publication than they do the work of others; that authors show a strong tendency to return the favor of a citation from another author, especially a previous coauthor; that, contrary to some recent conjectures, having a common coauthor does not make two authors likely to collaborate in future [135, 45, 87]; and that there has not (at least within the journals we study) been any increase over time in self-citations, the number holding roughly constant at about 20% of all citations for over a century. Overall, we contribute an innovative model for understanding academic networks and add to the body of knowledge about scientific publishing.

5.2 The data set

In its raw form the data set we study contains records for 462 090 papers published in the various Physical Review journals, each identified with a unique numerical label. Data for each paper include paper title, date of publication, the published names and affiliations of each of the authors, and a list of the numerical labels of previous Physical Review papers cited. The data set is unusual in two respects: the long period of time it covers, which spans 116 years from 1893 to 2009, and the fact that it includes citation data and hence allows us to compare coauthorship patterns with citations, at least for that portion of the citation network that appears in the Physical Review—citations to and from non-Physical-Review journals, of which there are many, are not included.

Before performing any analysis, however, there are some hurdles to overcome. Foremost among them is the fact that the name of an author alone does not necessarily identify him or her uniquely. Two authors may have the same name, or the same author may be identified differently in different publications (with or without a middle initial, for example). Unlike some journals, such as those of the American Mathematical Society,[2] the Physical Review does not maintain unique author identifiers that can be used to attribute authorship unambiguously. As a first step in analyzing the data, therefore, we have processed it using a number of disambiguation techniques in order to infer actual author identity from author names as accurately as possible. Details of the disambiguation process are given in Appendix B.1.

In addition, we have performed a modest culling of the data to remove outliers, the most substantial action being the removal of all papers with fifty or more authors, which are primarily recent papers in experimental high-energy physics. (Almost all of them, about 91%, were published either in Physical Review D, which covers high-energy physics, or Physical Review Letters; the remainder were in Physical Review C, which covers nuclear physics.) As we show shortly, though papers with more than fifty authors are only a small fraction of the whole (about 0.7%), their inclusion skews results for the last thirty years substantially by comparison with the rest of the time period. For results whose outcome depends strongly on the presence or not of these papers, we quote results both with and without, for comparison.

Table 5.1 gives some basic parameters of the resulting data set.

[2] A description of the unique author identifier system used by the American Mathematical Society can be found at http://www.istl.org/01-summer/databases.html.

	All papers	Papers with 50 authors or fewer
Total papers	460889	457516
Total authors	235533	226641
Authors per paper	5.35	3.34
Citations per paper	10.16	10.16
Number of collaborators	59.44	17.24
Papers per author	10.47	6.74

Table 5.1: Mean values of some statistics for our data set, with and without papers having over 50 authors.

5.3 Analysis

In the next few sections we present a variety of analyses of the Physical Review data set. We begin by looking at some basic parameters of authorship and coauthorship.

5.3.1 Authorship patterns

Figure 5.1 shows a cumulative distribution function for the number of papers an author publishes, aggregated over the entire data set. That is, the figure shows the fraction of authors who published n papers or more as a function of n, which is a crude measure of scientific productivity. The axes in the figure are logarithmic, and the approximate straight-line form of the distribution function implies that scientific productivity follows, roughly speaking, a power law, a result known as Lotka's law, first observed by Alfred Lotka in 1926 [113] and confirmed by numerous others since. (It has also been suggested that the distribution is log-normal rather than power-law [168]. It is known to be hard to distinguish empirically between log-normal and power-law distributions [40].) In Fig. 5.1 we give separate curves with and without the papers that have fifty or more coauthors. As the figure shows, the difference between the two is primarily in the tail of the distribution, among the authors who have published the largest number of papers, indicating that a significant fraction

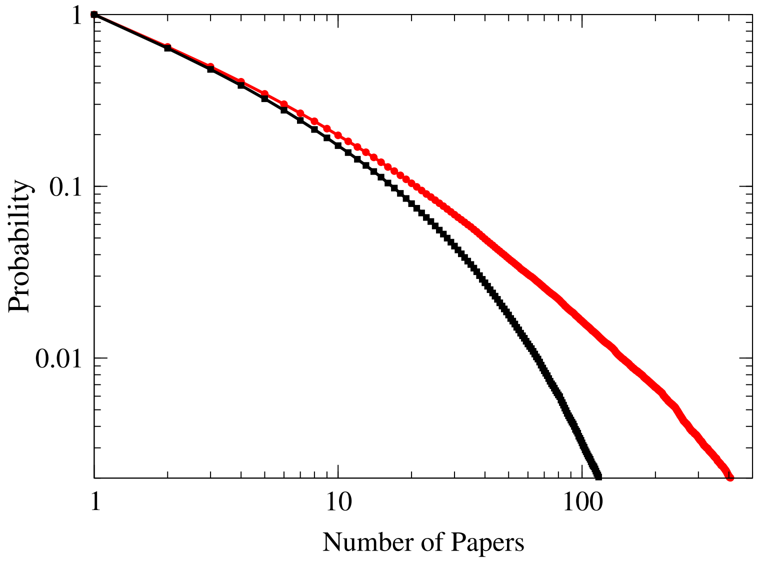


Figure 5.1: Probability that an author wrote more than a given number of papers. Red circles indicate values calculated from the full data set; black squares are values from the data set after papers with fifty or more authors have been removed. The plot is cut off around 500 papers because there is very little data beyond this point.

of the most productive authors are those in large collaborations. In fact, if one compiles a list of the fifty authors publishing the largest numbers of papers, only one of them remains on that list after papers with fifty or more authors are excluded. This probably results from a combination of two effects: first, larger groups can publish more papers simply because they have more people available to write them; and second, a large and productive group of collaborators contributes many apparently prolific authors to the statistics—each of the many coauthors separately gets credit for being highly productive. It is precisely because of biases of this kind that we exclude papers with many authors from some of our calculations.

We can remedy this problem to some extent by measuring productivity in a more sophisticated fashion. Rather than just counting up all the papers an author was listed on, we can instead divide up the authorship credit for a paper among the contributing authors so that, for example, each author on a two-author paper is credited with half an authorship for that paper. This reduces significantly the impact of large collaborations on the statistics, though the distribution of number of papers

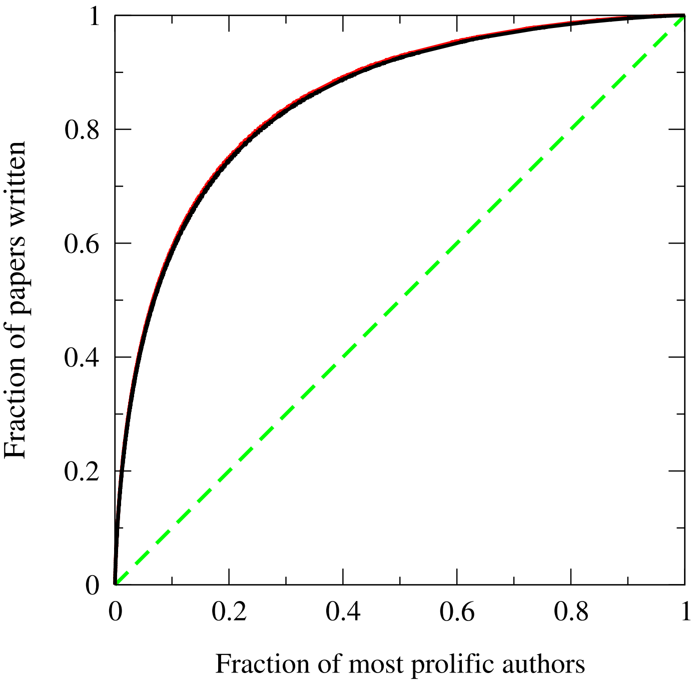


Figure 5.2: Fraction of papers written by the most prolific authors (with credit for multi-author papers divided among coauthors, as described in the text). The red (grey) curve represents values calculated from the full data set; the black curve represents values after papers with fifty or more authors have been removed. Note that the two curves are almost indistinguishable. The dashed line indicates the form the curve would take if all authors published the same number of papers.

authored is still highly skewed, with certain authors producing much more science than others. A common way to visualize such skewed distributions is to use a Lorenz curve, a plot of the fraction of papers produced by the most prolific authors against the fraction of authors that produced them. Such a curve is shown for our data set in Fig. 5.2, and the sharp rise in the curve at the left-hand side indicates the concentration of scientific productivity among the most productive scientists. Note for instance that productivity appears roughly to follow the so-called 80–20 rule, such that about 80% of the output is produced by the 20% most productive authors. Notice also that there is almost no difference in the Lorenz curves with and without the 50-plus-author papers, precisely because we have divided up the authorship credit so that the effect of many-author papers is diminished.

The distribution can be further quantified by measuring a Gini coefficient, which is defined as the excess area under the Lorenz curve compared to the case where everyone has the exact same productivity. In our data set, the Gini coefficient is 0.70, a relatively large figure as such coefficients go, indicating high skew. (Gini coefficients are perhaps best known in the context of wealth inequality. For comparison, the Gini coefficient of the global household wealth distribution in 2000 was 0.892 [46].)

The data set also allows us to measure the productivity of the entire field of physics over time, something that cannot be done with many other data sets. Figure 5.3 shows the total number of papers published in the Physical Review in five year time blocks since 1893. With the important caveat that these results are for a single collection of journals only, and one moreover whose role within the field has evolved during its history from provincial up-start to one of the leading physics publications on the planet, we see that there is a steady increase in the volume of published work, which appears roughly to follow an exponential law (a straight line on the semi-logarithmic scales of the figure). An interesting feature is the dip in the curve in the 1940s, which coincides with the second World War, followed by a recovery in the 1950s, perhaps

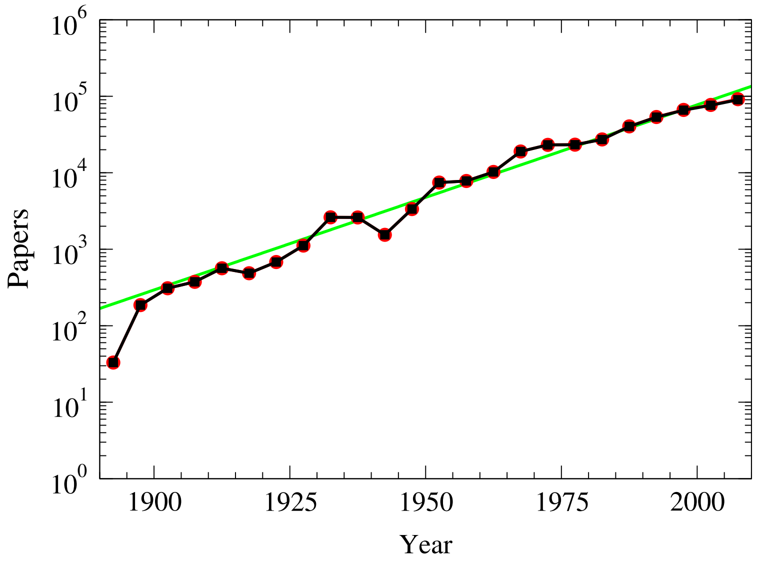


Figure 5.3: Number of papers published in each five-year block. Red circles indicate numbers calculated from the full data set; black squares are calculated from the data set after papers with fifty or more authors have been removed. Note that the two values are almost indistinguishable. The straight line is the best-fit exponential.

attributable in part to increased science funding in the postwar period. The combined result of these deviations, however, is only to put the curve back on the same path of exponential growth after the war that it was already on before it. In his early studies of secular trends in scientific output, Derek de Solla Price [48, 49] noted a similar exponential growth interrupted by the war, and measured the doubling time of the growth process to be in the range from 10 to 15 years. The best exponential fit to our data gives a compatible figure of 11.8 years.

Figure 5.4 shows the corresponding plot of the number of unique authors in the data set in each five-year block as a function of time. Like the number of papers published, the number of authors appears to be increasing exponentially, and with a roughly similar (but slightly smaller) doubling time of 10.4 years. Thus, despite the marked increase in productivity of the field as a whole, it appears that each individual scientist has produced a roughly constant, or even slightly decreasing, number of papers per year over time.

The natural complement to measurement of the number of papers per author is

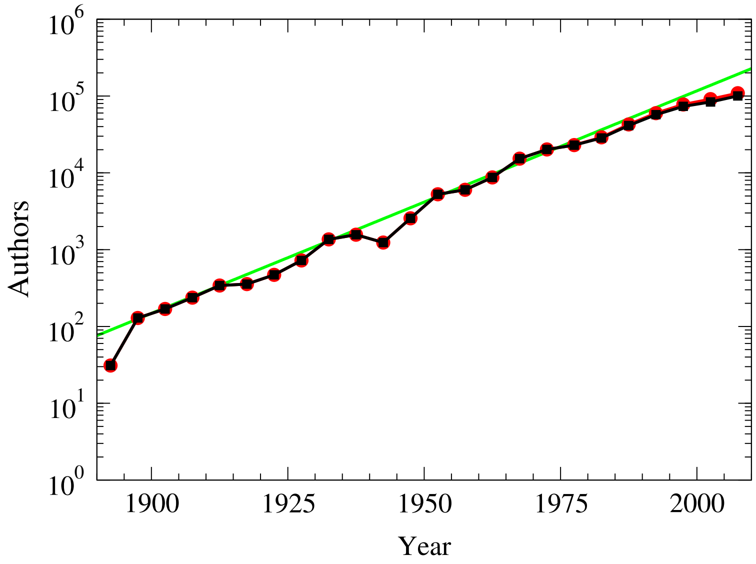


Figure 5.4: Number of unique authors who published a paper in each five-year block. Red circles indicate numbers calculated from the full data set, while black squares are calculated from the data set after papers with fifty or more authors have been removed. Note that the two values are almost indistinguishable. The straight line is the best-fit exponential.

