Title
Nonlinear opinion models and other networked systems

Author
Kureh, Yacoub Hanna

Publication Date
2020

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Nonlinear opinion models and other networked systems

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Yacoub Hanna Kureh

2020

ABSTRACT OF THE DISSERTATION

Nonlinear opinion models
and other networked systems

by

Yacoub Hanna Kureh
Doctor of Philosophy in Mathematics
University of California, Los Angeles, 2020
Professor Mason Alexander Porter, Chair

Networks play a critical role in many physical, biological, and social systems. In this thesis, we investigate tools to model and analyze networked systems. We first examine some of the ways in which we can model social dynamics that take place on networks. We then study two recently developed data-analysis methods that employ a network framework and explore new ways in which they can be used to find meaningful signals in large data sets.

In the first half of the thesis, we study opinion dynamics on networks. We begin by examining a class of opinion models, known as coevolving voter models (CVM), that couple the mechanisms of opinion formation and changing social connections. We then propose a version of CVMs that incorporates nonlinearity. In our models, we assume that individuals strive to achieve harmony and avoid disagreement, both by changing their social connections to reflect their opinions and by changing their opinions to reflect their social connections. By taking a minimalist approach to modeling social dynamics, we hope to gain a deeper understanding of how these two mechanisms can give rise to social phenomena such as the "majority illusion". Comparing several versions of CVMs, we find that seemingly small changes in update rules can lead to strikingly different behaviors. A particularly interesting feature of our nonlinear CVMs is that, under certain conditions, the opinion state that is held initially by a minority of the nodes can effectively spread to almost every node in a network if the minority nodes view themselves as the majority. We then discuss an

ongoing project that involves another class of opinion models called bounded-confidence models. Specifically, we examine extensions of bounded-confidence models on hypergraphs and discuss some preliminary findings.

In the second half of the thesis, we study problems in data analysis. We begin by considering topological structures as a tool to study integrated circuit (IC) devices. In particular, we examine a problem in the design and manufacturing of IC devices using topological data analysis (TDA), which is based on network structures called simplicial complexes. Failures in IC devices generally occur near the tolerance limits of photolithography systems, such as at the minimum separation distance between adjacent electronic components. However, for complex arrangements of electronic components, simply ensuring minimal separation is insufficient to guarantee that one can manufacture an IC design accurately and reliably. We apply tools from TDA to compare data from IC designs. Without inputting domain knowledge, we are able to infer several results about the IC design-manufacturing process. Finally, we discuss an ongoing project in the analysis of network data. Specifically, we explore applications of a recently developed algorithm called network dictionary learning (NDL) and discuss problems of network reconstruction and denoising using NDL on both synthetic and real-world networks.

The dissertation of Yacoub Hanna Kureh is approved.

Marcus Leigh Roper

Andrea L. Bertozzi

Christopher R. Anderson

Mason Alexander Porter, Committee Chair

University of California, Los Angeles

2020

To mom. I miss you every day.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am fortunate to have received funding support in my final two years from the Army Research Office (through MURI grant W911NF1810208) and the National Science Foundation (grant number 1922952) through the Algorithms for Threat Detection program.

I am grateful to Andrea Baronchelli, James Gleeson, Thilo Gross, Istvan Kiss, Joel Miller, David O'Sullivan, Samuel Scarpino, and Bill Shi, who helped me learn about opinion models and have given thoughtful comments on manuscripts.

I would like to thank all of my collaborators. Vito Dai and Luigi Capodieci provided me with a fascinating project to work on and amazing mentorship when I was interning at Motivo, Inc. I thank William Oakley, David Kempe, Jeffrey Brantingham, and Mason Porter, with whom I had the opportunity to work on many interesting projects in the last few years. I thank Alexis Piazza and Victor Leung for introducing me to the problems around the underreporting of youth homelessness and allowing me to work with them. Although our collaborations are still nascent, working with Heather Brooks, Michelle Feng, Hanbaek Lyu, and Josh Vendrow on various projects has been immensely rewarding.

I would next like to thank my thesis committee members — Chris Anderson, Andrea Bertozzi, and Marcus Roper — for their suggestions and insights during my ATC. I have also taken many stimulating courses and seminars with each of them. I would also like to thank Martha Contreras and Maida Bassili for their help with the administrative side of the PhD.

I would like to thank my high school teachers — Gary Fox, Maria Johnson, and Carol Smith — who went above and beyond to provide me with excellent guidance. I would also like to thank Sa'ed Atshan, who inspires me to go the extra mile and whose spirit and intellect I'll always admire.

I am exceedingly grateful to my advisor, Mason Porter. He has been extraordinarily generous with his time and energy to discuss ideas at all stages, write letters of recommendation, answer all kinds of questions, and provide detailed comments on even the rough manuscripts.

For all that and much more, I'm very thankful.

I am deeply indebted to my friends. In the last six years, I have met so many incredible people who have helped me grow as a mathematician and as a person. I thank Michael Andrews, Zachary Boyd, Jeremy Brightbill, Heather Brooks, Kevin Carlson, Shyr-Shea Chang, Peter Cheng, Praptee Chowdhury, Ian Coley, Xenia Deviatkina-Loh, Manpreet Dhillon, Brent Edmunds, Michelle Feng, Julian Gold, Androulla Hadjikyriacou, Robert Hannah, Amelia Hill, ChiChi Huang, Liz Jackson, Shawndeez Jadalizadeh, Ria Jain, Florian Klimm, Jonathan Koch, Rachel Kuria, Rahim Kurwa, Zane Li, Alex Lin, Bon-Soon Lin, Michael Lindstrom, Zhiyun Lu, Hanbaek Lyu, Jean-Michel Maldague, Mike Miller, Jacob Moorman, Yi Ming Lai, Denali Molitor, William Oakley, Agatha Palma, Kabita Parajuli, Eitan Peled, Minh Pham, Andre Pradhana, Prateek Puri, Juan Carlos Ramírez, Alice Schwarze, Yehonatan Sella, Faria Tahir, Olufemi Taiwo, Stephanie Wang, and Alex Wertheim for their time, support, and friendship. Before coming to UCLA, I was very fortunate to meet Krishna Dasaratha, Laura Desert, Suvai Gunasekaran, Nathaniel Hay, Regina Hodge, Kathy Lee, Ananda Martin-Caughey, Paul Rubenstein, and Anna Wright, each of whom holds a special place in my heart.

Finally, I would like to thank my family for their unwavering support and tremendous encouragement. Thank you to my brothers, Maher and Salah, who played pivotal roles in my education; there's no way that I would have made it this far without them. Thank you to my dad, who worked tirelessly so that I would be able to focus on my education. And most importantly, thank you to my mom, who was my biggest fan, my first teacher, my role model, and my best friend. There's not a day that goes by that I don't remember how much we used to laugh together.

VITA

2009–2013 A.B. Mathematics, Harvard University.

2013–2014 M.A.S. Mathematics, University of Cambridge.

PUBLICATIONS AND PRESENTATIONS

Yacoub H. Kureh and Mason A. Porter. "Fitting in and breaking up: A nonlinear version of coevolving voter models." *arXiv:1907.11608*, 2020. (*Physical Review E*, in press)

Yacoub Kureh, Vito Dai, and Luigi Capodieci. "Persistent homology analysis of complex high-dimensional layout configurations for IC physical designs." In *Design-Process-Technology Co-optimization for Manufacturability XIII*, volume 10962. International Society for Optics and Photonics, 2019.

Yacoub H. Kureh, P. Jeffrey Brantingham, and Mason A. Porter. "Coupled point processes and network evolution dynamics." Presented at *SIAM Workshop on Network Science*, 2019.

Leah Rosenbaum, Mohit Agrawal, Leah Birch, Yacoub Kureh, Nam Lee, James Hant, and Brian Wood. "Calculating call blocking and utilization for communication satellites that use dynamic resource allocation." Proceedings of the IEEE Aerospace Conference, 2012.

LIST OF NOTATION

N	Number of nodes
V	Node set
E	Edge set
$\mathbb{N}_0 = \{0, 1, 2, \ldots\}$	Non-negative integers
$\mathbb{R}_{\geq 0} = [0, \infty)$	Non-negative real numbers
O	Opinion space
S	State function
E_D	Discordant-edge set
N_X	Number of nodes in state X
$[X]$	Expected number of nodes in state X
σ_i	Local survey of node i
q	Nonlinearity parameter (see Chapter 3)
$\mathcal{U}(a, b)$	Continuous uniform distribution on $[a, b]$
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$\Gamma(i, E) := \{j \in V : (i, j) \in E\} \cup \{i\}$	Closed neighborhood of node i
p	Edge probability in $G(N, p)$ random graph
$P_{\alpha\beta}$	Edge probabilities in SBM network
$\mathcal{P}(S)$	Power set of set S
$X_i,\ Y_j$	Critical dimensions in coverage data
ϵ	Resolution parameter in persistent homology
β_i	The ith Betti number
$\|M\|_F := \left(\sum_i \sum_j M_{ij}^2\right)^{1/2}$	Frobenius norm

CHAPTER 1

Introduction

1.1 Opinion Models on Networks

Whether it's which candidate to choose in an election or where to eat, our social circles influence our opinions and how they change [Ban71]. When choosing the people with whom we interact, our opinions in turn influence our social circles [MSC01]. The desire to improve understanding of this complex interplay between opinion formation and evolving social connections has become increasingly prominent amidst the ubiquity of social media and intensifying political polarization [Sun18]. Mathematical modeling of this interplay — such as with agent-based models — can yield insights into many pertinent questions [GJ10]. For example, do social-media platforms amplify extreme viewpoints by creating majority illusions [LYW16], such that certain opinions appear more mainstream than they really are? How do so-called "echo chambers", in which individuals have selective exposure to ideas that reinforce their own views, develop in these environments? More broadly, under what conditions does this complex interplay become a positive feedback loop in which opinions and social circles reinforce one another, driving individuals to extremes and potentially stifling civil discourse? To help understand these phenomena and examine these questions, it is important to develop sociological theory and accurate models of this interplay.