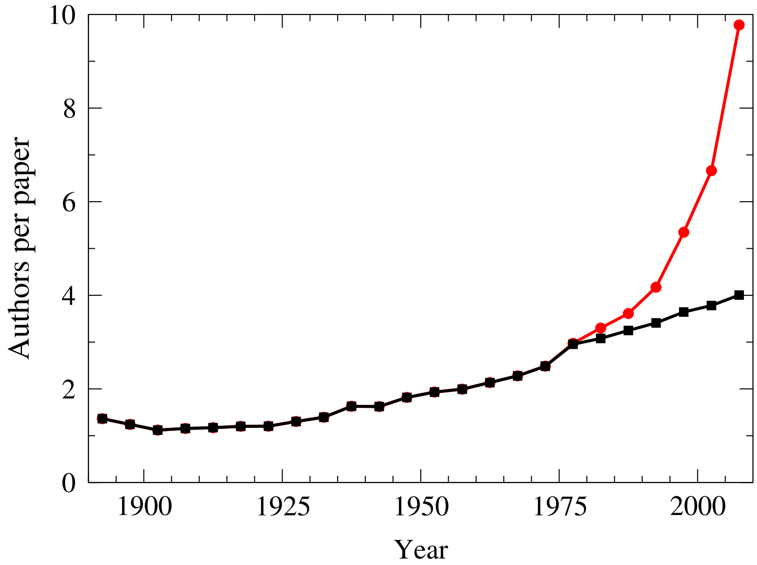


Figure 5.5: Number of authors per paper averaged over five-year blocks. Red circles indicate the full data set; black squares are the data set after papers with fifty or more authors have been removed.

measurement of the number of authors per paper, i.e., the size of collaborative groups. Figure 5.5 shows the mean number of authors per paper in our data set as a function of time, and there is a clear increasing trend throughout most of the time period covered, with the average size of a collaborative group rising from a little over one a century ago to about four today. A similar effect has been noted previously by, for example, Grossman and Ion [77], for the case of mathematics collaborations. In our calculations we have again calculated separate curves with and without papers having fifty or more authors and a comparison between the two reveals a startling effect: while there is almost no difference at all between the curves prior to about 1975, there is a large and rapidly growing gap between them in the years since. Without these papers the growth in group sizes has been slow and steady for decades; with them it departs dramatically from historical trends after the 1970s, indicating a large and growing role in physics (or at least in physics publication) for big collaborations.

An alternative view of the same trend is given in Fig. 5.6, which shows the number of unique coauthors an author has, on average, during each five year time block. Every coauthor in a time block is counted, even if he or she was also counted in a previous time block (but previous coauthors are not counted unless they are also coauthors in the new time block). As the figure shows, this number has also risen significantly over the last century, from a little over one to more than ten today (and more than sixty if one includes collaborations with fifty or more members). Since we only have data from the Physical Review, it is likely that we miss some collaborators, so these numbers are in practice only lower bounds on the actual numbers.

5.3.2 Citation patterns

Let us now add the citation portion of the data set to our analyses and examine citation patterns over time in the Physical Review, as well as interactions between citation and coauthorship.

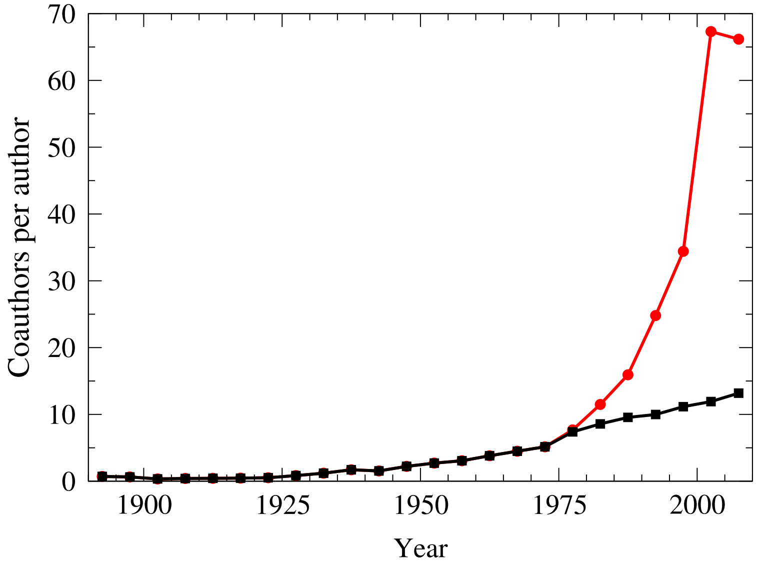


Figure 5.6: Average number of unique coauthors of an author, averaged in five-year blocks. Red circles indicate the full data set; black squares are the data set after papers with fifty or more authors have been removed.

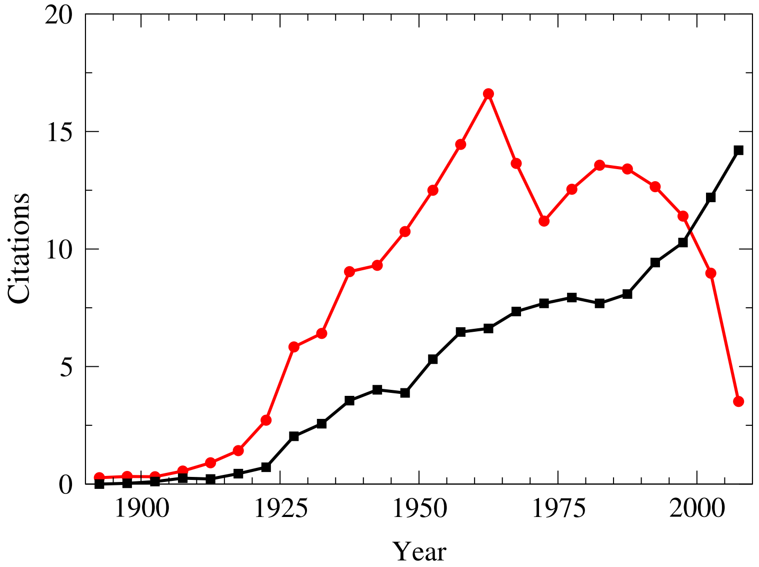


Figure 5.7: Average numbers of citations made (black squares) and received (red circles) per paper, in five-year blocks.

Figure 5.7 shows the average number of citations by a paper and to a paper, over the time period covered by the Physical Review data set. The black curve, the number of citations that a paper makes, shows a steady increase over time—authors used to cite fewer papers and have been citing steadily more in recent decades. One possible explanation for this phenomenon is the increase in the volume of literature available to be cited, although it has also been conjectured that authors have been under greater pressure in recent decades, for example from journal editors or referees, to add more copious citations to papers [185].

The red curve in Fig. 5.7 is the average number of citations received by a paper, which shows more irregular behavior, rising to a peak twice before dropping off in recent times. A number of effects are at work here. First, if (as we will shortly see) most citations are to papers in the recent past, then a steady increase in citations *by* papers should lead to an increase in citations *to* papers published slightly earlier. Behavior of this kind has been observed in previous studies, such as the comprehensive study by Wallace *et al.* using data from the Web of Science [177]. The growth in number of citations received cannot continue to the very end of the data set, however, since the most recent papers are too recent to have accrued a significant number of citations and hence we expect a drop at the rightmost end of the curve, as seen in the figure.

There is, however, also a notable dip in the red curve around 1970, whose origin is less clear. (It is not seen, for instance, in the work of Wallace *et al.*) In examining the data for this period in detail, we find that the dip in citations per paper is due primarily to an increase in the number of papers published in the Physical Review (which expanded considerably during this period), while the number of citations received by those papers, in aggregate, remains roughly constant. The increase in papers published may have been in part a response to the general expansion of US physics research during the 1960s, following the establishment of the National Science

Foundation, but the data indicate that the greater volume of research did not, at least initially, result in a greater number of citations received, and hence the ratio of the two displays the dip visible in Fig. 5.7. However, the upward trend in the curve reestablishes itself from about 1970 onward, suggesting that in the long run there was an increase not only in the number of papers published, but also in the number that are influential enough to be later cited.

It is interesting to compare the data for citations received with the predictions of theoretical models for the citation process. Perhaps the best known class of models are the preferential attachment models we describe in Section 1.2.7 [14], and particularly the 1976 model of Price [154], a simple model in which the rate at which a paper receives citations is assumed to vary linearly with the number it already has. In its most naive application, this model makes predictions that differ strongly from the observations plotted in Fig. 5.7. The model predicts that the largest number of citations should go to the oldest papers and the smallest to the youngest, so that the red curve in the figure should be monotonically decreasing. There are a number of possible explanations for the disagreement. A popular theory is that papers “age” over time, becoming less well cited as they become older [193, 166], perhaps because their field has moved on to other things, because they have been superseded by more advanced or accurate work, or because their results are so well known that authors no longer feel the need to cite them. Were this the case, most citations would be to recent papers, and the curve of citations received would mostly mirror the curve of citations given, albeit with a time lag whose length would be set by the rate at which papers age. An alternative theory, for which there is some empirical evidence, is that preferential attachment models do represent citation patterns quite well within individual subfields [139], but not when applied to the literature as a whole. A central parameter in the preferential attachment models is the date of the start of a subfield, and since different subfields have different start dates, the model might be expected

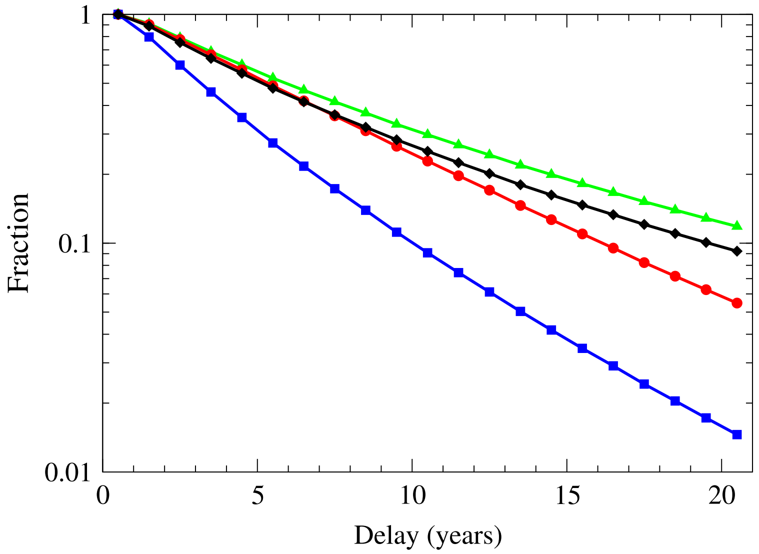


Figure 5.8: Fraction of citations made more than a given number of years after publication. Black diamonds include all citations, blue squares are self-citations, red circles are co-author citations, and green triangles are distant citations.

to work within subfields but not for the overall data set.

Figure 5.8 tests the aging of papers within the Physical Review data set by plotting the fraction of citations that are to papers a certain time in the past. Let us focus for the moment on the black curve, which includes all citations in the entire data set. The figure shows that there does indeed appear to be a strong aging effect, with the citation rate dropping off approximately exponentially over time (which would be a straight line on the semi-logarithmic scales of the plot). This finding is in agreement with previous studies of aging [193], which also found exponential decay. An alternative interpretation of the data, however, is that there is no aging occurring at all, and that the drop in citations is a purely mechanical effect that results from dilution of the literature—in a small, young field there are only a few papers to cite and hence each receives a lot of citations; in an older field there are more papers and so individual citation rates fall off. To the extent that it has been tested, the latter theory appears to agree well with available citation data and also with the prediction of the preferential attachment models [94], so at present the evidence for (or against)

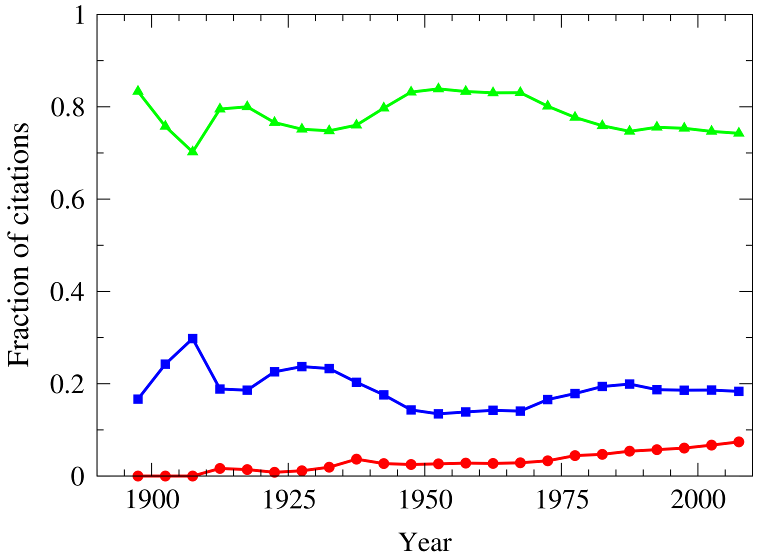


Figure 5.9: Fraction of citations made, by type, in five-year blocks. There were no citations made in the 1890–1894 block. Blue squares represent self-citations, red circles are co-author citations, and green triangles are distant citations.

aging in our data set is inconclusive.

5.3.3 Interactions between citation and coauthorship

Perhaps the most interesting aspect of the Physical Review data, however, is the window it gives us on the interplay between citation and coauthorship. One way to probe this interplay is to divide citations according to the collaborative roles assumed by the authors of the citing and cited papers and then compare the resulting citation patterns. In the present work, we divide citations into three classes, following Wallace *et al.* [178]: self-citations, where the citing and cited papers shared at least one coauthor; coauthor citations, where at least one author of the citing paper has previously collaborated with at least one author of the cited paper (but there are no common authors between papers, so that self-citations and coauthor citations are disjoint); and distant citations, which includes all citations other than self-citations and coauthor citations. (Other authors who have examined citation and collaboration have gone further and considered also citations between coauthors of coauthors [178], but this proves computationally unfeasible in the present case because of the size of the Physical Review data set.) We emphasize that we only consider individuals to be coauthors if they have *previously* coauthored when the citation occurs. Coauthorship that comes after the citation is not counted. Also our data are limited to the Physical Review, so the number of coauthor citations will in reality be higher than presented here, both because some citations are missing from our data and because some coauthorships are.

Figure 5.9 shows the fraction of citations that fall into each of the three classes as a function of the year of publication of the citing paper. Roughly speaking, the three curves appear flat over time. There is a modest increase in the fraction of coauthor citations (the lowest, red curve in the figure), but this can be explained by the increase in the number of coauthors available for citation, shown in Fig. 5.6, which is of a similar magnitude. In other respects, the rule of thumb seems to be that a constant 20% or so are self-citations, 75 or 80% are distant citations, and the small remaining fraction are to coauthors.

The distribution of time between the publication dates of a new paper and the papers it cites is shown for the three classes of citation in Fig. 5.8, as the blue, red, and green curves. Here we do notice a significant difference between the classes. In particular, the self-citations (in blue) fall off faster than coauthor and distant citations. This implies that a larger fraction of self-citations occur rapidly after publication, compared with citations in the other classes. This is not unexpected, given that a researcher presumably knows about their own research sooner, and in more detail, than they know about others'. We note also that coauthor citations are slightly earlier than distant citations, which again seems reasonable. One must be careful in the interpretation of these results, however. An alternative explanation for the same observations is that a paper can be cited by others long after the author retires or leaves the field, which could make the average delay for citations by others

Citation type	Mean delay (years)
Self-citations	4.12
Coauthor citations	6.92
Distant citations	9.02
All citations	7.89

Table 5.2: Mean time delay between a paper's publication date and the dates of the papers it cites.

Citation type	Made (%)	Received (%)
Self-citation	68.9	60.3
Coauthor citation	42.0	31.3
Both	35.6	26.3
Either	75.0	64.2
Either given both possible	76.4	66.4

Table 5.3: Percentage of papers that make or receive at least one citation of a given type.

longer than that for self-citation. There is no way to tell, purely from the delay statistics themselves, which explanation is the better one.

Table 5.2 summarizes the mean delay to citation for the three citations classes. We explore the differences between citation classes further in the next section.

5.3.4 Self-citation and coauthor citation

Consider Table 5.3, which gives the percentages of papers that make or receive at least one self-citation or coauthor citation, provided that such a citation is possible. Nearly 70% of papers cite at least one paper by the same author (or one of the same authors, if there are several), and 60% of them receive such a citation. These numbers may at first appear large, and raise concerns, given the use of citation counts as a measure of impact, that authors might be inflating their counts by self-citing [84, 15]. But taken with the fact that the number of citations per paper and the fraction which are self-citations are both sizable, these large numbers are not unexpected. Figure 5.9 shows that overall self-citation has remained constant and moderate, around 20%, and that there has been no sizable recent excess in self-citation.

A more interesting question is whether researchers have a tendency to reciprocate citations by others. If author A cites a paper of author B, does B return the favor by later citing A? To address this question we measure the fraction of citations of one author by another (excluding citations of one's own papers) that are reciprocated in one or more later publications. We calculate separate figures for pairs of authors who have previously co-authored a paper and those who have not and find that 13.5% of citations between non-coauthors are reciprocated when possible, while an impressive 43.8% of citations between coauthors are reciprocated. (Keep in mind that no authors can overlap between a citing and a cited paper for the citation to be considered a coauthor citation and not a self-citation.) Both these numbers are very high compared to the expected reciprocity if citations were made uniformly at random, but this doesn't necessarily imply a tit-for-tat return of citations. A citation is presumptively an indication that two papers fall in similar subject areas, and thus the presence of a citation greatly increases the chances that the authors are working in the same area, which in turn increases the likelihood of citation in general and therefore the chances of reciprocated citation. In the case of previous coauthors the chances of working in the same field are likely even higher. Unfortunately, we currently do not have any model of the citation process detailed enough to make a quantitative prediction of the size of this effect against which we could compare our measurements to test for significance.

5.3.5 Transitivity

In coauthorship, it has been observed that if A has coauthored a paper with B and B with C, then A and C are more likely also to have coauthored a paper. In Section 1.2.4 we define a so-called clustering coefficient that quantifies this effect, measuring the average probability that the coauthor of your coauthor is himself your coauthor [182], and such coefficients have been measured in many networks [13, 93, 45, 143]. Typically one finds that the values are significantly higher than one would expect if network connections were made purely at random, and our coauthorship network is no exception. For the data set studied here we find a clustering coefficient of 0.212, which is comparable with other figures reported for coauthorship networks [135].

In this case, however, the nature of the data set allows us to go further. The conventional explanation for high transitivity in networks relies on a triadic closure mechanism, under which two authors who share a common coauthor are more likely to collaborate in future, perhaps because they revolve in the same circles, attend the same conferences, work at the same institution, or are introduced to one another by their common acquaintance [135, 45, 87]. The present data set's time-resolved nature allows us to test this hypothesis directly. We can calculate what fraction of the time individuals who share a common coauthor but have not previously collaborated themselves later write a paper together. This is related to an independently derived measure by Opsahl [147], and similar to our nonbacktracking centrality in Chapter III in that no backtracking is allowed between the two networks. When we make this measurement for the Physical Review data we find the fraction of such author pairs to be only 0.0345—a much smaller fraction than the clustering coefficient of the whole network reported above. One reason for this small figure is that a large fraction of the transitivity seen in coauthorship networks comes from papers with three or more authors, which automatically contribute closed triads of nodes to the coauthorship network. Such triads however are excluded from our calculation of the probability of later collaboration. The large difference between the two probabilities we calculate implies that only a small fraction of the network transitivity comes from true triadic closure processes.

Nonetheless, the triadic closure process does appear to be present in our data set. Figure 5.10 shows the probability of future coauthorship between two individuals as

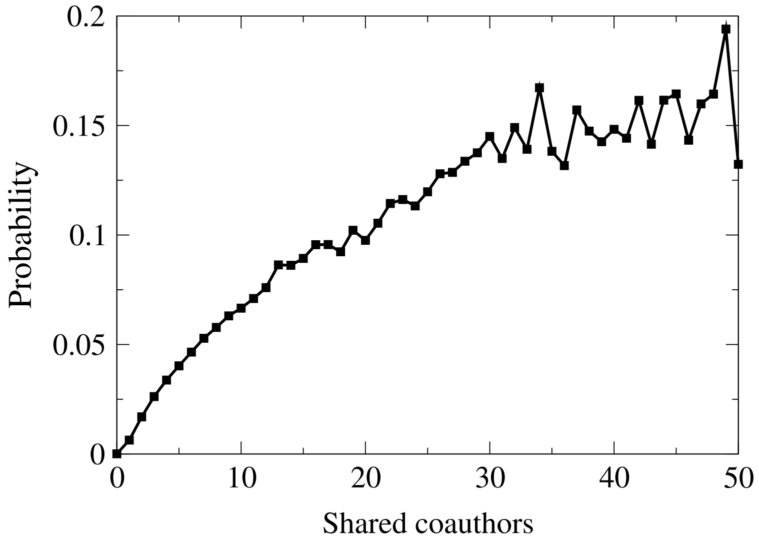


Figure 5.10: Probability of future coauthorship with another author as a function of the number of shared coauthors. The number of shared coauthors is counted at the time of first coauthorship or the date of either coauthor's last published paper, whichever comes first.

a function of their number of common coauthors, and we see that the probability increases sharply, a finding that is consistent with previous results [135, 23].

5.4 Conclusions

In this chapter I have analyzed a large data set from the Physical Review family of journals, taking a network perspective. Rather than focus solely on either citation or coauthorship networks, as most previous studies have done, I have instead combined the two, which allows the study of questions about the ways in which people—and not just papers—cite one another, and the extent to which scientists collaborate with those they cite or cite those with whom they collaborate. The time-span of the data set is unusually large, covering more than a century of publication, which allows us to study long-term changes in collaboration and citation patterns that are not accessible with smaller data sets.

My main findings are that the Physical Review appears to be growing exponentially, with a doubling rate slightly less than 12 years, and the number of citations per paper within the journals also appears to be growing. The fraction of self-citations and citations among coauthors is more or less constant over time, and authors tend to cite their own papers sooner after publication than do their coauthors, who in turn cite sooner than non-coauthors. We observe a strong tendency towards reciprocal citations, researchers who cite another author often receiving a citation in return later on, with especially high rates for citations between coauthors. Contrary to some previous claims [135, 45, 87], however, there is only a small triadic closure effect in the coauthorship patterns; two researchers who share a common coauthor but have never collaborated themselves have only a rather small probability of collaborating in future—about 3.5%. This number is nonetheless much higher than the probability for two randomly chosen researchers, and moreover increases sharply as the number of common coauthors increases.

A limitation of our analysis is that the data come from a single family of journals in a single field. There are, however, some results for other journals and fields that suggest that the observed patterns extend beyond physics and the Physical Review. In one recent study, for example, Huang *et al.* [89] examined a collection of papers in computer science drawn from the CiteSeer database of online preprints. They find, as we also do, that the number of papers and number of authors both increase roughly exponentially over time, while the number of authors per paper and number of coauthors per author increase roughly linearly. Wuchty *et al.* [186] examined a large set of papers drawn broadly from the sciences and engineering, using data from the commercial Web of Science database (formerly the Science Citation Index). They observe in particular that the average number of authors on a paper has increased steadily over time, at least for papers with more than one author, which again agrees qualitatively with our observations. Döbler [176] studied a data set representing the fields of mathematics, logic, and physics from 1800 to 1998 and found again that collaboration has increased over time, albeit intermittently, and at a rate that depends on the field.

There are many other questions that could be addressed with the data we have analyzed, the unusually long time-span and combination of publication and citation data opening up a variety of possibilities. For instance, we know which papers are published in which of the various Physical Review journals, and hence we have a crude measure of paper topic, which would allow us to answer questions about how the patterns of coauthorship and citation vary between fields within physics. We could also study geographical variations by making use of the data on authors' institutional affiliations [149]. Our analysis of long-term historical trends could also be extended; for the researcher interested in the history of US physics, there are, no doubt, many interesting signatures of historical events hidden within the data. The data set also offers the possibility of tracking the careers of individual scientists, possibly over long periods of time, or of tracking research on a particular topic. And finally, any of our analyses could be extended to data sets that cover other journals or fields other than physics, if and when such data become available. All of these would make excellent subjects for future investigation.

CHAPTER VI

Conclusions

Better network science tools deepen our ability to analyze and understand the world and increase network data collection. A profusion of network data sets, however, places increasing demands on the tools of network theory. This thesis addresses four network science issues, deepening our ability to understand network data. I explore the potential impact of modeling assumptions on network processes, and I highlight the power of assumptions about node decision making behavior in the context of network cascades. I address measures of network structure, and use spectral techniques to refine a popular network centrality construct. I improve methods of incorporating available data into network inference, specifically the usage of probabilistic information about the presence of edges. Lastly, I examine the combination of multiple network types for data analysis, specifically by analyzing joint coauthorship and citation networks. I describe these issues in detail below.

Strategic cascades In Chapter II I have shown the importance of assumptions about agent decision making in modeling and have improved our understanding of cascade scheduling on networks with strategic agents. For the specific model of Chierichetti et al. [35], I have shown that assuming strategic instead of myopic agent decision making can lead to different cascade behaviors to the largest extent that the model allows. More broadly, I illustrate methods for reasoning about strategic cascade behavior and characterize the contrasting behavior of strategic and myopic agents in a range of qualitatively distinct settings.

I consider cascades to be representative of a broader class of scenarios involving dynamic decision on networks. For these too one should expect the spectrum of behaviors, myopic to strategic, to exhibit qualitative variety in generated outcomes. This chapter helps guide the creation of better models for networks of strategic agents in the future.

Nonbacktracking centrality In Chapter III I have shown that the widely used network measure known as eigenvector centrality fails under commonly occurring conditions because of a localization transition in which most of the weight of the centrality concentrates on a small number of vertices. The phenomenon is particularly visible in networks with high-degree hubs or power-law degree distributions, which includes many important real-world examples. I propose a new spectral centrality measure based on the nonbacktracking matrix which rectifies the problem, giving values similar to the standard eigenvector centrality in cases where the latter is well behaved, but avoiding localization in cases where the standard measure fails. The new measure is found to give significant decreases in localization on both synthetic and real-world networks. Moreover, the new measure can be calculated almost as quickly as the standard one, and hence is practical for the analysis of very large networks of the kind common in recent studies. This chapter gives a better tool for ranking nodes in networks with hubs, and has already attracted interest from practitioners studying the US road network and other networks with hubs.

Uncertain networks In Chapter IV I have described methods for the analysis of networks gathered by uncertain measurements of their edges. I gave a method for performing the common task of community detection on such networks by fitting a network generative model to the data using a combination of an EM algorithm and belief propagation. I also show how the resulting fit can be used to reconstruct the true underlying network by predicting which nodes are connected by edges. Using controlled tests on computer-generated benchmark networks, I show that these methods give better results than previously used techniques that rely on simple thresholding of probabilities to turn indefinite networks into definite ones. Additionally I have given an example application of our methods to a bacterial protein interaction network taken from the STRING database [174]. The methods I present have been used by scientists studying brain networks and protein-protein interaction networks.

The methods described in this chapter could be extended to the detection of other types of structure in networks. They have attracted interest from theorists interested in extending my method to networks with correlated noise.

Coauthorship and citation In Chapter V I have analyzed a large data set from the Physical Review family of journals, taking a network perspective. Rather than focus solely on either citation or coauthorship networks, as most previous studies have done, I have instead combined the two. This combination allows the study of questions about the ways in which people—and not just papers—cite one another, and the extent to which scientists collaborate with those they cite or cite those with whom they collaborate. The timespan of the data set is unusually large, covering more than a century of publication, allowing us to study long-term changes in collaboration and citation patterns that are not accessible with smaller data sets. This chapter has contributed a new lens for viewing academic networks, and has resulted in numerous findings which increase our understanding of the physics community.

The expansion of network science continues, with endless room for better theories. The widening scope of network science calls for a unification of the fragmented studies. This is an important direction of future work—the development of fundamental network science techniques to unite the many possible avenues of network science application.

APPENDICES

APPENDIX A

Strategic cascades

A.1 Clique

Proof of Theorem 2.1. The proof of this theorem follows naturally from several lemmas which we prove in the rest of this section. We outline the proof here.