Opinion and social network formation have been studied using a variety of lenses from numerous disciplines [DVB16, JD15, SZL18, GLH19], including longstanding efforts to develop mathematical frameworks for modeling opinion dynamics by employing ideas from subjects such as statistical physics and nonlinear dynamics [CFL09]. There has also been much cross-fertilization with research in the modeling of disease spread [KMS17]. For ex-

ample, one can examine the "virality" of memes or seemingly contagious behaviors [LA18]. In developing mathematical frameworks for studying opinion dynamics, incorporating social network structure can significantly improve both the accuracy of mathematical models and the understanding of spreading processes [NVT19, LA18, PCV15]. For example, research on the severe acute respiratory syndrome (SARS) outbreaks by L. A. Meyers et al. illustrated the utility of accounting for social networks when assessing public-health strategies [MPN05]. This stems from a network's influence on the properties of dynamical processes that take place on them [PG16]. Additionally, networks themselves are typically not time-independent, as they often evolve in response to a dynamical process and in turn influence that process [SPS13, DVB14]. For instance, in the spread of diseases, networks of interactions can change as a result of quarantines or different daily habits when somebody is ill. Similarly, in social media, individuals can choose to "follow" or "unfollow" other individuals (or other types of accounts) in response to posted content. The interplay of dynamics on networks and dynamics of networks[1] is a rapidly growing area of study in many disciplines [PG16, LA18]. Although much of the prior work on such "coevolving" (also known as "adaptive") network dynamics has focused on studying the complex behavior of simple models in abstract settings, there have also been efforts at incorporating further realism into such models [Red19] and at applying such models to study empirical data in situations — including vote shares in United States elections [FSR14] and the swarming dynamics of locusts [HZD11] — that can involve notions of "opinions" and consensus.

In Chapter 2, we briefly review the literature on voter models by giving a short history of their origin. We present several variants of voter models and discuss their differences. We then discuss some of the techniques that are used to study these models, with particular attention to moment equations. We then describe some of the (many) extensions of voter models that incorporate various additional features. Finally, we focus in on one particular extension, coevolving voter models (CVMs), in which node states and network structure update as a coupled stochastic process.

[1]See Refs. [HS12, Hol15, HS19] for reviews of research on time-dependent networks.

In Chapter 3, we investigate a nonlinear version of CVMs. In contrast to linear versions of CVMs, in our nonlinear version, the probability that a node rewires or adopts is a function of how well it "fits in" with the nodes in its neighborhood. To explore this idea, we incorporate a local-survey parameter σ_i that represents the fraction of neighbors of an updating node i that share its opinion state. In an update, with probability σ_i^q (for some nonlinearity parameter q), the updating node rewires; with complementary probability $1 - \sigma_i^q$, the updating node adopts a new opinion state. We study this mechanism using three rewiring schemes: after an updating node deletes one of its discordant edges, it then either (1) "rewires-to-random" by choosing a new neighbor in a random process; (2) "rewires-to-same" by choosing a new neighbor in a random process from nodes that share its state; or (3) "rewires-to-none" by not rewiring at all (akin to "unfriending" on social media). We compare our nonlinear CVM to several existing linear CVMs on various network architectures. Relative to those models, we find in our model that initial network topology plays a larger role in the dynamics and that the choice of rewiring mechanism plays a smaller role. A particularly interesting feature of our model is that, under certain conditions, the opinion state that is held initially by a minority of the nodes can effectively spread to almost every node in a network if the minority nodes view themselves as the majority. In light of this observation, we relate our results to recent work on the majority illusion in social networks. Chapter 3 contains work done in collaboration with Mason A. Porter (UCLA) in Ref. [KP20].

In Chapter 4, we turn our attention to bounded-confidence models (BCM), another class of opinions models on networks. We discuss work that is a collaboration with Heather Z. Brooks (UCLA), Michelle Feng (UCLA), and Mason A. Porter on adaptations of BCMs to hypergraphs. We discuss the model formulation, preliminary analysis, and a few numerical computations.

1.2 Data Analysis of Networked Systems

Physical defects and electrical failures of integrated circuit (IC) devices generally occur near the tolerance limits of photolithography systems, such as at the minimum separation distance

between adjacent electronic components. However, for complex arrangements of electronic components, simply ensuring minimal separation is insufficient to guarantee that one can manufacture the IC design accurately and reliably. In Chapter 5, we apply tools from topological data analysis to data sets that were extracted from IC layouts using pattern matching. We find that plotting Betti numbers (a topological invariant [Hat05]) as a function of a resolution parameter enables both (1) a compressed yet robust signature of IC layout data sets and (2) an efficient means of quantitatively comparing data within a single IC layout and across multiple IC layouts. Without inputting domain knowledge, we are able to infer several results about the design-manufacturing process. This chapter contains work that was done in collaboration with Vito Dai (Motivo, Inc.) and Luigi Capodieci (Motivo, Inc.) in Ref. [KDC19].

In Chapter 6, we discuss an ongoing collaboration on data analysis of networks. We explore applications of a new algorithm called network dictionary learning (NDL) that was recently developed by Lyu et al. in Ref. [LNB19]. We discuss problems of network reconstruction and denoising using NDL on both synthetic and real-world networks. This in-progress work is in collaboration with Hanbaek Lyu (UCLA), Josh Vendrow (UCLA), and Mason A. Porter.

CHAPTER 2

Background on Voter Models

2.1 Brief History of Voter Models

> "The main reason for [the] introduction [of voter models] was not so much a desire to model political systems, as the name might suggest, but rather the fact that voter models are exactly the class of spin systems to which duality can be applied most completely and successfully." – Thomas M. Liggett [Lig99]

The family of interacting particle systems known as "voter models" arose in the 1970s independently by Clifford and Sudbury [CS73] and by Holley and Liggett [HL75]. Clifford and Sudbury interpreted voter models in the context of spatial conflict. Two populations — call them red and blue — compete for territory, where each region is a node on a network. The types of networks that they considered are d-dimensional integer lattices. Each node in the lattice is held either by the red or the blue population. A stochastic invasion process converts a pair of adjacent red–blue territories (i.e., nodes connected by an edge held by different populations) into either red–red or blue–blue, with both of these events occurring as a Poisson process with rate 1. That is, a node held by the red population is invaded at a rate proportional to the number of neighboring nodes that are controlled by the blue population, and vice versa. For Holley and Liggett, the interpretation was nominally one of opinion dynamics, such that individuals intermittently reevaluate their position on some issue. Each individual is a node in a network (specifically, a d-dimensional integer lattice), and and each holds one of two possible positions, denoted by 0 and 1, on an issue. An individual waits an exponential time with parameter 1 (which is equivalent to Poisson process with rate 1) and then adopts the position of a neighbor chosen uniformly at random.

The voter models of Refs. [CS73, HL75] are Markov processes with two absorbing states. A model that is similar to voter models arose a decade earlier in the work of Glauber [Gla63]. Glauber formulated a stochastic version of the Ising model[1] to study its time-dependent statistics. In Glauber's stochastic Ising model, particles are arranged in a ring and have spin $+1$ or -1. Each particle's spin randomly flips between $+1$ or -1 at some Poisson rate. Glauber's model includes the tendency for the spin of a particle to positively correlate (or negatively correlate) with its neighboring particles' spins by having the transition probability for a particle's spin depend on the spins of its neighbors. The stochastic Ising model is also a Markov process, but (unlike the voter models of Refs. [CS73, HL75]) it does not have an absorbing state.

Since their inception, voter models have stimulated much research in fields such as probability and statistical mechanics [Lig99, CFL09, PG16]. Soon after, (perhaps due to their simplicity and partly due to their name), voter models started to spark interest in mathematical sociology [KS80]. Over the past forty years, scientists have studied various aspects of voter models, such as stationary states and how long it takes to converge to those states. Researchers have also used voter models as a starting point to formulate more intricate opinion models. In many cases, one endeavors to imbue a voter model with realism based on sociological theory, real-word phenomena, or current events. In Section 2.2, we attempt to disentangle some of the ambiguities that have arisen in the definition of "the" voter model. In Section 2.3, we review some of the tools that have been used for analyzing voter models. We pay special attention to moment closure approximations [Kue16]. In Section 2.4, we discuss some reality-inspired extensions of voter models. Finally, in Section 2.5, we focus our study on one particular extension, coevolving voter models (which we then generalize in Chapter 3).