There are two cases of behavior to consider: $\pi < 1$ and $\pi \geq 1$. In the first case a simple backward induction argument shows that all cliques are in $\mathbf{C}_{PNC}$ (Lemma A.1). The second case is more involved. We first show that if $K_n \in (\mathbf{C}_{PNC} \cup \mathbf{C}_{TC})$ for some n, p, and π, then all larger cliques must also be in $(\mathbf{C}_{PNC} \cup \mathbf{C}_{TC})$ for the same p and π (Lemmas A.2 and A.3). Finally, we show that any p and π combination eventually gives a $K_n \in (\mathbf{C}_{PNC} \cup \mathbf{C}_{TC})$ for some large enough n (Lemma A.4). The intuition for this final lemma is that, for very large cliques, the cost of ending up on the wrong side of a cascade is very large. Thus, agents always prefer to join a guaranteed cascade over choosing their type *if* choosing their type has some probability of being on the wrong side of the cascade.

Let $d = m_Y - m_N$ denote the difference between the current number of Y decisions and the current number of N decisions. We begin by addressing clique behavior in the first case, $\pi < 1$.

Lemma A.1. *If $\pi < 1$ then every clique is in $\mathbf{C}_{TC}$.*

Proof. Consider the behavior of the last scheduled node. It chooses its type only if $d = 0$. If $d \neq 0$, then it receives at least 1 more utility for choosing the majority, but only $\pi < 1$ more utility for choosing its own type. Now by induction, assume that agents after time $\tau > 1$ choose their type if $d = 0$ and otherwise choose the current majority. We must show that the node α at time τ will do the same.

If $d \geq 2$ then no matter what α chooses, by the inductive hypothesis, the rest of the nodes will be in a Y-cascade. Thus α will receive at least 2 more utility for choosing the majority (Y), but only π more utility for choosing its type, so it will always choose Y.

If $d = 1$, then if α chooses Y, it will cause a Y-cascade and receive $\tau - 1$ utility for agreement with currently undecided nodes and receive 1 more utility for its agreement with currently decided nodes. If α chooses N, then with probability $1 - p$ the next node will cause an N-cascade, but with probability p the next node will cause a Y-cascade. In the former case α receives $\tau - 1$ utility for its agreement with currently undecided nodes. In the latter case, α receives 0 utility for its agreement with currently undecided nodes. α's expected payoff from agreement with currently undecided nodes for choosing N is $(1 - p)(\tau - 1)$. Without considering payoff from choosing its type, $c_\alpha = Y$ yields $1 + p(\tau - 1)$ more utility than $c_\alpha = N$. So, no matter the value of t_α, $c_\alpha = Y$.

If $d = 0$, the inductive hypothesis gives that c_α starts a cascade of α's choice. Thus by playing $c_\alpha = t_\alpha$, it gets π additional utility. The analysis for $d = -1$ and $d \leq -2$ are analogous to the cases already covered. □

For the remainder of the section we assume $\pi \geq 1$. Additionally, we refer to agents on the clique according to when they are scheduled. On K_n, we call the first scheduled node n and the last scheduled node 1. We begin by showing that, as n increases, cliques that enter into $(\mathbf{C}_{PNC} \cup \mathbf{C}_{TC})$ stay that way.

d:	-1	0	1
n	N	N	?
$n+1$	N	N	N

(a) N-type decision

d:	-1	0	1
n	N	N	?
$n+1$	N	?	?

(b) Y-type decision

Figure A.1: $\mathbf{C}_{PNC}$ behavior

Lemma A.2. *For large enough cliques such that $(n+1)(1-p)^2 > \pi$, if $K_{n-1} \in (\mathbf{C}_{PNC} \cup \mathbf{C}_{TC})$ and $K_n \in \mathbf{C}_{PNC}$ (a predetermined N-cascade), then $K_{n+1} \in (\mathbf{C}_{PNC} \cup \mathbf{C}_{TC})$.*

Proof. We prove this by cases, depending on how n behaves if $d = 1$. A diagram of behavior we know by assumption or can readily infer is shown in Figure A.1.

We characterize c_{n+1} if it is t_{n+1}, conditional on c_n when $d = 1$.

Case 1: $c_n = Y$ if $d = 1$ and $t_n = N$. Then it must be the case that $K_{n-1} \in \mathbf{C}_{TC}$, so this results in a Y-cascade. Thus $c_{n+1} = t_{n+1}$ and $K_{n+1} \in \mathbf{C}_{TC}$.

Case 2: $c_n = N$ if $d = 1$ and $t_n = N$. This corresponds to $\mathbf{C}_{PNC}$. Consider the possible payoffs. $c_{n+1} = Y$ gives a $(1-p)^2$ probability of $c_{n-1} = c_n = N$. This causes a N-cascade, by the assumption that $K_{n-1} \in (\mathbf{C}_{PNC} \cup \mathbf{C}_{TC})$, and results in only π payoff:

$$\text{Payoff for } N = n+1$$

$$\text{Payoff for } Y \leq (1-(1-p)^2)(n+1+\pi) + (1-p)^2\pi = n+1+\pi-(1-p)^2(n+1)$$

So in case 2, $c_{n+1} = N$, regardless of type, if $(n+1)(1-p)^2 > \pi$. Thus $K_{n+1} \in \mathbf{C}_{PNC}$. □

Lemma A.3. *For large enough cliques such that $(n+1)(1-p)^2 > \pi$, if $K_{n-1} \in (\mathbf{C}_{PNC} \cup \mathbf{C}_{TC})$ and $K_n \in \mathbf{C}_{TC}$ (the first agent, n, chooses t_n and starts a t_n-cascade), then $K_{n+1} \in (\mathbf{C}_{PNC} \cup \mathbf{C}_{TC})$.*

Proof. We prove this by cases, conditional on c_n and d. A diagram of behavior we know by assumption or can readily infer is shown in Figure A.2.

d:	-1	0	1
$n-1$	N	N	Y
n	N	N	?
$n+1$	N	N	?

(a) N-type decision

d:	-1	0	1
$n-1$	N	?	Y
n	?	Y	Y
$n+1$	?	?	?

(b) Y-type decision

Figure A.2: $\mathbf{C}_{TC}$ behavior

d:	-1	0	1
$n-1$	N	N	Y
n	N	N	N
$n+1$	N	N	?

(a) N-type decision

d:	-1	0	1
$n-1$	N	Y	Y
n	Y	Y	Y
$n+1$	?	?	?

(b) Y-type decision

Figure A.3: $\mathbf{C}_{TC}$, case 2 behavior

We characterize c_{n+1} if $t_{n+1} = Y$, conditional on c_n and d.

Case 1: $c_n = Y$ if $d = 1$ and $t_n = N$. Then it must be the case that $K_{n-1} \in \mathbf{C}_{TC}$, so this results in a Y-cascade. Thus $c_{n+1} = t_{n+1}$ and $K_{n+1} \in \mathbf{C}_{TC}$.

Case 2: $c_n = N$ if $d = 1$ and $t_n = N$, and $c_n = Y$ if $d = -1$ and $t_n = Y$. Then it must be the case that $K_{n-1} \in \mathbf{C}_{TC}$. The known behavior for this case is shown in Figure A.3. Since $c_n = Y$ if $d = -1$ and $t_n = Y$, $c_{n+1} = Y$ when $d = 0$ and $t_{n+1} = Y$. Thus $K_{n+1} \in \mathbf{C}_{TC}$

Case 3: $c_n = N$ if $d = 1$ and $t_n = N$, and $c_n = N$ if $d = -1$ and $t_n = Y$. Then it must be the case that $K_{n-1} \in \mathbf{C}_{TC}$. The known behavior for this case can be seen in Figure A.4. By the same math as in case 2 of the proof of Lemma A.2, we get $c_{n+1} = N$, regardless of type, if $(n+1)(1-p)^2 > \pi$. Thus $K_{n+1} \in \mathbf{C}_{PNC}$.

□

d:	-1	0	1
$n-1$	N	N	Y
n	N	N	N
$n+1$	N	N	?

(a) N-type decision

d:	-1	0	1
$n-1$	N	Y	Y
n	N	Y	Y
$n+1$	?	?	?

(b) Y-type decision

Figure A.4: $\mathbf{C}_{TC}$, case 3 behavior

Next, we show that a clique eventually falls into $\mathbf{C}_{PNC}$ or $\mathbf{C}_{TC}$.

Lemma A.4. *For any p and π, there exists some large enough n such that $K_n, K_{n+1} \in (\mathbf{C}_{PNC} \cup \mathbf{C}_{TC})$.*

Proof. Fix p and π. By backward induction one can see that there is always an N-cascade when $d = \lfloor -\pi \rfloor$. Call this threshold $\underline{d}$. Additionally, note that an N-type node always chooses N when $d \leq 0$. This can be seen by application of Theorem 2.17 and Lemma A.5.

So, for large enough n, there must be an N-cascade when $d = \underline{d} + 1$. This can be seen by application of Lemma A.6 and the observations that $d = \underline{d}$ gives a certain N-cascade but $d = \underline{d} + 2$ does not give a certain Y-cascade.

This argument can be repeated to show that there must be an N-cascade when $d = \underline{d} + 2$. This reasoning can be iterated until it breaks down at $d = 0$. But the node choosing at $d = 0$ always has the option of a certain N-cascade (for large enough n). So it will choose Y only if it also is faced with the option of a certain Y-cascade. In this case, a node chooses its type t and starts a cascade of that type. By the same reasoning the next node $n + 1$ also either chooses N or its type and starts a cascade. □

Lemma A.5. *Decisions are monotonic in d. If $c_n = Y$ for some difference d and type t, then $c_n = Y$ for difference $d + 1$ and the same type. Similarly, if $c_n = N$ for some difference d and type t, then $c_n = N$ for difference $d - 1$ and the same type.*

Proof. Assume, without loss of generality, that agent n is scheduled to make a decision and $d > 0$, $t_n = N$. Let $q_Y = 1$ be the probability of a Y-cascade if $c_n = Y$ and $q_N < 1$ be the probability of an N-cascade if $c_n = N$. Also assume that, if an N-cascade does not occur, then a Y-cascade occurs and n gets only some constant payoff D. Then

k	d:	-2	-1	0	1	2
0	Choice:	N	S	S	S	Y
1	Choice:	N	N	S	Y	Y
	E[N matches]:	3	2	$1-p$	$1-p$	$1-p$
	E[Y matches]:	p	p	p	2	3
2	Choice:	N	N	S	Y	Y
	E[N matches]:	4	3	2	$1-p$	0
	E[Y matches]:	0	p	2	3	4

Figure A.5: The choice of a node on the clique for varying d, k, and t

consider n's payoffs:

$$\text{Payoff for } N = q_N n + (1 - q_N)D + \pi$$

$$\text{Payoff for } Y = n$$

For large enough n, the payoff for Y will always surpass the payoff for N. □

Lemma A.6. *On a large enough clique K_n, a node always prefers a certain (probability 1) cascade over an uncertain (probability less than 1) cascade, no matter d, p, or π.*

These two lemmas, combined with Theorem 2.17, are used to show that the range of differences for which nodes consider choosing their type shrinks as n grows larger. Eventually the range shrinks enough that nodes have the option of a guaranteed cascade, completing the proof of Theorem 2.1. □

Proof of Theorem 2.2. We consider strategic users on a clique of size > 1 with $1 \leq \pi < 1 + p$. We show, using backward induction, that the first agent to choose always selects its type and that all following agents select the same choice. A demonstration of the backward induction can be found in Figure A.5.

Figure A.5 displays the choice a node would make if there were k undecided agents and the difference between choices already made, $d = m_Y - m_N$, is given in the top row. N (resp. Y) means that both agents choose N (resp. Y) no matter their type. S means that the choices are Split: agents choose their type. The rows below "Choice" display the expected number of matches from other agents if an agent chooses N or Y.

To obtain the behavior observed for $k = 1$, we need $(1-p)+\pi < 2$ and $p+\pi < 2$, or, rearranging: $\pi < 1 + p$ and $\pi < 1 + (1 - p)$. Since $p < (1 - p)$, this is satisfied with any $\pi < 1 + p$.

The behavior for $k = 2$ follows directly from the behavior observed for $k = 1$, as all inequalities are only made looser. The behavior for $k > 2$ follows inductively from the behavior for $k \leq 2$. Assuming an agent with $k - 1$ remaining undecided agents chooses its type only when $d = 0$, an agent with k remaining undecided agents will behave similarly. The inequalities for this decision are identical to the inequalities for $k = 2$ with a multiplier of k on each side. Our table shows that, once the balance of choices shifts in one direction, it is in a node's best interest to choose the same way. Thus the first node to choose starts a cascade of the type that it selects. □

Theorem 2.2 has the following immediate corollary:

Corollary A.7. *For any clique graph with any* $1 \leq \pi < 1 + p$, *in the limit of* n, *the optimal strategic performance is* $1 + \frac{1}{p}(1 - p)(1 - 2p)$ *times the optimal myopic performance.*

Proof. This follows from the fact that two Y decisions are required to start a myopic Y-cascade and only one is required to start a strategic Y-cascade. Thus in the limit of n, the probability of a Y-cascade is $\frac{p^2}{p^2+(1-p)^2} \approx p^2$ for myopic agents, a well-known bound from the "gamblers ruin" problem, but is p for strategic agents. The result immediately follows. □

k	d:	-3	-2	-1	0	1	2	3
0	Choice	S	S	S	S	S	Y	Y
1	Choice	N	S	S	S	S	Y	Y
	E[N matches]	4	$3-p$	$2-p$	$1-p$	$1-p$	$1-p$	$\dots$
	E[Y matches]	$1+p$	$1+p$	$1+p$	$1+p$	3	4	5
2	Choice	$\dots$	N	$\dots$	S	$\dots$	Y	$\dots$
	E[N matches]	$\dots$	4	$\dots$	$2-2p$	$\dots$	$(1-p)(2-p)$	$\dots$
	E[Y matches]	$\dots$	$1+2p$	$\dots$	$\dots$	$\dots$	5	$\dots$
3	Choice	$\dots$	$\dots$	N	$\dots$	S	$\dots$	$\dots$
	E[N matches]	$\dots$	$\dots$	4	$\dots$	$3-O(p)$	$\dots$	$\dots$
	E[Y matches]	$\dots$	$\dots$	$1+O(p)$	$\dots$	5	$\dots$	$\dots$
4	Choice	$\dots$	$\dots$	$\dots$	N	$\dots$	$\dots$	$\dots$
	E[N matches]	$\dots$	$\dots$	$\dots$	4	$\dots$	$\dots$	$\dots$
	E[Y matches]	$\dots$	$\dots$	$\dots$	$1+O(p)$	$\dots$	$\dots$	$\dots$

Table A.1: Utilities for node decisions on a clique with one external neighbor choosing Y.