[1] See Refs. [Len20, Isi25] for the original formulations of the Ising model.

2.2 "The" Voter Model

In the decades following the introduction of Holley and Liggett's voter model [HL75], there has been substantial research on the topic. During this time, the precise definition of "voter model" has splintered into several versions. Nowadays, voter models refer to a class of similar but unidentical models; unfortunately, this class of models is occasionally called "the" voter model in the literature (leading to ambiguity). Although it is not entirely clear what delineates a voter model from another similar opinion model, a voter model generally consists of the following features:

1. individuals (nodes),
2. their connections (edges) to other individuals (neighbors),
3. their opinions (so-called "opinion states"),
4. and a rule (a stochastic process) for updating opinions.

The first two items, nodes and edges, yield a network on which the dynamics takes place. For simplicity, we disallow multi-edges (i.e., more than one edge between two nodes) and self-edges (i.e., an edge between a node and itself). In Section 3.4, we discuss some situations in which one can relax these stipulations. The third item, opinion states, represent the possible opinions that the nodes may hold, with each node holding exactly one opinion at a time. The system's space of states, which we call "system states" to distinguish them from the opinion states, is given by the space of triples (V, E, S), where V is the set of nodes, $N := |V|$ denotes the number of nodes (if the network is finite), E is the set of edges[2], $S : V \to O$ is the *state function* that records the opinion state of each node, and O is the space of opinions. Most voter models use a binary opinion space such as $O = \{A, B\}$, but some use arbitrarily large sets [VR04, HN06, SMD13]. We focus on spaces with binary opinion states.

The fourth item, a stochastic process, defines how the system updates. In defining the update process, one must choose whether the system evolves in discrete or continuous time. The voter models in Refs. [CS73, HL75] were formulated in continuous time to

[2]We focus on edges that are pairwise, undirected, and unweighted but one can consider more general types of edges [SKV09, HUK10, CT12, BCP11, HK20].

obtain a system that evolves between system states following some transition rates. Discrete-time voter models, in which transition rates between systems states are replaced by transition probabilities, have also been studied frequently [Lig99, SR05]. The ways in which continuous-time and discrete-time voter models coincide and differ is an interesting topic; see Refs. [FMG16, SAR08] for details. The decision to use discrete-time or continuous-time involves many factors, such as the structure of one's data and the ability to simulate a model numerically[3]; or it may simply be personal/mathematical preference. In this thesis, we consider only discrete-time voter models and their extensions.

Discrete-time voter models evolve between system states in a stochastic, memoryless fashion, so we can construe each such system as a first-order Markov chain[4] in which the stochastic process determines the transition probabilities. For voter models on time-independent networks, the set V of nodes and the set E of edges remain fixed as a system evolves, but the function S updates as nodes change states. When we refer to the state function at a certain time, we use the notation S^t to explicitly indicate time-dependence, but otherwise we suppress explicit notation for time. When evaluating the function S for a specific node, we use square brackets, so $S[i]$ indicates the state of node i and $S^t[i]$ indicates the state of node i at time t. We use E_D [and $E_D(t)$ when specifying the time t explicitly] to denote the set

$$E_D = \{(u, v) \in E : S[u] \neq S[v]\} \tag{2.1}$$

of *discordant edges*, which are edges between disagreeing nodes, in E. We refer to edges between agreeing nodes as *concordant edges*.

An important choice to make when defining update rules for a discrete-time voter model is the use of synchronous versus asynchronous updating. Synchronous updating entails updating all of the nodes in unison at each time step. By contrast, in asynchronous updating, one chooses a node (at random using a specified random process) at each step to interact with one of its neighbors, while the state of the rest of the system is fixed. If the random

[3]See Appendix A, *Stochastic simulation of epidemics*, of Ref. [KMS17] for an introduction to simulating dynamics on a network.

[4]A Markov chain is the discrete-time version of a Markov process.

process for choosing which node to update has a uniform distribution over the nodes, then after N steps in an asynchronous voter model, one updates each node once on average in an N-node network. That is, N steps in an asynchronous model corresponds to one time step in a synchronous model [SAR08]. Therefore, when comparing an asynchronous model that uses a uniform random distribution to a synchronous model, the former evolves at a rate that is scaled by $1/N$ relative to the latter. The choice between synchronous and asynchronous updating is an important one, as it can have significant effects on the dynamics of voter models (in addition to the time scaling), including differences in the number of absorbing states [Gas15]. See Refs. [HG93, RCS09, FES11] for detailed discussions of the effects of synchronous versus asynchronous updating in voter models.

Another important choice to make is that between "node-based" and "edge-based" updating. In node-based voter models, in each step, one first selects a node i (at random following some distribution over the set of nodes) and then chooses a neighbor j (at random following some distribution over the set of neighbors of node i). If node i is isolated, the system does not change in that step. Another modeling choice that one needs to make is whether to update node i or node j. Under "direct" node-based rules, one updates the state of node i by copying the state of node j; under "reverse" node-based (which is also called "invasion process") rules, the roles are switched, so one updates the state of node j by copying the state of node i [SAR08]. The two node-based rules place focus on the entities of the system. By contrast, under edge-based (i.e., link-based) rules, in each step, one first selects an edge (i, j) at random following some distribution over the set of edges and then uniformly randomly selects one of the edge's incident nodes to update. That is, with equal probability, one updates node i by copying the state of node j or one updates node j by copying the state of node i.

The seemingly minor choice between direct node-based, reverse node-based, and edge-based rules can have very substantial effects on the dynamics of a voter model, including convergence time and steady-state behavior [NKB08, SR05, DVB14, PG16]. One can observe an immediate difference between the three rules based on how they bias the relationship between the degrees of nodes i and j. In the above node-based rules, the expected degree

of node j can differ from the expected degree of node i [Fel91, New18]. By contrast, in the above edge-based rules, it follows from symmetry that nodes i and j have identical expected degrees.

2.3 Analysis of Voter models

The tools that one uses to study voter models are often context-dependent. For questions about stationary system states on the d-dimensional integer lattice $\mathbb{Z}^d$, techniques from random-walk theory and duality theorems have been very useful [HL75, Lig99]. For questions about the time to consensus or probability of a certain type of consensus on finite heterogeneous networks, the Kolmogorov equations provide an effective framework in which to study voter models [SAR08]. In this section, we present tools that have more generic utility but are geared more for approximations than for exact results. These tools are particularly useful to study extensions of voter models [DGL12, DVB14, Jed17].

A complete description of the system state of a voter model is given by (V, E, S). Even for small networks, this amount of information is difficult to study, so we seek a coarse-grained description of the dynamics [Kue16]. One type of summary is a count of "state-specified motifs", which are subgraphs H in which the nodes of H are in specified states. Relevant counts of state-specified motifs include

1. $N_X(t) = |\{i \in V : S^t[i] = X\}|$, the number of nodes i at time t that are in state $X \in O = \{A, B\}$;

2. $N_{XY}(t) = |\{(i,j) \in V \times V : (i,j) \in E(t)\,, S^t[i] = X\,, S^t[j] = Y\}|$, the number of node pairs (i,j) at time t in which i and j are adjacent, node i is in state $X \in O = \{A, B\}$, and node j is in state $Y \in O$; and

3. $N_{XYZ}(t) = |\{(i,j,k) \in V \times V \times V : (i,j),(j,k) \in E(t)\,, S^t[i] = X\,, S^t[j] = Y\,, S^t[k] = Z\}|$, the number of node triples (i,j,k) at time t in which i and j are adjacent, j and k are adjacent, node i is in state $X \in O = \{A, B\}$, node j is in state $Y \in O$, and node k is in state $Z \in O$.

One can also compute counts for state-specified motifs with more than three nodes or with other subgraph structures, but we will not pursue this. Note that

$$\begin{aligned} N_A(t) + N_B(t) &= N\,, \\ N_{AA}(t) + N_{AB}(t) + N_{BA}(t) + N_{BB}(t) &= 2|E|\,, \\ N_{AB}(t) = N_{BA}(t) &= |E_D(t)|\,. \end{aligned}$$

We refer to a node in state X as an X-node, to a pair in the N_{XY} count as an XY-pair, and to a triple in the N_{XYZ} count as an XYZ-triple. It is also helpful to examine *state densities* $N_X(t)/N$, which give the fraction of nodes in a state X. We are also interested in the expected values (i.e., *moments*) of these quantities [KMS17]. Three examples of moments are

$$\begin{aligned} [X] = [X](t) &= \langle N_X(t) \rangle = \mathbb{E}[N_X(t)]\,, \\ [XY] = [XY](t) &= \langle N_{XY}(t) \rangle = \mathbb{E}[N_{XY}(t)]\,, \\ [XYZ] = [XYZ](t) &= \langle N_{XYZ}(t) \rangle = \mathbb{E}[N_{XYZ}(t)]\,. \end{aligned} \tag{2.2}$$

We call moments for state-specified motifs with k nodes $(k-1)$th-order moments.

One way of approximately capturing the dynamics of voter models is by writing balance equations for the moments, which are called *moment equations*. Writing moment equations involves calculating the rates of all processes that result in either the creation or destruction of state-specified motifs. In Figure 2.1, we show a flux diagram that captures this information for an asynchronous edge-based voter model. For example, the rate at which nodes in state A are converted to nodes in state B is proportional to the number of AB-pairs and BA-pairs. With rates from Figure 2.1, we arrive at the following system of differential equations:

$$\begin{aligned} \frac{d}{dt}[A] &= \frac{d}{dt}[B] = 0\,, \\ \frac{d}{dt}[AA] &= [AB] - [AAB] + [ABA]\,, \\ \frac{d}{dt}[BB] &= [AB] - [BBA] + [BAB]\,, \\ \frac{d}{dt}2[AB] &= -2[AB] - [ABA] - [BAB] + [AAB] + [BBA]\,. \end{aligned} \tag{2.3}$$

The vanishing of the time derivative of the 0th-order moments is a consequence of the symmetry that arises from edge-based updates. For all $k \geq 1$, the time derivative of a

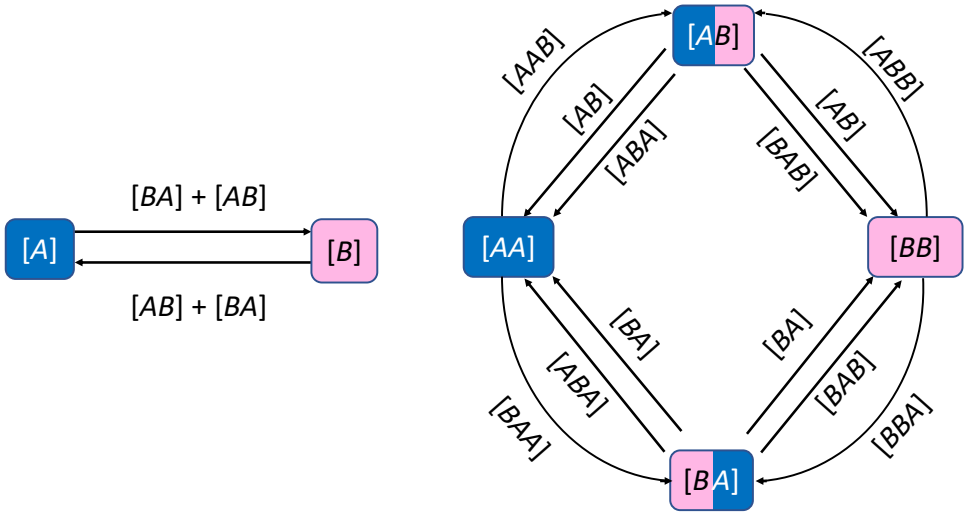


Figure 2.1: Voter model flux diagram for asynchronous, edge-based updates. We show the moments with rounded rectangles and draw arrows to indicate flows between them. We label the arrows with the transition rate that are associated with the corresponding creation–destruction processes. For example, the arrow from $[AA]$ to $[AB]$ with label $[AAB]$ corresponds to the destruction of an AA-pair and creation of an AB-pair via an AAB-triple in which the node in the third slot (which is in state B) causes the node in the middle slot to switch from state A to state B.

kth-order moment involves $(k+1)$th-order moments. The full system of moment equations is thus infinite-dimensional. To make such a system tractable for analytical and numerical techniques, it is often necessary to close the moment equations by approximating higher-order moments in terms of lower-order moments [Kue16].

For the system in Equation (2.3), the simplest (nontrivial) moment closure is by pair approximations, in which we approximate 2nd-order momenets in terms of 1st-order and 0th-order moments [KRM97, DVB14, Kue16]. One builds such an approximation based on assumptions about the system, and different assumptions lead to different approximations. As an example, let us write an approximation for the moment $[ABA]$ following the method

of Ref. [DVB14]. To estimate the number of ABA-triples, we start with the number of AB-pairs. We then find the expected number of nodes in state A that are attached to the B-node in an AB-pair. Let $\langle q_{B,AB} \rangle$ be the *mean excess degree* of B-nodes in AB-pair. Th mean excess degree $q_{Y,XY}$ is the expected number of additional edges incident to the Y-node in an XY-pair. If we assume that AB-pairs are uncorrelated (except for the effect on degree), then the probability that the additional edge is a BA-pair is approximately $\frac{[BA]}{[BA]+[BB]}$. This gives the pair approximation

$$[ABA] \approx [AB]\langle q_{B,AB} \rangle \frac{[BA]}{[BA]+[BB]} = c_B \frac{[AB]^2}{[B]}\,, \tag{2.4}$$

where c_B is the ratio of $\langle q_{B,AB} \rangle$ and the mean degree of the B-nodes. The parameter c_B is generally unknown unless the graph has certain properties (such as degree regularity). Demirel et al. report that $c_B = 1$ for ER networks [DVB14]. Substituting the pair approximation from Equation (2.4) into Equation (2.3), we arrive at the following closed system of differential equations:

$$\begin{aligned}
\frac{d}{dt}[AA] &= [AB] - \frac{[AA][AB]}{[A]} + \frac{[AB]^2}{[B]}\,, \\
\frac{d}{dt}[BB] &= [AB] - \frac{[BB][BA]}{[B]} + \frac{[AB]^2}{[A]} \\
\frac{d}{dt}2[AB] &= -2[AB] - \frac{[AB]^2}{[B]} - \frac{[AB]^2}{[A]} + \frac{[AA][AB]}{[A]} + \frac{[BB][BA]}{[B]}\,.
\end{aligned} \tag{2.5}$$

One can then study Equation (2.5) to gain an approximate understanding of the temporal evolution of the moments. For example, we can can compute the steady state of $[AA]$ to get

$$\frac{d}{dt}[AA] = 0 \implies [AB] = \frac{(\langle k \rangle - 1)}{N}[A](N - [A])\,. \tag{2.6}$$

This suggests that the asynchronous edge-based voter model has a quasi-stationary (i.e., quasi-steady-state) distribution [DGL12, Kue16], so after an initial transient, the system slowly evolves along the manifold that is described by Equation (2.6).

The type of moment approximation in Equation (2.6) is more precisely called a *homogeneous* moment approximation, because it involves only state-specified motif information [DVB14]. In addition to this information, one can also take into account the degrees

of each of the nodes in a motif; this yields a *heterogeneous* moment approximation. The difference between homogeneous and heterogeneous moment approximations is in their level of granularity. Two methods closely related to heterogeneous moment approximations are the improved compartmental formalism [NDB09, MNH10] and approximate master equations [Gle11, Gle13, DGL12].

2.4 Extensions of Voter Models

Voter models are undeniably a highly-idealized model for opinion dynamics.[5] However, their simplicity and tractability (from a mathematical perspective) make them an excellent foundation upon which one can build more intricate (and possibly more realistic) opinion models. In many cases, one formulates an extension of voter models with mechanisms, rules, or structures that are based on sociological theory, real-word phenomena, or current events. In Table 2.1, we list some notable extensions. For review articles on some extensions, see Refs. [CFL09, Red19, JS19].

The voter model extensions that we list in Table 2.1 are only some of the extensions that researchers have studied. One can combine or mix together the extensions to produce further extensions (e.g., threshold q-voter models [VA18] and q-voter model with zealots [Mob15]). One particularly interesting family of extensions are coevolving voter models, which we describe in detail in Section 2.5.

2.5 Coevolving Voter Models

We now study a popular extension of voter models, called *coevolving voter models* (CVM). that incorporates adaptive networks. In CVMs, node opinions (the node states) coevolve with network structure [PG16]. (These models are also sometimes called "adaptive voter models".) Coevolving voter models combine the classical framework of voter models [CS73, HL75, Cox89], in which individuals update their opinions based on their neighbors' opinions,

[5]The question of how well voter models can model voters was pursued in Ref. [FSR14].

Extension	References
contrarian voter models (similar to alternation processes)	[CS73, Mas13, DBZ13]
dissonant voter models	[Mat77]
biased voter models	[Fer90]
threshold voter models	[CD91, ALM92]
noisy voter models	[GM95, FSR14]
Axelrod models	[Axe97, Red19]
zealot voter models	[Mob03, HD19]
coevolving voter models	[HN06, ZG06, VES08]
vacillating voter models	[LR07]
voter models with heterozygosity selection	[SS08]
non-conservative voter models	[LR08]
q-voter models	[CMP09]
latent voter models	[LSB09, HD18]
partisan/stubborn voter models	[MGR10, YOA13]
rebellious voter models	[SV10]
concealed voter models	[GOG18, GTG19]

Table 2.1: Some notable extensions of voter models and relevant references.

with an evolving network structure (in which individuals change their relationships with other individuals in response to their opinions [EB02, BR00, GDB06, GB07]). A coevolving voter model consists of a network of individuals, two or more opinion states, and a rule (e.g., in the form of a stochastic process) for updating both the network and the states of its nodes or edges. We restrict our attention to a binary opinion space, but one can also study models with more than two opinions [HN06, SMD13]. There is also an interesting CVM that includes states both on the nodes and on the edges [SST19].