A.2 Council and cloud

Proof of Lemma 2.5. We prove this lemma by directly comparing expected utilities from Table A.1. We explicitly list utilities for direct comparison, when relevant and non-obvious. All utilities assume there is an external neighbor choosing Y with probability 1. Here $d = m_Y - m_N$ and k is the number of currently undecided agents in the clique.

We can see from Table A.1 that, for small enough p, and the appropriate $2 < \pi < 3$, the first scheduled node in the clique will choose N. We can also see that the table continues for cliques larger than size 6, as payoffs are shifting in favor of N as the clique grows. □

Proof of Lemma 2.9. If $a(1-p) + \pi < bp$, then agent 2 will always match the choice of 3. The case in which 2's payoff is greatest for not matching 3 (and thus the lemma is hardest to satisfy) is $t_2 = N$, $c_1 = N$, and $t_3 = Y$. In this case 2 will obtain payoff $a + \pi + b(1-p)$ for choosing N, or payoff $ap + b$ for choosing Y. A comparison of these payoffs shows that 2 will choose Y when $a(1-p) + \pi < bp$, which is true by assumption. So 2 will choose Y, matching with 3. All other combinations of types result only in a higher payoff to 2 for matching with 3. □

Proof of Lemma 2.10. Note that 3 will always match 2 because $b > b(1-p) + \pi > bp + \pi$, by assumption (these inequalities simplify to $bp > \pi$). In the worst case, 1 has chosen Y and $t_2 = N$. In this case, 2 will obtain utility of $a + b$ for choosing Y and utility of $a(1-p) + \pi + b$ for choosing N. Thus 2 will choose Y if $ap > \pi$, which we assume to be true. □

A.3 Omitted miscellaneous results

Proof of Theorem 2.12. This theorem follows from the fact that the cascade scheduling problem can be expressed as a finite extensive form game with perfect information. When nodes are never indifferent between choices, the unique PBE can be constructed using backward induction, following Mas-Colell et al. [121, Prop. 9.B.2]. When nodes resolve their indifference in a consistent way, such as choosing their type, the same backward induction still selects a unique PBE. □

Proof of Theorem 2.13. Let I be the set containing all agent-situation pairs (i, R) where an agent is indifferent between its two choices. (i, R) means that the situation is R and the next agent to choose is i. Let $\overline{I}$ be the set of agent-situation pairs where agents are not indifferent. $\overline{I}$ is finite, so there must be some pair $(i^*, R^*) \in \overline{I}$ where i^* has minimal difference between utility for Y and N. Denote this difference δ, and let $\epsilon = |\frac{\delta}{2}|$. Then no other agent-situation pair in $\overline{I}$ results in a different decision in F', assuming behavior of i for all $(i, R) \in I$ remains the same. Under F, all agents in all situations in I have equal utility for both choices but choose their type, by assumption. Under F', all agents in all situations in I have ϵ greater utility for their type, thus make the same decision as in F. □

The following Lemma shows that a Y-type node will always choose Y if an N-

type node would have chosen Y in a similar situation and will be used in the proof of Theorem 2.14

Lemma A.8. *Let i_Y be an agent in situation R_Y and i_N be an agent in situation R_N, with $t_{i_Y} = Y$ and $t_{i_N} = N$. If R_Y and R_N are identical except for some N decisions in R_N may be Y decisions in R_Y, and the nonadaptive schedule S is the same for both nodes, then it is never the case that $c_{i_N} = Y$ but $c_{i_Y} = N$.*

Proof of Lemma A.8. Let us compare the utilities from choosing Y for i_Y and i_N, and assume for the sake of contradiction that i_N prefers Y but i_Y does not. Then i_Y's utility must be greater for choosing N, and i_N's utility must be greater for choosing Y. Below we let $\#Y(U)$, $\#N(U)$ be the expected number of Y, N decisions, respectively, in the set of agents $U \subseteq V$ at the end of the game. The expected value E is over the randomness of agent types.

Utility comparison for i_Y:

$$\pi + E(\#Y(nb(i_Y)) \mid c_{i_Y} = Y) < E(\#N(nb(i_Y)) \mid c_{i_Y} = N).$$

Utility comparison for i_N:

$$E(\#Y(nb(i_N)) \mid c_{i_N} = Y) > \pi + E(\#N(nb(i_N)) \mid c_{i_N} = N).$$

Between these two inequalities only π has moved, and the conditions of R_Y favor the left side of the first inequality. Facing the same schedule, both inequalities cannot be true, thus we have reached a contradiction. □

Proof of Theorem 2.14. Generate the type distribution for F' in the following way. Independently draw $n = |V|$ types from $\{Y, N\}$, selecting Y with probability p and N with probability $1 - p$. This gives us a vector of F's *base types*, $\mathbf{t} = (t_1, \ldots, t_n)$. Next generate a vector of F's *true types*, $\mathbf{t}' = (t'_1, \ldots, t'_n)$, by switching N-types to

Y-types with probability $(p' - p)/(1 - p)$.

$$t'_i = \begin{cases} Y & : t_i = Y \\ Y\text{, with probability } \frac{p'-p}{1-p} & : t_i = N \\ N & : \text{otherwise} \end{cases}$$

This results in each agent in F' being Y-type with independent probability p', as desired. However, now each random draw of base types for F', $\mathbf{t}$, can be coupled with a corresponding draw of types for F, $\mathbf{s}$. The true types of F', $\mathbf{t}'$, have Ys in the same places as $\mathbf{s}$ but with some additional Ns turned to Ys. By Lemma A.8 one can see that $\mathbf{t}'$ results in at least as many Y choices. □

Proof of Lemma 2.20. First, we note that $\hat{c}_i^p$ is monotone in its inputs and in p by Lemma A.9.

In the base case, we have that $c_1^p(t_1) = \mathring{c}_1^p(t_1) = \hat{c}_1^p(t_1)$ which is monotone in the inputs and in p because $\hat{c}_1^p$ is.

Assume that the statement is true for all $j < i$. Note that $c_i^p(t^{(i)}) = \hat{c}_i^p(\mathring{c}_{i-1}^p(t^{(i-1)}), t_i)$. Thus $\hat{c}_i^p$ is monotone in inputs and p because we know $\hat{c}_i^p$ is and $\mathring{c}_{i-1}^p$ is by induction. Also $\mathring{c}_i^p(t^{(i)}) = \mathring{c}_{i-1}^p(t^{(i-1)}) \circ \hat{c}_i^p(\mathring{c}_{i-1}^p(t^{(i-1)}), t_i)$. So $\mathring{c}_i^p$ is monotone in inputs and p because we know $\hat{c}_i^p$ is and $\mathring{c}_{i-1}^p$ is by induction. □

Lemma A.9. *Let R_F be a situation in game $F = (G, p, \pi)$ and $R_{F'}$ be a situation in game $F' = (G, p', \pi)$ with $0 < p < p' < .5$. Let R_F and $R_{F'}$ have the same nonadaptive schedule and let R_F and $R_{F'}$ be identical except for some N decisions in R_F may be Y decisions in $R_{F'}$. If the next scheduled agents in both games are the same type, and if the agent chooses Y in game F, then the agent in game F' also chooses Y.*

Proof of Lemma A.9. The proof of this lemma proceeds by a coupling of the unscheduled agents of F and F'. First, each individual agent in F' is at least as likely to be Y-type as a corresponding agent in F. By Lemma A.8, each individual agent is more likely to choose Y in F' as in F, and thus the current agent will only choose Y in F' if it would in F. □

A.4 Omitted star results

Section A.4 is devoted to proving Theorem 2.15, which states that performance is always greater with myopic agents than strategic agents on the star, no matter the situation. A sketch is as follows.

We break the proof into two cases depending on the value of π. We first handle the case where $\pi \geq 1$ in Theorem A.10. We next consider the case where $\pi < 1$ and the scheduler is limited to nonadaptive schedules in Theorem A.12. The final case, where $\pi < 1$ with an adaptive schedule, is more involved. We first give a schedule, S_{opt}, and show that it is weakly optimal in both the strategic and myopic settings in Lemma A.14. We prove optimality by showing that scheduling the interior node before a Y majority has been reached is never a better option. Lastly, we show higher myopic performance under this optimal adaptive schedule by detailing the behavior of agents under this schedule.

We begin by analyzing the case $\pi \geq 1$.

Theorem A.10. *For any star graph with $\pi \geq 1$, for* any *schedule, the myopic performance is greater than the strategic performance.*

Proof. Game behavior is simple when $\pi \geq 1$: every exterior agent, strategic or myopic, chooses its type. It is only the interior agent's behavior that *might* differ.

For any fixed draw of agent types $\mathbf{t} = (t_1, \ldots, t_n)$ with schedule S, myopic and strategic agents behave identically except for the interior agent i. Thus, for any fixed $\mathbf{t}$, the situations in which S selects i to decide next for myopic agents are the same as for strategic agents. A strategic i reasons that each of its undecided exterior neighbors will choose Y with probability $p < .5$, and therefore expects more of its undecided neighbors to choose N than Y. A myopic interior agent ignores this fact and chooses as if equal numbers of undecided neighbors will choose N and Y. Thus a strategic interior agent is less likely to choose Y because it shifts its expected payoff in favor of N. □

Next, we consider the case $\pi < 1$. We first give a cursory examination of agent behavior. A myopic exterior agent chooses its type if the interior agent has not yet chosen. If the interior agent has chosen c and an exterior agent (strategic or myopic) is scheduled to decide, it will also choose c because $\pi < 1$ guarantees that an agent prefers a matching choice over choosing its type. Knowing this, a strategic interior agent always chooses the majority choice of the already decided agents and breaks ties with its type. A myopic interior agent behaves the same way.

We summarize the behavior of myopic exterior agents in a formal theorem, for reference below.

Theorem A.11. *For any star graph with $\pi < 1$, myopic exterior agents choose their type if scheduled before the interior node and match the choice of the interior node if scheduled after it.*

Strategic and myopic agents differ only in the behavior of exterior agents scheduled before the interior agent has decided. Myopic agents choose their type, as noted above, but strategic agents choose based on what they expect the interior agent to choose. A strategic Y-type exterior agent might choose N if it sees that many exterior agents have already chosen N, and thus that the interior agent is likely to choose N. This assessment, however, depends on the schedule.

We first address the case where the scheduler is limited to nonadaptive schedules.

Theorem A.12. *For any star graph with $\pi < 1$, an optimal nonadaptive schedule selects the interior agent first for both strategic and myopic agents.*

Proof. If the interior agent is scheduled first, it will choose its type and the exterior agents will follow. This happens for both myopic and strategic agents, and guarantees a performance of p. By Theorem 2.17 this is the best possible performance for a nonadaptive schedule. Thus choosing the interior agent first is an optimal nonadaptive schedule and, in this case, the strategic and myopic performance is identical. □

We next define the adaptive schedule S_{opt} and prove its optimality for both strategic and myopic agents.

Definition A.13. Schedule S_{opt} is the following:

1. Schedule exterior agents until a majority decide Y or all have decided, whichever comes first.

2. Schedule the interior agent.

3. Schedule all remaining exterior agents.

The optimality of S_{opt} is summarized in the lemma below.

Lemma A.14. *For any star graph with $\pi < 1$, S_{opt} is a weakly optimal adaptive schedule for both strategic and myopic agents.*

Proof. The combination of Theorems A.15 and A.16 shows that S_{opt} is weakly optimal for myopic and strategic agents (establishing Lemma A.14). It is only weakly optimal: for some parameter settings other schedules give equal performance. For example, when π is sufficiently small, a strategic exterior Y-type node will choose N when $d = -1$, and thus scheduling the interior node first yields equal performance. □

Theorem A.15. *For both strategic and myopic agents on the star graph, S_{opt} gives no worse performance than scheduling the interior agent first.*

Proof. For both strategic and myopic agents, scheduling the interior agent first guarantees all exterior agents will match its decision. So all agents choose Y with probability p and N with probability $1 - p$. This gives pn performance.

S_{opt} calls for scheduling an exterior agent, e, first. With probability p, $t_e = Y$. If $t_e = Y$ and $c_e = Y$, then S_{opt} schedules the interior agent next and all agents choose Y. This gives e utility $\pi + 1$, its maximum possible utility. So strategic e always chooses Y if $t_e = Y$. And myopic e will choose its type, so will also choose Y if $t_e = Y$. This alone gives pn performance, without considering outcomes for $t_e = N$. □

Theorem A.16. *For any star graph, while exterior Y-decisions are* not in the majority*, scheduling an exterior agent results in performance at least as high as scheduling the interior agent.*

Proof. Let $d = m_Y - m_N$ denote the difference between the current number of Y decisions and the current number of N decisions. First consider the case $d < 0$. If scheduled, the interior node will choose N and all remaining exterior nodes will choose N. This results in zero additional Y-adoptions, so scheduling an exterior node instead, as S_{opt} does, must be at least as good.

The other possibility is $d = 0$, which reduces to the situation covered in the proof of Theorem A.15. □

Now that we've shown S_{opt} is an optimal adaptive schedule for strategic agents, we give a detailed characterization of agent behavior under S_{opt} and show that this behavior cannot lead to higher strategic performance than myopic performance.

The probability of ever getting a majority of exterior Y-adoptions, in the limit of large n, is $\frac{p}{1-p}$, a well known result in the mathematics of biased random walks.

We seek to show that strategic agents choose N in any situation where myopic agents would choose N, and thus that strategic performance is lower than myopic performance. By Theorem A.11 and Lemma A.14, it is sufficient to show that, under S_{opt}, N-type exterior nodes that are scheduled before the interior node always choose N. To prove this we start by characterizing these agents' strategies.

d:	$-\infty$	$\dots$	$Y^*(k)$	$\dots$	$N^*(k)$	$\dots$	1
$t_e = N$	N	N	N	N	N	Y	Y
$t_e = Y$	N	N	Y	Y	Y	Y	Y

Figure A.6: Threshold behavior for exterior nodes, by type.

For the remainder of the section, we omit the word "exterior" when it is clear from context and refer to the interior node as i. For a given situation, let d be the difference between the number of exterior nodes who have chosen Y and the number who have chosen N: $d = m_Y - m_N$. A low d value indicates that more nodes have chosen N and indicates a higher likelihood of $c_i = N$. Thus nodes are more inclined to choose N for low values of d and more inclined to choose Y for high values of d.