One of the motivations for studying coevolving voter models is their fascinating dynamics,

and scholars have analyzed them using approaches from subjects like dynamical systems, statistical physics, partial differential equations, and probability theory [HN06, DGL12, DVB14, SDH14, CM20]. A particularly interesting aspect of the dynamics of CVMs is the apparent phase transition that can occur in "linear" CVMs. In this context, "linear" refers to the linearity of the rewiring probability function $f_{\mathrm{r}}(x)$ and adoption probability function $f_{\mathrm{a}}(x)$: A node with a fraction x of disagreeing neighbors has a probability $f_{\mathrm{r}}(x) = \alpha x$ (for some parameter α) to rewire and a probability $f_{\mathrm{r}}(x) = (1 - \alpha)x$ to adopt. The parameter α is sometimes called the "rewiring rate", and $(1 - \alpha)$ is sometimes called the "adoption rate".

In some variants of linear CVMs (such as those in Refs. [HN06, DGL12]), there appears to be a phase transition as one increases the adoption rate (and thus decreases the rewiring rate) from a regime of "rapid disintegration" to a regime of "prolonged persistence of the dynamics". In the former regime, a network separates into components so quickly that the densities of the opinion states are unable to change significantly. In the latter regime, the system progresses slowly toward a steady state in which almost all nodes has the same opinion state. Basu and Sly presented a mathematically rigorous proof of the existence of phase transitions for two variants of linear CVMs on dense Erdős–Rényi (ER) networks (using the $G(N, p)$ model with $p = 1/2$) [BS17]. However, it has not yet been proven that a phase transition occurs for sparse networks or other classes of dense networks. In Section 2.6, we review some of the existing computational results for linear CVMs on sparse ER networks and present computational results for these CVMs on sparse networks that we construct from a stochastic block model (SBM). We demonstrate that, although these linear CVMs modify the structure of their associated network, their steady-state behavior appears to be insensitive to the examined initial network structures in a sense that we make precise in Section 2.6.

We examine three different rewiring schemes, which we illustrate in Figure 2.2. First, we explore a "rewire-to-random" (RTR) scheme, in which nodes sever connections (specifically, discordant edges) with disagreeing neighbors and replace them with new connections to nodes in a way that is agnostic to opinion states. We then explore a "rewire-to-same" (RTS) scheme, in which nodes sever connections with disagreeing neighbors and replace

them with new connections to nodes that share their opinion state. Finally, we explore a "rewire-to-none" (RTN) scheme, in which nodes sever connections with disagreeing neighbors without forming any new connections to other nodes. This third type of rewiring models behavior on social media in which individuals "unfriend" (or "unfollow") someone after a disagreement [Bod16, YBR17, ZSS17]. A fascinating feature of linear CVMs is that the choice of rewiring scheme has a dramatic impact on their steady-state properties [DGL12]. However, for the nonlinear CVM that we introduce in Chapter 3, the choice between the above rewiring schemes does not seem to have major qualitative effects on their steady-state behavior. Although we study each of the rewiring schemes separately, it is possible to combine the heterophilic aspects of RTR and the homophilic aspects of RTS into a single CVM [KH08].

For CVMs, as with voter models on time-independent networks, the set V of nodes remains fixed as a system evolves, but now both the set E of edges and the function S can update. When we refer to the edge set at a certain time, we use the notation $E(t)$ to explicitly indicate time-dependence, but otherwise we suppress explicit notation for time. As before, we use E_D [or $E_D(t)$ when specifying the time t explicitly] to denote the set

$$E_D = \{(u, v) \in E(t) : S[u] \neq S[v]\} \tag{2.7}$$

of discordant edges in $E(t)$.

2.6 Linear Coevolving Voter Models on Erdős–Rényi Networks

2.6.1 Linear Rewire-to-Random CVM on ER Networks

Using simulations, we examine the linear RTR-CVM that was studied in Ref. [DGL12]. This CVM is asynchronous; for each *elementary time step* $t = \{1, 2, \ldots\}$, we update the state of one node or rewire one edge. In an elementary time step, we select a discordant edge (i, j) uniformly at random from E_D. Equivalently, we can think of choosing the edges from E

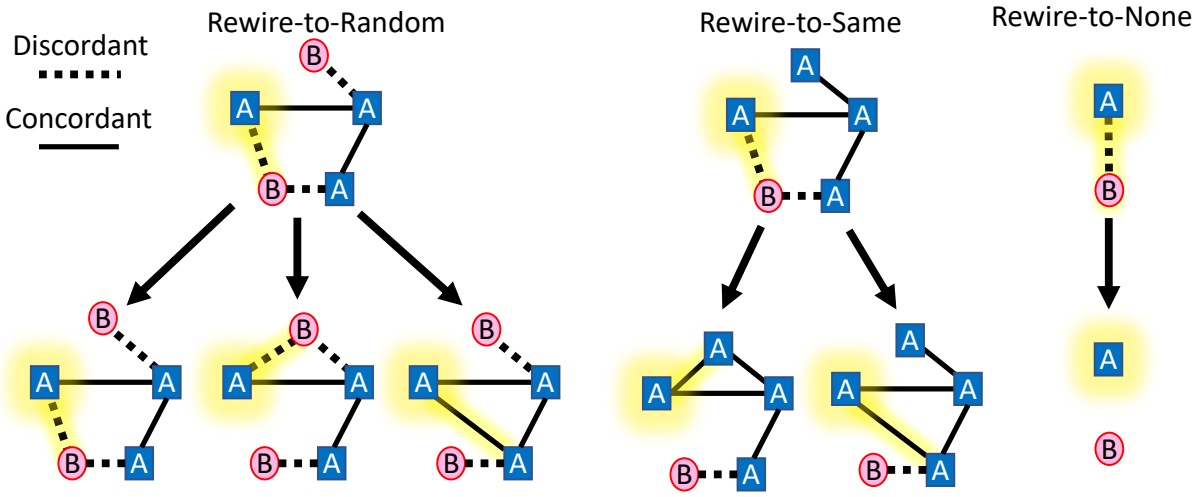


Figure 2.2: Illustration of three rewiring schemes for CVMs. The highlighted edges in the top part of the figure are the discordant edges that we will rewire, and the highlighted nodes are the focal nodes. The bottom part of the figure shows all possible rewiring outcomes for each scheme. (Left) Rewire-to-random scheme, in which the focal node either rewires to some node to which it is not adjacent or restores the original edge. (Center) Rewire-to-same scheme, in which the focal node rewires only to nodes that share its opinion state. (Right) Rewire-to-none scheme, in which we delete the discordant edge.

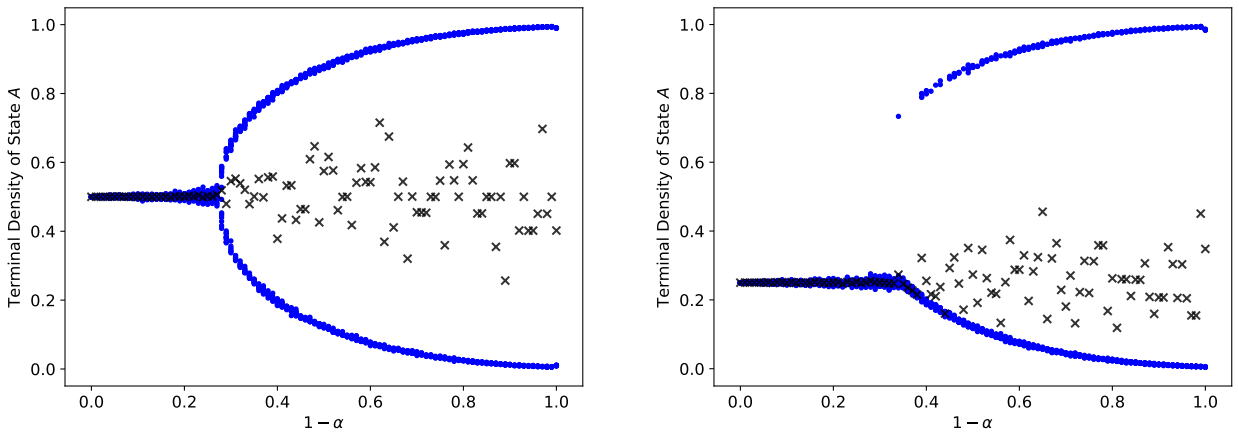
according to the probability mass function

$$f_E((i,j)) = \begin{cases} \frac{1}{|E_D|}\,, & (i,j) \in E_D \\ 0\,, & (i,j) \notin E_D\,. \end{cases}$$

The only effect that choosing uniformly at random from E_D, rather than uniformly at random from E, has on the dynamics is that we can skip steps in which nodes i and j already share the same state, as such steps do not affect the state of the system. This leads to a logarithmic speedup in the time to reach a steady state (compare this to the coupon-collector problem [Fel68]) and was called an "efficient version" of CVMs in Ref. [BDZ15]. We select one of the two nodes (which we can take to be the one with the label i without loss of generality) at random with equal probabilities to be the primary node; we take the other node (j) to be the secondary node. With probability α, node i performs a rewiring action; with probability $1-\alpha$, it adopts j's opinion. A rewiring action consists of deleting edge (i,j) and then uniformly randomly creating a new edge (i,k) where node k is chosen uniformly at random from the set $V \setminus \Gamma(i,E)$ of nodes that are not in the closed neighborhood $\Gamma(i,E)$ of i. Node j is available for selection.

Here and throughout this thesis, when we simulate a voter model, we let the system evolve until it terminates in an absorbing steady state (i.e., until there are no discordant edges). This yields a "terminal" state. We focus on studying properties of terminal states, so we are interested in what happens as $t \to \infty$. For example, we examine the terminal state densities $N_A(t)/N$ and $N_B(t)/N$ and the terminal minority-state density $\min\{N_A(t)/N, N_B(t)/N\}$.

In our first simulation, we seed the system using ER $G(N,p)$ networks with $N = 20,000$ nodes and an edge probability of $p = \frac{4}{N-1}$. The ER $G(N,p)$ random graph model generates a random graph on N nodes in which each pair of nodes is connected by an edge with probability p independent from each other [New18]. In Figure 2.3a, we initialize half of the nodes in state A and the other half in state B. In Figure 2.3b, we initialize $1/4$ of the nodes in state A and the other $3/4$ of nodes in state B. In Figure 2.3, we plot the terminal density of state A for 20 realizations of the simulations for each value of q. In this figure, we also show the means of terminal densities for each q as crosses ($\times$).



(a) $N_A(0)/N = 1/2$

(b) $N_A(0)/N = 1/4$

Figure 2.3: Terminal density of state A in simulations for the linear RTR-CVM from Ref. [DGL12] for $\alpha \in [0, 1]$ with a step size of $\Delta\alpha = 0.01$. For each value of α, we simulate 20 realizations. We seed each realization with a different ER network with $N = 20,000$ nodes and an edge probability of $p = \frac{4}{N-1}$. In (a), we initialize half of the nodes in state A and half of the nodes in state B. In (b), we initialize 1/4 of the nodes in state A and 3/4 of the nodes in state B. We plot individual realizations with blue dots and means over the 20 realizations with $\times$ symbols. Note that the horizontal axis is $(1 - \alpha)$.

In Figure 2.3a, when $1 - \alpha \lessapprox 0.25$, rewiring actions dominate. The fraction of nodes that terminate in state A is approximately constant, with a value of 0.5. For progressively larger values of $1 - \alpha$ (i.e., as we consider a progressively smaller rewiring rate α and hence a progressively larger adoption rate), rewiring and adoption actions begin competing and the plot appears to branch, with one branch decreasing to 0 and the other increasing to 1 as $1 - \alpha \to 1$. This illustrates that, by the time the system terminates, there are larger changes to the state densities of the system for progressively larger values of $1 - \alpha$. Because the system begins with $N_A(0)/N = N_B(0)/N = 1/2$, terminating along either branch (i.e., whether there is a positive or negative change for $N_A(t)/N$) is equally probable, as indicated by the values of the means of the terminal state-A densities.

When $1 - \alpha = 1$, no rewiring occurs, so isolated nodes do not change their opinion state. In our simulations in Figure 2.3a, the seed ER networks have an expected mean degree of

4. In our realizations, the seed networks have a largest connected component (LCC) that consists of approximately 98.2% of the nodes on average, and most of the other nodes have degree 0. When the system terminates and the nodes in the LCC settle in either state A or state B, approximately 0.9 % of nodes are in the other opinion state, because we initialize half of the nodes in each state and those nodes never update their states.

In Figure 2.3b, we observe behavior that is qualitatively similar to that in Figure 2.3a. For $1 - \alpha \lessapprox 0.35$, rewiring actions dominate. The fraction of nodes that terminate in state A is approximately constant, with a value of 0.25. For progressively larger values of $1 - \alpha$, rewiring and adoption actions begin competing and the plot appears to branch, with one branch decreasing to 0 (indicating a negative change to $N_A(t)/N$) and the other increasing to 1 (indicating a positive change to $N_A(t)/N$) as $1-\alpha \to 1$. Unlike in Figure 2.3a, we begin with $N_A(0)/N = 1/4$. We observe that approximately $1/4$ of the realizations terminate along the upper branch (i.e., with a positive change to $N_A(t)/N$) by examining the means of the terminal state-A densities. This result is expected based on random-walk theory. See Section 3.3.1 for details.

2.6.2 Linear Rewire-to-Same CVM on ER Networks

We now perform a brief exploration of the linear RTS-CVM that was studied in Ref. [DGL12]. The update rules for this model are identical to those described in Section 2.6.1, except for the rewiring action. Under this RTS scheme, a rewiring action consists of deleting edge (i, j) and then uniformly randomly creating a new edge (i, k) where node k is chosen uniformly at random from the set $\{k \in V \setminus \Gamma(i, E) : \ S[k] = S[i]\}$ of nodes that are in the same opinion state as i but not in the closed neighborhood of i. There is a boundary case to consider when this set is empty, which is not mentioned explicitly in Ref. [DGL12]. For the simulations here, we account for this by reforming the recently deleted edge (i, j). We discuss this boundary case of RTS schemes in detail and discuss some of the possible ways to resolve it in Section 3.4.

For our first simulation, we seed the system using ER $G(N, p)$ networks with $N = 20,000$

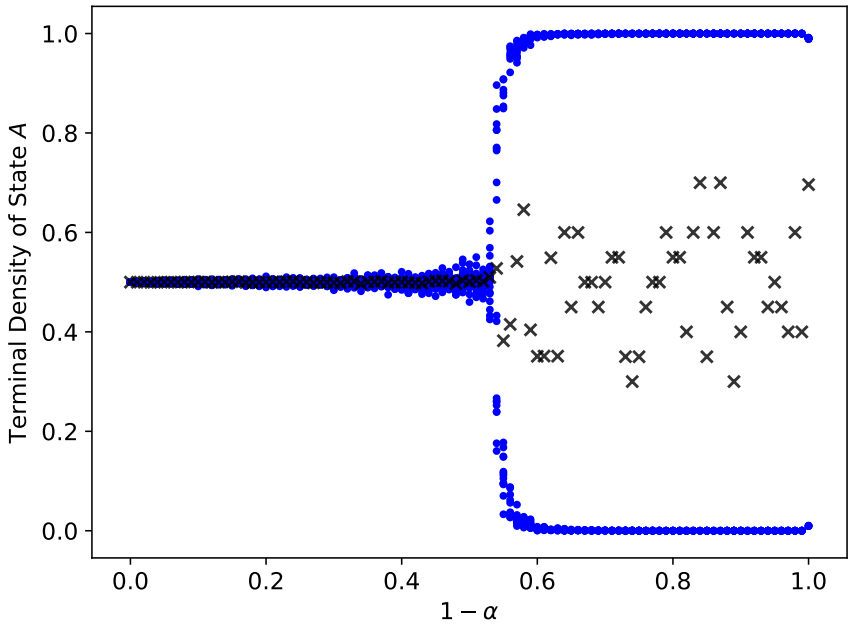


Figure 2.4: Terminal density of state A in simulations of the linear RTS-CVM from Ref. [DGL12] for $\alpha \in [0, 1]$ with a step size of $\Delta\alpha = 0.01$. For each value of α, we simulate 20 realizations. We seed each realization with a different ER network with $N = 20,000$ nodes, an edge probability of $p = \frac{4}{N-1}$, half of the nodes in state A, and half of the nodes in state B. We plot individual realizations with blue dots and means with $\times$ symbols.

nodes and an edge probability of $p = \frac{4}{N-1}$. We initialize half of the nodes in state A and the other half in state B. In Figure 2.4, we observe what appears to be a discontinuous phase transition for a critical value of α. For $1 - \alpha \lessapprox 0.57$, rewiring dominates and state densities do not change significantly. However, when $1 - \alpha \gtrapprox 0.57$, almost all nodes terminate in the same state. The system begins with $N_A(0)/N = N_B(0)/N = 1/2$, so it is equally probable for almost all nodes to terminate in state A or almost all nodes to terminate in state B. As we noted in Section 2.6.1, when $1 - \alpha = 1$, there are no rewiring actions, so isolated nodes do not change their opinion state.

2.6.3 Linear Rewire-to-None CVM on ER Networks

We now discuss some results from our simulations of linear CVMs with edge deletion, which we call rewire-to-none (RTN) to parallel the rewire-to-random and rewire-to-same terminology. As in the previous CVMs, this model involves picking a discordant edge uniformly at random from the set of discordant edges and choosing one of the incident nodes uniformly at random to be the primary node. With probability α, one then deletes the discordant edge or, with complementary probability $1 - \alpha$, one updates the primary node's opinion state. Although the number $|E(t)|$ of edges is not conserved, edge deletions occur at a fixed rate α, so the expected number of edges is $\mathbb{E}[|E(t)|] = |E(0)| - \alpha t$, which is valid until the system terminates, after which the number of edges is constant. A generalized version of this linear RTN-CVM was studied in Refs. [GZ06, ZG06].