Let $i(d, k)$ denote the probability that node i will choose Y, as assessed from the perspective of an exterior node in a situation with difference d and k unscheduled exterior nodes. Let $u(t, c, d, k)$ denote the expected utility of a type t node, making choice c, with a difference of d when there are k unscheduled exterior nodes. We begin counting at 1, so $k = 1$ refers to the choice of the final unscheduled exterior node.

We say that exterior nodes execute a *threshold strategy* if there exists thresholds $Y^*(k)$ and $N^*(k)$ such that:

- $d < Y^*(k)$: all agents choose N when k nodes remain to choose.
- $Y^*(k) \leq d \leq N^*(k)$: agents choose their type when k nodes remain to choose.
- $N^*(k) < d$: all agents choose Y when k nodes remain to choose.

An illustration of these thresholds can be seen in Figure A.6.

We start out with the following theorem, which shows that the interior agent is always more likely to choose Y if scheduled immediately instead of after one additional exterior node. This theorem serves as the base case for inductive arguments in Theorem A.18 and Lemma A.19, defined below.

Theorem A.17. *$i(d,1) \geq i(d,0)$ for all d, and both are monotone increasing in d. Moreover, the final exterior agent and the interior agent (if scheduled last) play threshold strategies where $N^*(1) = N^*(0) = 1$.*

Proof. Behavior of exterior agents, and thus the value of $i(d,k)$, falls into two classes: $\pi + p < 1$ and $\pi + p \geq 1$. Our proof proceeds by finding $i(d,0)$ (the probability that the interior node chooses Y when scheduled immediately) in both cases, and then examining $i(d,1)$ (the probability that the interior node chooses Y when scheduled after one additional exterior node) for each case individually.

In either case, the interior node chooses its type when it is scheduled only if $d = 0$. It will choose Y if $d > 0$ and N if $d < 0$. Thus:

$$i(d,0) = \begin{cases} 0 & d \leq -1 \\ p & d = 0 \\ 1 & d \geq 1 \end{cases}$$

In the case that $\pi + p < 1$, $i(d,1) = i(d,0)$. We analyze $i(d,1)$ by considering the behavior of the exterior node scheduled immediately before the interior node. We call this final exterior node e. If $d = 0$ and $k = 1$, the interior node will match c_e. Knowing this, e should choose its type if $d = 0$ and $k = 1$.

If $d = -1$ then e should always choose N. If $t_e = N$ this will guarantee a payoff of $1 + \pi$, the maximum possible. If $t_e = Y$, then selecting N yields payoff 1. Selecting Y yields $1 + \pi$ if the interior node is Y-type, but only π if the interior node is N-type. This gives expected payoff of $p + \pi < 1$.

In the case that $\pi + p \geq 1$, similar analysis shows that:

$$i(d,1) = \begin{cases} 0 & d \leq -2 \\ p^2 & d = -1 \\ p & d = 0 \\ 1 & d \geq 1 \end{cases}$$

The theorem follows. □

We use the below theorem to show that nodes always behave according to a threshold strategy.

Theorem A.18. *For any star graph with $\pi < 1$, under schedule S_{opt}, exterior nodes scheduled before the interior agent execute a threshold strategy.*

Proof. The proof proceeds by induction. At $k = 0$, by Theorem A.17, the interior agent plays a threshold strategy. Assuming that after some node e all agents play a threshold strategy, we show that e does also.

Assume without loss of generality that $t_e = Y$. Assume, for the sake of contradiction, that e plays a non-threshold strategy. This necessitates $c_e = Y$ for some d but $c_e = N$ for some $d + 1$. In this case, e changes its choice to N if an additional node has chosen Y. However, because all later scheduled nodes play according to threshold strategies, an additional node choosing Y only increases the chance that the interior node chooses Y, so e's strategy is not rational. This gives a contradiction, as desired. □

Knowing that nodes behave according to a threshold strategy, we can prove the following lemma, which is enough to complete the proof of Theorem 2.15, as discussed above.

Lemma A.19. *For any star graph with $\pi < 1$, under schedule S_{opt}, exterior nodes scheduled before the interior always choose N if they are N-type.*

Proof. Lemma A.19 The proof of this lemma proceeds by induction on a slightly stronger statement. By induction on k we show that $N^*(k) = 0$ (which implies the lemma), and that $Y^*(k)$ is monotone decreasing in k. The base case is covered by Theorem A.17.

We now prove the inductive step. Let k denote the the number of unscheduled exterior nodes, and assume that for all $k < \ell$, $N^*(k) = 0$ and $Y^*(k)$ is monotone decreasing in k. We now prove the statement for $k = \ell$.

Note that $u(Y, N, Y^*(\ell - 1), \ell) = 1 = u(Y, N, Y^*(\ell - 1), \ell - 1)$. By induction, we know that $Y^*(k)$ is monotone decreasing for $k < \ell$, so for any $d < Y^*(\ell - 1)$ we are in an N-cascade situation. In this case agent ℓ receives a payoff of 1 for choosing N because it will guarantee a match with the interior agent, but does not pick its own type.

Also note that $u(Y, Y, Y^*(\ell - 1), \ell) \geq u(Y, Y, Y^*(\ell - 1), \ell - 1)$. This follows from a coupling argument between two situations which we define. In situation 1, $d = Y^*(\ell - 1) + 1$ and there are $\ell - 1$ unscheduled exterior nodes. In situation 2, we have that $d = Y^*(\ell - 1) + 1$ and there are $\ell - 2$ unscheduled exterior nodes.

Couple the randomness so that the type of the agent scheduled with k unscheduled exterior agents in situation 1 is the same as the type of the agent scheduled with $k - 1$ unscheduled exterior agents in situation 2. Note that the randomness of the final unscheduled exterior agent has not yet been fixed in schedule 1. Then by monotonicity of $Y^*(k)$ established by induction, whenever d decreases in situation 1, it also decreases in situation 2 (though the converse is not necessarily true). The Y surplus d of situation 1 with 1 unscheduled exterior node ($k = 1$), is at least the Y surplus d of situation 1 with 0 unscheduled exterior nodes ($k = 0$). By Theorem A.17 we have that the probability that the interior agent chooses Y in situation 1 is at least the probability that the interior agent chooses Y in situation 2.

Finally, note that $u(Y, Y, Y^*(\ell - 1), \ell - 1) \geq u(Y, N, Y^*(\ell - 1), \ell - 1)$ because, by the definition of $Y^*(\ell-1)$ in this situation, Y-type agents choose Y.

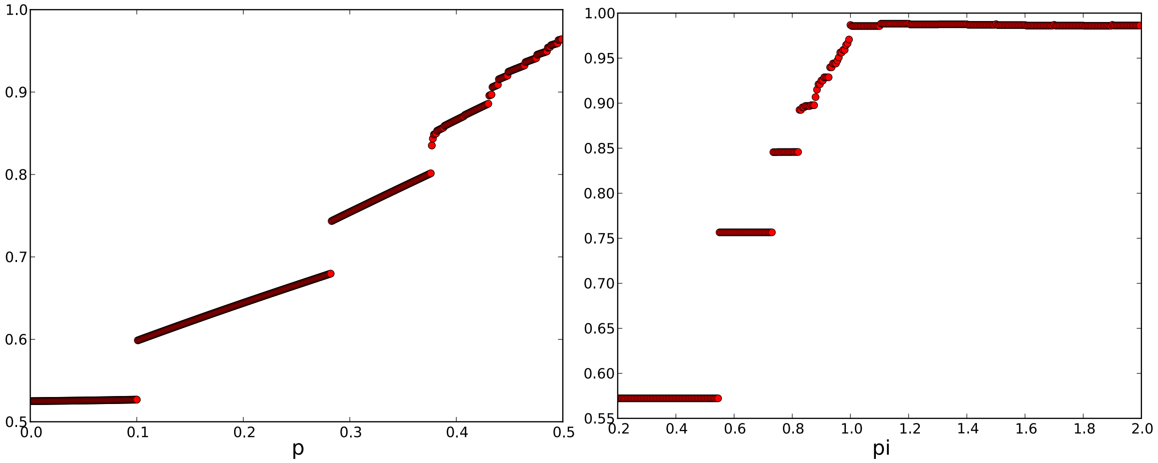
Putting this together we get that $u(Y,Y,Y^*(\ell-1),\ell) \geq u(Y,Y,Y^*(\ell-1),\ell-1) \geq u(Y,N,Y^*(\ell-1),\ell-1) = 1 = u(Y,N,Y^*(\ell-1),\ell-1)$. This shows that at $Y^*(\ell-1)$ with k nodes left unscheduled, a Y-type node chooses Y. Thus we have that $Y^*(k)$ is monotonically decreasing.

Lastly, we show that $N^*(\ell) = 0$. This is equivalent to proving that a rational N-type agent would choose N over Y, or that $u(N,N,0,\ell) \geq u(N,Y,0,\ell)$, which can also be written $(1 - i(-1,\ell-1)) + \pi \geq 1$. Note that by the definition of $Y^*(\ell)$ we have that $i(Y^*(\ell)+1,\ell-1)+\pi \geq 1$. But by the Lemma A.20 we see that $i(Y^*(\ell)+1,\ell-1) \leq (1-i(-1,\ell-1))$, and so $(1-i(-1,\ell-1))+\pi \geq 1$, as desired. □

This lemma relies on the following theorem, which allows us to compare the probabilities of two random walks trying to reach opposite endpoints, or thresholds, by traveling similar distances. Thus, whatever the thresholds $Y^*(k)$ and $N^*(k)$ end up being, the N-type node at $N^*(k)$ has a higher chance of ending up in an N-cascade than the Y-type node at $Y^*(k)$. By applying the following theorem, Lemma A.19 shows that $N^*(k) = 0$ is always a valid threshold.

Theorem A.20. *$i(Y^*(\ell)+1,\ell-1) \leq 1 - i(-1,\ell-1)$.*

Proof. The proof follows from a coupling argument and some case analysis. Consider two situations on the star with $\ell-1$ undecided nodes: situation 1 with $d_1 = Y^*(\ell)+1$ and, situation 2 with $d_2 = -1$. Couple the randomness so that whenever a node scheduled in situation 1 is Y-type the corresponding node in situation 2 is N-type. It follows that at any later step either a) situation 1 has reached a Y-cascade, or b) $d_2 \leq Y^*(\ell) - d_1$. We show if a) is not satisfied then b) is. When a) is not satisfied, situation 1 is not in a Y-cascade and agents choose their type. In this case, d_1 can increase only if the currently choosing agent in situation 1 is Y-type. However, whenever this happens the agent in situation 2 is N-type and so d_2 decreases. This



(a) Varying p, $\pi = 0.9$

(b) Varying π, $p = 0.45$

Figure A.7: Strategic-to-myopic performance ratio on star, n=21.

preserves the truth of b).

The interior node in situation 1 chooses Y only if 1) a Y-cascade is reached, or $d_1 = 0$ when there are no remaining exterior nodes ($k = 0$) and $t_i = Y$. If situation 1 enters a Y-cascade, then situation 2 enters an N-cascade because the former happens only if d_1 ever reaches 1, but then $d_2 \leq Y^*(\ell) + 1$ so that situation 2 is an N-cascade.

If $d_1 = 0$ when there are no unscheduled exterior nodes remaining and $t_i = Y$ in situation 1, then in situation 2, we have that $d_2 \leq 0$ and, by coupling, that $t_i = N$, so that the interior node always chooses N.

We have shown than whenever the interior node chooses Y in situation 1, the interior node chooses N in situation 2. □

Computational star performance Figure A.7 graphs values of the strategic-to-myopic performance ratio for a star of 21 agents to show asymptotic behavior, and shows that the ratio approaches 1 but never exceeds it, as per Theorem 2.15.

APPENDIX B

Coauthorship and citation

B.1 Data processing

As mentioned in the main text, we performed some pre-processing on the raw Physical Review data to disambiguate author names and remove extreme outliers. This appendix describes the steps taken.

B.1.1 Author name disambiguation

The data were supplied in two blocks: (1) a list of papers with associated information, such as authors, author affiliation, journal, and year of publication; (2) a list of citations, using unique paper identifiers that correspond to entries in the first block. There are, however, no unique identifiers for authors that are consistent between papers, making unambiguous author identification difficult. Not all authors use the same form for their name on every publication, and there are many examples of distinct researchers with the same name. Before using the data set, therefore, we made an effort to associate names of authors with unique people. As in previous work on author disambiguation, our process starts by assuming every name on every paper to represent a different individual [156], then computes a number of measures of author similarity and assumes authors who are sufficiently similar by these measures to be the same person. After completing this disambiguation process we checked a subset of the results by hand to estimate error rates for the process and found that it performs well. Details are as follows.

Our approach relies not only on the author names themselves to establish similarity, but also on collaboration patterns and institutional affiliation, since authors with similar names who have many of the same collaborators or who are at the same institution are more likely to be the same person. Affiliation information, however, like the author names themselves, tends to be ambiguous and inconsistent, so our first step is to combine affiliations that are deemed similar enough. We measure similarity using a variant of edit distance applied to the affiliation text strings, implemented using the Python difflib library.

Once the affiliations are processed in this way, we process the author names as follows:

1. We combine all authors with identical names who share an institutional affiliation. It appears to be uncommon for two physicists at the same university to publish under identical names, so this seems to be a safe step.

2. We find author pairs with similar but not identical names. Our criterion for similarity at this stage is that authors should have identical last names and compatible first/middle names (i.e., identical if fully written out, or compatible initials where initials are used). Also authors should not have published together on the same paper (which rules out, for example, family members with similar names who publish together). For all pairs with similar names we then calculate a further similarity measure based on how many affiliations they share, how many coauthors they share, whether their full names are identical, and whether they have published in the same journal. Authors with a high enough similarity are combined, most similar pairs first.

We have tested the accuracy of this process by drawing two lists at random from its output, the first containing 79 instances in which authors with similar names have been combined into a single author, and the second containing 111 instances in which they have not. We then performed, by hand, a blind search—without knowing the choice the algorithm has made—for publicly available on-line information about the names in question, to determine whether they do indeed represent the same or distinct researchers. We find the false positive rate to be 3% (i.e., 3% of pairs are incorrectly judged to be the same person when in reality they are distinct) and the false negative rate to be 12%.