In Figure 2.5, we plot the terminal minority-state density of our simulations. We seed the system with an ER $G(N, p)$ network with $N = 20,000$ nodes and an edge probability of $p = \frac{4}{N-1}$. Initially, there are approximately $20,000$ discordant edges. We compute an estimate for the terminal minority-state density in terms of α by assuming that the adoption mechanism does not significantly increase or decrease the number $N_{AB}(t)$ of discordant edges. Because we delete discordant edges at a constant rate α, we expect the system to terminate in approximately $\frac{E_D(0)}{\alpha}$ elementary time steps. During this time, the expected number of adoption actions is $\frac{(1-\alpha)E_D(0)}{\alpha}$. Each adoption action increases the number $N_A(t)$ of nodes in

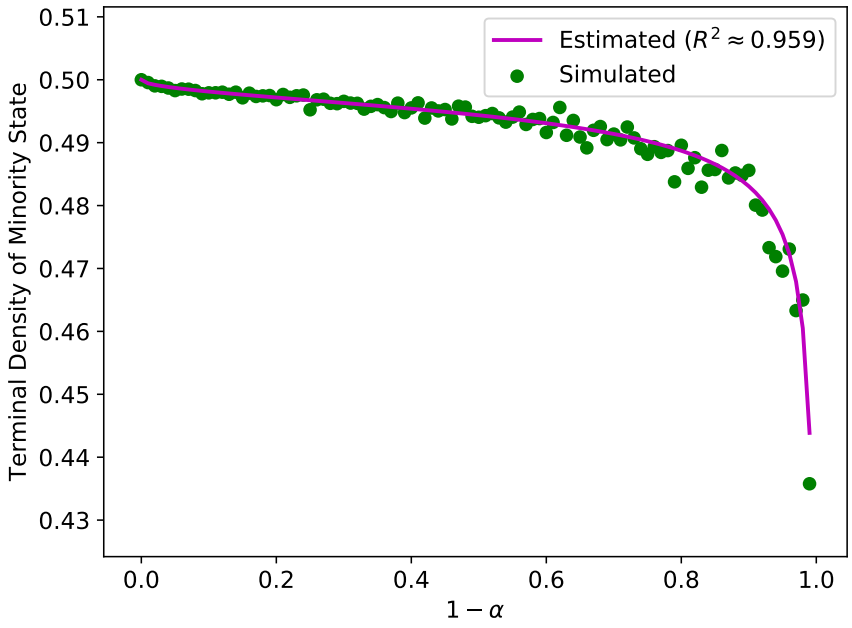


Figure 2.5: Terminal minority-state density in simulations of a linear CVM with edge deletion (i.e., using a rewire-to-none scheme) for $\alpha \in [0, 1]$ with a step size of $\Delta\alpha = 0.01$. For each value of α, we simulate 20 realizations. We seed each realization with a different ER network with $N = 20,000$ nodes and an edge probability of $p = \frac{4}{N-1}$. Half of the nodes start in state A, and the other half start in state B. Each green point is the mean over 20 realizations for a given value of α. The magenta curve is the estimated terminal minority-state density from Equation (2.8).

state A by 1 with probability 1/2 and decreases it by 1 with probability 1/2. We can thus think of $N_A(t)$ as a simple, symmetric 1D random walk during the steps in which adoption occurs. For a simple, symmetric 1D random walk that starts at the origin, the expected distance of the walker to the origin after n steps is $\sqrt{\frac{2n}{\pi}}$ [Fel68]. Therefore, after $\frac{(1-\alpha)E_D(0)}{\alpha}$ steps, we expect that $N_A(t)$ has either increased or decreased by $\frac{1}{N}\sqrt{\frac{2(1-\alpha)E_D(0)}{\pi\alpha}}$.

In Figure 2.5, we include a plot of our estimate

$$\frac{1}{2} - \frac{1}{N}\sqrt{\frac{2(1-\alpha)E_D(0)}{\pi\alpha}} \tag{2.8}$$

for the terminal minority-state density, where the 1/2 term comes from the initial densities of 1/2 and we subtract from this value because we are calculating the minority-state density. The plot illustrates that our estimate captures the behavior of the linear RTN-CVM accurately, with a coefficient of determination of $R^2 \approx 0.959$.

CHAPTER 3

Nonlinear Coevolving Voter Models

The following sections are adapted from an original paper [KP20] that I co-authored with my advisor Mason A. Porter.

3.1 Making a Coevolving Voter Model Nonlinear

In this chapter, we propose a simple mathematical model that combines nonlinear q-voter models and coevolving voter models (CVM) that couples the mechanisms of opinion formation and changing social connections. By taking a minimalist approach to modeling social dynamics, we are able to gain a deeper understanding of how these two mechanisms can give rise to phenomena that we observe in modern social discourse. For example, in our model, if an extreme opinion is held only by a small portion of a population but there is an "illusion" that the opinion is prevalent because of the social network structure, the extreme opinion can spread to everyone. We explore our new model numerically through Monte Carlo simulations on a variety of random networks and analytically through moment equations and mean-field approximations. We observe that our model behaves remarkably differently from prior CVMs. One of the most striking differences is that the choice of rewiring mechanism, which significantly affected steady-state dynamics in some linear models, has little qualitative impact on the steady-state behavior our model. Our model provides a basis for further studies of various important questions of social dynamics, especially the coevolution of opinion propagation and network structure. Our model may also be helpful for designing and testing intervention methods to curtail ideological extremism and opinion segregation in online social media.

For over a decade, linear CVMs have been a challenging, popular, and fruitful topic to study. From a practical perspective, however, nonlinearity appears to be a critical ingredient for opinion models. For example, Couzin et al. [CID11] successfully predicted the existence of novel collective behaviors of schooling fish using nonlinear adaptive-network opinion models. In our setting, individuals seek to achieve social harmony (in the form of having the same opinion state as all of their neighbors) by rewiring and adopting at rates that depend on the states of their neighbors in a nonlinear way. Each node is in one of two states, and neighboring nodes "agree" if they are in the same opinion state and "disagree" if they are in different opinion states. As in Chapter 2, we refer to edges between agreeing neighbors as *concordant* and edges between disagreeing neighbors as *discordant*. When updating, a node that is not in a local consensus (i.e., it disagrees with at least one neighbor) performs one of two actions with respect to a disagreeing neighbor: (1) it adopts the opinion of the disagreeing neighbor, causing other neighbors who had been in agreement with it to now disagree; or (2) it abandons the edge that connects it to the disagreeing neighbor and possibly forms a new connection. Similar to the CVM in Ref. [HN06], a node makes a random choice between options (1) and (2). However, unlike in that model, the probability that a node chooses a given option is not homogeneous; instead, it depends nonlinearly on the states of the node's neighbors. In our model, nodes conduct a local survey of all of their neighbors. Those who agree with a large fraction of their neighbors (i.e., those who "fit in") are more likely to remove the edge and possibly rewire than to adopt a new opinion (which can place them in a local minority opinion). Conversely, nodes that are in a local minority among their neighbors are more likely to adopt a neighbor's opinion (which can place them in a local majority opinion) than to remove a discordant edge and possibly rewire.

A node's local survey provides it with a sample view of a population. The sample is susceptible to bias, as nodes survey only their neighbors in a network. Under certain conditions, the local surveys can accurately estimate global statistics, such as an opinion's popularity, which is equal to the fraction of nodes that hold that opinion. However, it is possible to configure systems such that the sampling bias leads nodes to construe globally popular opinions as locally unpopular, and vice versa. In some situations, we find in our

nonlinear CVM that when certain nodes' local surveys are so distorted that they perceive the minority opinion to instead be the majority opinion, almost every node eventually adopts the opinion that was initially unpopular. Consequently, such distorted sampling in local surveys, which we relate to recent work by Lerman et al. [LYW16] on what they called the "majority illusion", has important implications for the dynamics of our nonlinear CVM.

We examine our nonlinear CVM with three different rewiring schemes (see Figure 2.2 in Section 2.5 for a schematic). In Section 3.3, we explore a rewire-to-random (RTR) scheme, in which nodes sever discordant edges and replace them with new edges to nodes in a way that is agnostic to opinion states. In Section 3.4, we explore a rewire-to-same (RTS) scheme, in which nodes sever discordant edges and replace them with new edges to nodes that share their opinion state. In Section 3.5, we explore a rewire-to-none (RTN) scheme, in which nodes sever discordant edges without forming any new edges to other nodes. This third type of rewiring models behavior on social media in which individuals unfriend (or unfollow) someone after a disagreement [Bod16, YBR17, ZSS17]. In Section 3.6, we examine a linear CVM with the RTR, RTS, and RTN rewiring schemes on SBM networks to facilitate comparisons with our nonlinear CVM (see Section 2.5 for additional experiments of a linear CVM on ER networks). In Section 3.7, we summarize our results and discuss possibilities for future work. We find that our nonlinear CVM has several features, such as a strong dependency on network structure, that distinguish it from previously studied linear CVMs.