We also tested the effect on our results of the disambiguation process by calculating a number of the statistics reported in this chapter both for the disambiguated data and for the raw data set before disambiguation, in which we naively assume that every unique author string represents a unique author and every pair of authors with the same string are the same person. We found substantial differences between the two in some of the most basic statistics, such as total number of distinct authors: the number was 328 938 in the raw data set, but fell to 235 533 after disambiguation. On the other hand some other statistics changed very little, indicating that these are not particularly sensitive to details of author identification. For example, the clustering coefficient changes from 0.222 in the raw data set to 0.212 in the disambiguated data set.

B.1.2 Data culling

In addition to author disambiguation we cull the data according to a few simple rules. There are a number of papers in the data set that have no authors listed, primarily editorials and other logistical articles without scientific content. These we remove entirely. As mentioned in the text, we also identify all papers with fifty or more coauthors, and many of our calculations are performed in two versions, with

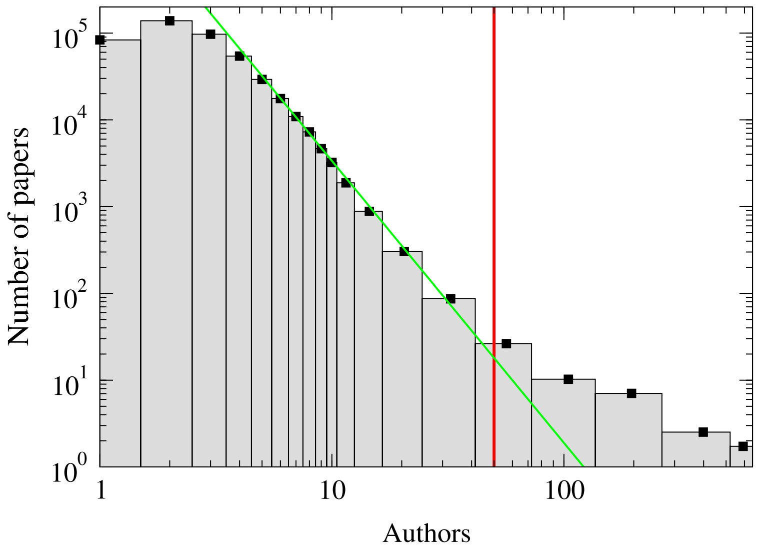


Figure B.1: Histogram of the number of papers with a given number of authors. The vertical line falls at fifty authors and corresponds roughly to the point at which the distribution deviates from the power-law form indicated by the fit. The data for ten authors and more have been binned logarithmically to minimize statistical fluctuations.

and without these papers. The choice of fifty authors as the cutoff point was made by inspection of the distribution of author numbers shown in Fig. B.1. As the figure shows, the number of papers with a specific number of coauthors appears, roughly speaking, to follow a power law (in agreement with some previous studies [136], but not others [88]), but there is a marked deviation from the power-law form for the highest numbers of coauthors, above about fifty, indicating potentially different statistical laws in this regime, and possibly different underlying collaborative processes.

We also removed from the data a small number of citations. In a few cases a paper is listed as citing itself, which we assume to be an error. In a number of other cases papers cite others that were published at a later time, which violates causality. These too are assumed to be erroneous and are removed. Finally, the data indicate that some papers cited the same other paper several times within the one bibliography; such multiple citations we count as a single citation.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] C. Aicher, A. Z. Jacobs, and A. Clauset. Learning latent block structure in weighted networks. *Journal of Complex Networks*, June 2014.

[2] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

[3] N. Alon, M. Babaioff, R. Karidi, R. Lavi, and M. Tennenholtz. Sequential voting with externalities: Herding in social networks. In *13th ACM Conference on Electronic Commerce*, page 36, 2012.

[4] E. Altman, F. De Pellegrini, R. El-Azouzi, D. Miorandi, and T. Jimenez. Emergence of equilibria from individual strategies in online content diffusion. In *NetSciCom Workshop*, 2013.

[5] Arash A. Amini, Aiyou Chen, Peter J. Bickel, and Elizaveta Levina. Pseudo-likelihood methods for community detection in large sparse networks. *Annals of Statistics*, 41:2097–2122, 2013.

[6] Reid Andersen and Yuval Peres. Finding sparse cuts locally using evolving sets. In *41st annual ACM Symposium on Theory of Computing*, pages 235–244. ACM, 2009.

[7] W. B. Arthur. Competing technologies, increasing returns, and lock-in by historical events. *Economic Journal*, 99(394):116–131, 1989.

[8] Venkatesh Bala and Sanjeev Goyal. A noncooperative model of network formation. *Econometrica*, 68(5):1181–1229, 2000.

[9] Brian Ball and M. E. J. Newman. Friendship networks and social status. *Network Science*, 1:16–30, 2013.

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

[11] A. Banerjee, A. G. Chandrasekhar, E. Duflo, and M. O. Jackson. The diffusion of microfinance. *Science*, 341(6144), 2013.

[12] A. V. Banerjee. A simple model of herd behavior. *The Quarterly Journal of Economics*, 107(3):797–817, 1992.

[13] David L. Banks and Kathleen M. Carley. Models for network evolution. *Journal of Mathematical Sociology*, 21:173–196, 1996.

[14] Albert-Laszlo Barabási and Reka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[15] Christoph Bartneck and Servaas Kokkelmans. Detecting h-index manipulation through self-citation analysis. *Scientometrics*, 87(1):85–98, 2011.

[16] F. M. Bass. A new product growth for model consumer durables. *Management Science*, 15(5):215–227, 1969.

[17] Danielle S. Bassett, Nicholas F. Wymbs, Mason a. Porter, Peter J. Mucha, and Scott T. Grafton. Cross-linked structure of network evolution. *Chaos*, 24(1): 013112, 2014.

[18] V. Batagelj, A. Mrvar, and M. Zaveršnik. Network analysis of dictionaries. In *Language Technologies*, pages 135–142, Ljubljana, Slovenia, 2002.

[19] Joshua Batson, Daniel A Spielman, Nikhil Srivastava, and Shang-Hua Teng. Spectral sparsification of graphs: theory and algorithms. *Communications of the ACM*, 56(8):87–94, 2013.

[20] Carl T. Bergstrom, Jevin D. West, and Marc A. Wiseman. The eigenfactor metrics. *J. Neurosci.*, 28:11433–11434, 2008.

[21] S. Bikhchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy*, 100(5):992–1026, 1992.

[22] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[23] Mindaugas Bloznelis and Valentas Kurauskas. Clustering function: a measure of social influence. Preprint arxiv:1107.1155, 2012.

[24] L. E. Blume. The statistical mechanics of strategic interaction. *Games and Economic Behavior*, 5(3):387–424, 1993.

[25] Phillip F. Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2:113–120, 1972.

[26] Stephen P Borgatti. Centrality and network flow. *Social networks*, 27(1):55–71, 2005.

[27] Stephen P Borgatti and Martin G Everett. Models of core/periphery structures. *Social networks*, 21(4):375–395, 2000.

[28] Stephen P Borgatti and Martin G Everett. A graph-theoretic perspective on centrality. *Social networks*, 28(4):466–484, 2006.

[29] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks*, 30:107–117, 1998.

[30] J. J. Brown and P. H. Reingen. Social ties and word-of-mouth referral behavior. *Journal of Consumer Research*, 14:350–362, 1987.

[31] Z. Cao, X. Chen, and C. Wang. How to schedule the marketing of products with negative externalities. In *Computing and Combinatorics*, pages 122–133. 2013.

[32] Damon Centola and Michael Macy. Complex contagions and the weakness of long ties. *American journal of Sociology*, 113(3):702–734, 2007.

[33] P. Chen and S. Redner. Community structure of the physical review citation network. *Journal of Informetrics*, 4:278–290, 2010.

[34] Justin Cheng, Lada A Adamic, Jon M Kleinberg, and Jure Leskovec. Do cascades recur? In *25th International Conference on World Wide Web*, pages 671–681. International World Wide Web Conferences Steering Committee, 2016.

[35] F. Chierichetti, J. Kleinberg, and A. Panconesi. How to schedule a cascade in an arbitrary graph. In *13th ACM Conference on Electronic Commerce*, pages 355–368, 2012.

[36] J. P. Choi. Herd behavior, the "penguin effect," and the suppression of informational diffusion: An analysis of informational externalities and payoff interdependency. *RAND Journal of Economics*, 28(3):407–425, 1997.

[37] Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99 (25):15879–15882, 2002.

[38] Fan Chung, Linyuan Lu, and Van Vu. Spectra of random graphs with given expected degrees. *Proc. Natl. Acad. Sci. USA*, 100:6313–6318, 2003.

[39] Aaron Clauset, Cristopher Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[40] Aaron Clauset, Cosma R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51:661–703, 2009.

[41] J. Coleman, E. Katz, and H. Menzel. The diffusion of an innovation among physicians. *Sociometry*, 20:253–270, 1957.

[42] J. S. Coleman. Social capital in the creation of human capital. *American Journal of Sociology*, 94:95–120, 1988.

[43] T. G. Conley and C. R. Udry. Learning about a new technology: Pineapple in Ghana. *American Economic Review*, 100(1):35–69, 2010.

[44] Mihai Cucuringu and Michael W. Mahoney. Localization on low-order eigenvectors of data matrices. Preprint arxiv:1109.1355, 2011.

[45] Jörn Davidsen, Holger Ebel, and Stefan Bornholdt. Emergence of a small world from local interactions: Modeling acquaintance networks. *Physical Review Letters*, 88, 2002.

[46] James B. Davies, Susanna Sandström, Anthony B. Shorrocks, and Edward N. Wolff. The level and distribution of global household wealth. Working Paper 15508, National Bureau of Economic Research, 2009.

[47] Wouter de Nooy, Andrej Mrvar, and Vladimir Batagelj. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, Cambridge, 2 edition, 2011.

[48] Derek J. de Solla Price. *Science since Babylon*. Yale University Press, New Haven, 1961.

[49] Derek J. de Solla Price. *Little Science, Big Science*. Columbia University Press, New York, 1963.

[50] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Phys. Rev. Lett.*, 107:065701, 2011.

[51] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84:066106, 2011.

[52] E. Dekel and M. Piccione. Sequential voting procedures in symmetric binary elections. *Journal of Political Economy*, 108(1):34–55, 2000.

[53] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Statist. Soc. B*, 39:185–197, 1977.

[54] P. Domingos and M. Richardson. Mining the network value of customers. In *7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–66, 2001.

[55] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Size-dependent degree distribution of a scale-free growing network. *Physical Review E*, 63: 062101, 2001.

[56] Sergey N Dorogovtsev and José Fernando F Mendes. Effect of the accelerating growth of communications networks on their structure. *Physical Review E*, 63 (2):025101, 2001.

[57] David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world.* Cambridge University Press, 2010.

[58] G. Ellison. Learning, local interaction, and coordination. *Econometrica*, 61: 1047–1047, 1993.

[59] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[60] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

[61] Paul Erdős and Alfréd Rényi. On the strength of connectedness of a random graph. *Acta Mathematica Scientia Hungary*, 12:261–267, 1961.

[62] Kasper Astrup Eriksen, Ingve Simonsen, Sergei Maslov, and Kim Sneppen. Modularity and extreme edges of the Internet. *Phys. Rev. Lett.*, 90:148701, 2003.

[63] Leonardo Ermann, KlausM. Frahm, and DimaL. Shepelyansky. Spectral properties of google matrix of wikipedia and other networks. *The European Physical Journal B*, 86(5):193, 2013.

[64] Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, 8:128–140, 1741.

[65] Illés J. Farkas, Imre Derényi, Albert-László Barabási, and Tamás Vicsek. Spectra of "real-world" graphs: Beyond the semicircle law. *Physical Review E*, 64: 026704, 2001.

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

[67] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[68] A. Galeotti, .S Goyal, M. O. Jackson, F. Vega-Redondo, and L. Yariv. Network games. *Review of Economic Studies*, 77(1):218–244, 2010.

[69] K.-I. Goh, B. Kahng, and D. Kim. Spectra and eigenvectors of scale-free networks. *Physical Review E*, 64:051903, 2001.

[70] Anna Goldenberg, Alice X. Zheng, Stephen E. Feinberg, and Edoardo M. Airoldi. A survey of statistical network structures. *Foundations and Trends in Machine Learning*, 2:1–117, 2009.

[71] J. Goldenberg, B. Libai, and E. Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review*, 9(3):1–18, 2001.

[72] A. V. Goltsev, S. N. Dorogovtsev, J. G. Oliveira, and J. F. F. Mendes. Localization and spreading of diseases in complex networks. *Phys. Rev. Lett.*, 109: 128702, 2012.

[73] Sanjeev Goyal, Hoda Heidari, and Michael Kearns. Competitive contagion in networks. *Games and Economic Behavior*, 2014.

[74] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83:1420–1443, 1978.

[75] M. S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78:1360–1380, 1973.

[76] Peter Grassberger. On the critical behavior of the general epidemic process and dynamical percolation. *Mathematical Biosciences*, 63(2):157–172, 1983.

[77] J. W. Grossman. The evolution of the mathematical research collaboration graph. *Congressus Numerantium*, 158:202–212, 2002.

[78] Jerrold W. Grossman and Patrick D. F. Ion. On a portion of the well-known collaboration graph. *Congressus Numerantium*, 108:129–131, 1995.

[79] Roger Guimerà and Marta Sales-pardo. Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National Academy of Sciences*, 106(52):22073–22078, 2009.

[80] Roger Guimera, Marta Sales-Pardo, and Luís A Nunes Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101, 2004.

[81] M. Hajiaghayi, H. Mahini, and A. Sawant. Scheduling a cascade with opposing influences. In *Algorithmic Game Theory*, volume 8146 of *Lecture Notes in Computer Science*, pages 195–206. 2013.

[82] David G Harris and Aravind Srinivasan. Improved bounds and algorithms for graph cuts and network reliability. In *25th ACM-SIAM Symposium on Discrete Algorithms*, pages 259–278, 2014.

[83] K. Hashimoto. Zeta functions of finite graphs and representations of p-adic groups. *Adv. Stud. Pure Math.*, 15:211–280, 1989.

[84] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Science*, 102(46):16569–16572, 2005.

[85] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97 (460):1090–1098, 2002.

[86] P W Holland, K B Laskey, and S Leinhardt. Stochastic blockmodels: Some first steps. *Social Networks*, 5:109–137, 1983.

[87] Petter Holme and Beom Jun Kim. Growing scale-free networks with tunable clustering. *Physical Review E*, 65:026107, 2002.

[88] Jiann-Wien Hsu and Ding-Wei Huang. Distribution for the number of coauthors. *Physical Review E*, 80, 2009.

[89] Jian Huang, Ziming Zhuang, Jia Li, and C Lee Giles. Collaboration over time: characterizing and modeling network evolution. In *2008 International Conference on Web Search and Data Mining*, pages 107–116. ACM, 2008.

[90] Matthew O Jackson. A survey of network formation models: stability and efficiency. *Group Formation in Economics: Networks, Clubs, and Coalitions*, pages 11–49, 2005.

[91] Matthew O Jackson and Asher Wolinsky. A strategic model of social and economic networks. *Journal of Economic Theory*, 71(1):44–74, 1996.

[92] Matthew O Jackson and Yves Zenou. Games on networks. *Handbook of game theory*, 4, 2014.

[93] Emily M. Jin, Michelle Girvan, and M. E. J. Newman. Structure of growing social networks. *Physical Review E*, 64, 2001.

[94] Brian Karrer and M. E. J. Newman. Random graph models for directed acyclic networks. *Physical Review E*, 80:046110, 2009.

[95] Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83:016107, 2011.

[96] Brian Karrer, M. E. J. Newman, and Lenka Zdeborová. Percolation on sparse networks. *Physical Review Letters*, 113(20):208702, 2014.

[97] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[98] Michael Kearns, Michael L Littman, and Satinder Singh. Graphical models for game theory. In *17th Conference on Uncertainty in Artificial Intelligence*, pages 253–260. Morgan Kaufmann Publishers Inc., 2001.

[99] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[100] Brian W Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970.

[101] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *32nd annual ACM symposium on Theory of computing*, pages 163–170. ACM, 2000.

[102] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46:604–632, 1999.

[103] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[104] Nevan J Krogan, Gerard Cagney, Haiyuan Yu, Gouqing Zhong, Xinghua Guo, Alexandr Ignatchenko, Joyce Li, Shuye Pu, Nira Datta, Aaron P Tikuisis, Thanuja Punna, José M Peregrín-Alvarez, Michael Shales, Xin Zhang, Michael Davey, Mark D Robinson, Alberto Paccanaro, James E Bray, Anthony Sheung, Bryan Beattie, Dawn P Richards, Veronica Canadien, Atanas Lalev, Frank Mena, Peter Wong, Andrei Starostine, Myra M Canete, James Vlasblom, Samuel Wu, Chris Orsi, Sean R Collins, Shamanta Chandran, Robin Haw, Jennifer J Rilstone, Kiran Gandi, Natalie J Thompson, Gabe Musso, Peter St Onge, Shaun Ghanny, Mandy H Y Lam, Gareth Butland, Amin M Altaf-Ul, Shigehiko Kanaya, Ali Shilatifard, Erin O'Shea, Jonathan S Weissman, C James Ingles, Timothy R Hughes, John Parkinson, Mark Gerstein, Shoshana J Wodak, Andrew Emili, and Jack F Greenblatt. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, 440(7084):637–43, March 2006.

[105] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proc. Natl. Acad. Sci. USA*, 110:20935–20940, 2013.

[106] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D Sivakumar, Andrew Tomkins, and Eli Upfal. Stochastic models for the web graph. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 57–65. IEEE, 2000.

[107] Kenichi Kurihara, Yoshitaka Kameya, and Taisuke Sato. A frequency-based stochastic blockmodel. *Workshop on Information-Based Induction Sciences*, 2006.

[108] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4): 046110, 2008.

[109] Daniel B Larremore, Aaron Clauset, and Abigail Z Jacobs. Efficiently inferring community structure in bipartite networks. *Physical Review E*, 90(1):012805, 2014.

[110] K. Lerman and R. Ghosh. Information contagion: An empirical study of the spread of news on Digg and Twitter social networks. In *4th International Conference on Weblogs and Social Media*, pages 90–97, 2010.

[111] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, 2005. Association of Computing Machinery.

[112] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[113] A. J. Lotka. The frequency distribution of scientific production. *J. Wash. Acad. Sci.*, 16:317–323, 1926.

[114] James MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[115] V. Mahajan, E. Muller, and F. M. Bass. New product diffusion models in marketing: A review and directions for research. *Journal of Marketing*, 54: 1–26, 1990.

[116] Travis Martin, Brian Ball, Brian Karrer, and M. E. J. Newman. Coauthorship and citation patterns in the physical review. *Physical Review E*, 88(1):012814, 2013.

[117] Travis Martin, Grant Schoenebeck, and Mike Wellman. Characterizing strategic cascades on networks. In *15th ACM conference on Economics and computation*, pages 113–130. ACM, 2014.

[118] Travis Martin, Xiao Zhang, and M. E. J. Newman. Localization and centrality in networks. *Physical Review E*, 90(5):052808, 2014.

[119] Travis Martin, Brian Ball, and M. E. J. Newman. Structural inference for uncertain networks. *Physical Review E*, 93(1):012306, 2016.

[120] Travis Martin, Jake M Hofman, Amit Sharma, Ashton Anderson, and Duncan J Watts. Exploring limits to prediction in complex social systems. In *25th International Conference on World Wide Web*, pages 683–694, 2016.

[121] A. Mas-Colell, M.D. Whinston, and J.R. Green. *Microeconomic Theory*, volume 1. Oxford University Press, 1995.

[122] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. New York, 2nd edition, 2008.

[123] Frank McSherry. Spectral partitioning of random graphs. In *42nd Symposium on Foundations of Computer Science*, pages 529–537. IEEE, 2001.

[124] Sergey Melnik, Adam Hackett, Mason A. Porter, Peter J. Mucha, and James P. Gleeson. The unreasonable effectiveness of tree-based theory for networks with clustering. *Physical Review E*, 83(3), 2011.

[125] Marc Mézard and Andrea Montanari. *Information, Physics, and Computation.* Oxford University Press, Oxford, 2009.

[126] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303:1538–1542, 2004.

[127] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and Analysis of Online Social Networks. In *ACM/Usenix Internet Measurement Conference*, San Diego, CA, October 2007.

[128] Denis Mollison. Spatial contact models for ecological and epidemic spread. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 283–326, 1977.

[129] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6:161–179, 1995.

[130] Jacob L Moreno. *Who shall survive*, volume 58. JSTOR, 1934.

[131] S. Morris. Contagion. *Review of Economic Studies*, 67(1):57–78, 2000.

[132] E. Mossel, Joe Neeman, and Omer Tamuz. Majority dynamics and aggregation of information on social networks. *Autonomous Agents and Multi-Agent Systems*, pages 1–22, 2012.

[133] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[134] Raj Rao Nadakuditi and M. E. J. Newman. Spectra of random graphs with arbitrary expected degrees. *Physical Review E*, 87:012803, 2013.

[135] M. E. J. Newman. The structure of scientific collaboration networks. *Proc. Nat. Acad. Sci. USA*, 98(2):404–409, 2001.

[136] M. E. J. Newman. Scientific collaboration networks: I. Network construction and fundamental results. *Physical Review E*, 64:016131, 2001.

[137] M. E. J. Newman. Spread of epidemic disease on networks. *Physical review E*, 66(1):016128, 2002.

[138] M. E. J. Newman. Analysis of weighted networks. *Physical Review E*, 70: 056131, 2004.

[139] M. E. J. Newman. The first-mover advantage in scientific publication. *Europhys. Lett.*, 86(6), 2009.

[140] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, Oxford, 2010.

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

[142] M. E. J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[143] M. E. J. Newman and Juyong Park. Why social networks are different from other types of networks. *Physical Review E*, 68, 2003.

[144] M. E. J. Newman, Steven H. Strogatz, and Duncan J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64:026118, 2001.

[145] Kim Norlen, Gabriel Lucas, Mike Gebbie, and John Chuang. Eva: Extraction, visualization and analysis of the telecommunications and media ownership network. In *Proc. Internat. Telecom. Soc., Seoul Korea*, 2002.

[146] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *J. Amer. Stat. Assoc.*, 96:1077–1087, 2001.

[147] Tore Opsahl. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social Networks*, 35(2):159–167, 2013.

[148] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[149] Raj Kumar Pan, Kimmo Kaski, and Santo Fortunato. World citation and collaboration networks: uncovering the role of geography in science. *Scientific reports*, 2, 2012.

[150] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *AAAI*, pages 133–136, 1982.

[151] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, CA, 1988.

[152] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3): 61–74, 1999.

[153] Derek J. de Solla Price. Networks of scientific papers. *Science*, 149:510–515, 1965.

[154] Derek J. de Solla Price. A general theory of bibliometric and other cumulative advantage processes. *J. Amer. Soc. Inform. Sci.*, 27:292–306, 1976.

[155] Tai Qin and Karl Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. Technical report, 2013.

[156] Filippo Radicchi, Santo Fortunato, Benjamin Markines, and Alessandro Vespignani. Diffusion of scientific credits and the ranking of scientists. *Physical Review E*, 80(5):056103, 2009.

[157] Anatol Rapoport. Contribution to the theory of random and biased nets. *The bulletin of mathematical biophysics*, 19(4):257–277, 1957.

[158] W. Raub and J. Weesie. Reputation and efficiency in social interactions: An example of network effects. *American Journal of Sociology*, 96:626–654, 1990.

[159] Sydney Redner. Citation statistics from 110 years of Physical Review. *Physics Today*, 58:49–54, 2005.

[160] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 61–70, 2002.

[161] Antonio Robles-Kelly and Edwin R. Hancock. A probabilistic spectral framework for grouping and segmentation. *Pattern Recognition*, 37:1387–1405, 2004.

[162] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118, 2008.

[163] Tim Roughgarden and Éva Tardos. How bad is selfish routing? *Journal of the ACM (JACM)*, 49(2):236–259, 2002.

[164] M. Medo S. Gualdi and Y.-C. Zhang. Influence, originality and similarity in directed acyclic graphs. *Europhysics Letters*, 96(1), 2011.

[165] Alaa Saade, Marc Lelarge, Florent Krzakala, and Lenka Zdeborová. Spectral detection in the censored block model. In *International Symposium on Information Theory*, pages 1184–1188, 2015.

[166] Soma Sanyal. Effect of citation patterns on network structure. 2007.

[167] T. C. Schelling. Dynamic models of segregation. *Journal of Mathematical Sociology*, 1(2):143–186, 1971.

[168] William Shockley. On the statistics of individual variations of productivity in research laboratories. *Proc. IRE*, 45:279–290, 1957.

[169] T. A. B. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997.

[170] Gilbert Strang. *Introduction to Linear Algebra.* Wellesley Cambridge Press, Wellesley, MA, 2009.

[171] György Szabó and Gabor Fath. Evolutionary games on graphs. *Physics reports*, 446(4):97–216, 2007.

[172] Terence Tao. *Topics in Random Matrix Theory.* American Mathematical Society, Providence, RI, 2012.

[173] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, pages 425–443, 1969.

[174] Christian von Mering, Lars J Jensen, Berend Snel, Sean D Hooper, Markus Krupp, Mathilde Foglierini, Nelly Jouffre, Martijn A Huynen, and Peer Bork. STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic acids research*, 33(Database issue):D433–7, January 2005.

[175] H. Von Stackelberg. *Market Structure and Equilibrium.* Springer, 2011.

[176] Roland Wagner-Döbler. Continuity and discontinuity of collaboration behaviour since 1800——from a bibliometric point of view. *Scientometrics*, 52(3):503–517, 2001.

[177] Matthew L. Wallace, Vincent Larivière, and Yves Gingras. Modeling a century of citation distributions. *Journal of Informetrics*, 3:296–303, 2009.

[178] Matthew L. Wallace, Vincent Larivière, and Yves Gingras. A small world of citations? the influence of collaboration networks on citation practices. *PLoS ONE*, 7, 2012.

[179] Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

[180] Stanley Wasserman and Katherine Faust. *Social Network Analysis.* Cambridge University Press, Cambridge, 1994.

[181] D. J. Watts. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9):5766–5771, 2002.

[182] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[183] Duncan J Watts, Peter Sheridan Dodds, and M. E. J. Newman. Identity and search in social networks. *Science*, 296(5571):1302–1305, 2002.

[184] Harrison C White, Scott A Boorman, and Ronald L Breiger. Social structure from multiple networks. i. blockmodels of roles and positions. *American journal of sociology*, pages 730–780, 1976.

[185] Allen W. Wilhite and Eric A. Fong. Coercive citation in academic publishing. *Science*, 335(6068):542–543, 2012.

[186] Stefan Wuchty, Benjamin F Jones, and Brian Uzzi. The increasing dominance of teams in production of knowledge. *Science*, 316(5827):1036–1039, 2007.

[187] Jiaming Xu, Laurent Massoulié, and Marc Lelarge. Edge label inference in generalized stochastic blockmodels: From spectral theory to impossibility results. In *27th Conference on Learning Theory*, pages 903–920, 2014.

[188] Xiaoran Yan, Cosma Shalizi, Jacob E Jensen, Florent Krzakala, Cristopher Moore, Lenka Zdeborová, Pan Zhang, and Yaojia Zhu. Model selection for degree-corrected block models. *Journal of statistical mechanics*, page 5007, 2014.

[189] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

[190] Xiao Zhang and M. E. J. Newman. Multiway spectral community detection in networks. *Physical Review E*, 92(5):052808, 2015.

[191] Xiao Zhang, Raj Rao Nadakuditi, and M. E. J Newman. Spectra of random graphs with community structure and arbitrary degrees. *Physical Review E*, 89 (4):042816, 2014.

[192] Xiao Zhang, Travis Martin, and M. E J Newman. Identification of core-periphery structure in networks. *Physical Review E*, 91(3), 2015.

[193] Han Zhu, Xinran Wang, and Jian-Yang Zhu. Effect of aging on network structure. *Physical Review E*, 68, 2003.