3.2 Nonlinear Coevolvinvg Voter Model

We now formally introduce nonlinear CVM. In all variants of our nonlinear CVM, we consider binary opinion states $O = \{A, B\}$ and networks with finitely many nodes and edges. The space of *system states* is given by the space of triples (V, E, S), where V is the set of nodes, $N := |V|$ denotes the number of nodes, E is the set of edges (which represent changing, pairwise, undirected, unweighted ties between individuals), and $S : V \to O$ is a function that records the opinion state of each node. For simplicity, we disallow multi-edges and self-edges. In Section 3.4, we discuss some situations in which we can relax these stipulations.

In our nonlinear CVM, we find that it is simpler to define update rules in an asynchronous manner (see Section 2.2).[1] We use "edge-based" updates to select the updating node (see Section 2.2). We perform one update for each elementary time step $t = \{1, 2, \ldots\}$. In an elementary time step, we select a discordant edge (i, j) uniformly at random from E_D. Alternatively, one can think of choosing the edges from E according to the probability mass function

$$f_E((i,j)) = \begin{cases} \frac{1}{|E_D|}\,, & (i,j) \in E_D\,, \\ 0\,, & (i,j) \notin E_D\,. \end{cases}$$

We then select one of the two nodes (which we can take to be the one with the label i without loss of generality) at random with equal probabilities to be the primary node; we take the other node (j) to be the secondary node. After selecting the primary node i, it takes a local survey of it neighbors. We measure the result of the local survey by calculating $\sigma_i := s_i/k_i$, where $s_i = |\{j : (i,j) \in E\,, S[i] = S[j]\}|$ and k_i is the degree of node i. We also define $\bar{s_i} := k_i - s_i$, which counts the number of discordant edges that are incident to node i. Therefore, σ_i is the fraction of neighbors of node i that agree with i. For node i to be selected for updating, it needs to have at least one disagreeing neighbor (i.e., at least one discordant incident edge), so $\sigma_i \in [0, 1)$. Note that σ_i is not defined for isolated nodes; however, because our CVM is edge-based, it is not possible to select an isolated node for updating.

3.3 Rewire-to-Random Formulation

3.3.1 Model and Associated Discussion

Our nonlinear CVM has a parameter q that is akin to a parameter in nonlinear q-voter models [CMP09]. With probability σ_i^q, node i performs a rewiring action, in which it deletes its edge to the chosen secondary node and then uniformly randomly forms a new edge to a

[1] We are not aware of any studies of CVMs with synchronous updating. One challenge in formulating such a model is establishing a protocol for the situation in which two adjacent nodes select each other for updating and both simultaneously attempt to rewire the same edge.

node that is not currently one of its neighbors. This type of rewiring scheme, which yields an edge between node i and any node (irrespective of its opinion state), is known as a rewire-to-random (RTR) strategy [DGL12]. With complementary probability $1 - \sigma_i^q$, node i takes an adoption action, in which it adopts the opinion state of the chosen secondary node. We then repeat this process until there are no discordant edges in the network. If the system reaches a system state with no discordant edges at time t^*, then the dynamics reach a steady state, and we say that the system terminates at that system state, such that the system remains in that system state for all $t \geq t^*$. Both the rewiring and adoption actions conserve the number of edges, so $|E(t)|$ is constant in time.[2]

When q is a positive integer, one can interpret our updating process above as the primary node i randomly selecting a panel of q of its neighbors with repetition allowed. The secondary node j need not be in the panel. If any member of the panel is in a different opinion state from the primary node i, the latter undertakes an adoption action. Therefore, only if the panel and the primary node i are all in the same opinion state does node i perform a rewiring action. We summarize the rewiring process in Algorithm 1 and give a schematic of the process in one elementary time step in Figure 3.1.

Because $\sigma_i \in [0, 1)$ for nodes that can update, it follows that in the $q \to \infty$ limit, we recover a voter model with only adoption (and no rewiring). However, as $q \to 0^+$, we do *not* recover a model with only rewiring, because for some nodes i, it can be the case that $\sigma_i = 0$ if all of node i's incident edges are discordant; in that case, node i performs an adoption action for all $q > 0$. We also perform simulations with $q = 0$; for these simulations, we take 0^0 to be 1 to recover a pure rewiring model.

The nonlinear CVM that we just described is an absorbing Markov chain. The absorbing system states are those in which a network has no discordant edges. Such a situation occurs when each connected component of the network is in a consensus, but it does not necessarily

[2]Recently, Min and San Miguel [MS17] introduced a nonlinear CVM that also incorporates such a parameter q. Under their direct node-based rules, once one selects a node i, it is with probability $(1-\sigma_i)^q$ that node i performs any update at all, and then a separate parameter p determines the relative probabilities of rewiring and adoption actions. This differs from our edge-based rules; under our rules, we select a node i to update, and σ_i^q determines the relative probabilities of rewiring and adoption actions.

require all components to achieve a consensus with the same opinion state. There has been significant prior work on noisy voter models [GM95, CTS16], and some recent work has studied noisy CVMs with random opinion-state mutations [JXC13]. In these systems, in addition to the rewiring and adoption updates, there is also a mechanism that alters the opinion states of nodes according to some random process. Incorporating such noise yields a Markov chain that no longer is absorbing, because the system can exit a system state with component-wise consensus through random creation of discordant edges. The resulting models are ergodic, so one can approximate the non-Markovian second-order (and higher-order) moment terms (see Equation (2.2)) using Markovian terms [CM20].

Algorithm 1 Nonlinear Rewire-to-Random (RTR) Coevolving Voter Model

procedure FITTINGINVM(V, E, S, q) ▷ Input: Initial network and opinion states
  $E_D \leftarrow Discordant(V, E, S)$; $t \leftarrow 0$; $Record(V, E, S, t)$
  while $E_D \neq \emptyset$ **do** ▷ While there are disagreeing neighbors
    $(i, j) \leftarrow RandomChoice(E_D)$
    $PrimaryNode, SecondaryNode \leftarrow RandomPermutation(i, j)$
    $\sigma \leftarrow LocalVote(PrimaryNode, V, E, S)$
    $u \leftarrow \mathcal{U}(0, 1)$
    if $u \leq \sigma^q$ **then** ▷ Rewire
      $E.remove(PrimaryNode, SecondaryNode)$
      $NewNeighbor \leftarrow RandomChoice(V \setminus \Gamma(PrimaryNode, E))$
      $E.add(PrimaryNode, NewNeighbor)$
    else ▷ Adopt
      $S[PrimaryNode] \leftarrow S[SecondaryNode]$
    $E_D \leftarrow Discordant(V, E, S)$; $t \leftarrow t + 1$; $Record(V, E, S, t)$

We recall our notation for counts of state-specified motifs (see Section 2.3):

1. $N_X(t) = |\{i \in V : S^t[i] = X\}|$, the number of nodes i at time t that are in state $X \in O = \{A, B\}$ and

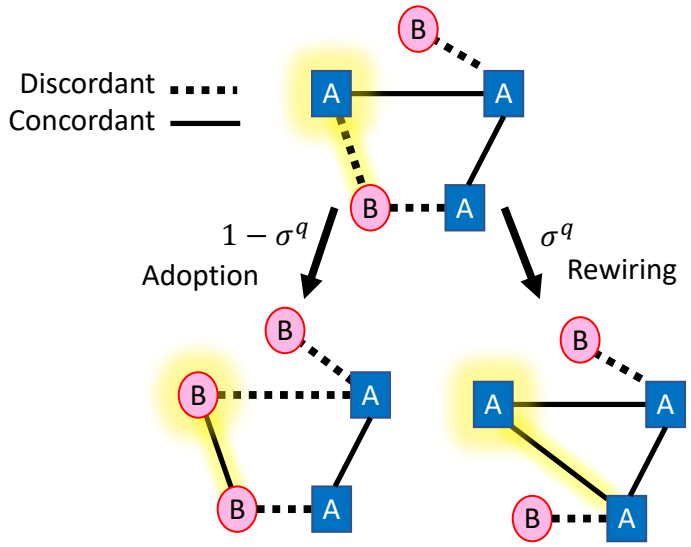


Figure 3.1: Schematic representation of an elementary time step in our nonlinear CVM. We highlight the selected discordant edge and the primary node. With probability $1-\sigma^q$, the primary node adopts the opinion state of its neighbor that is incident to this discordant edge. With complementary probability σ^q, the primary node performs a rewiring action. Under a RTR scheme, there are three possible outcomes of rewiring (see Figure 2.2). Each of the possible rewiring outcomes is equally probable. (For the rewire-to-same scheme, the depicted outcome is the only possible one, so it occurs with probability 1 if there is a rewiring action.) The value of σ in this example is $1/2$.

2. $N_{XY}(t) = |\{(i,j) \in V \times V : (i,j) \in E(t)\,, S^t[i] = X\,, S^t[j] = Y\}|$, the number of node pairs (i,j) at time t in which i and j are adjacent, node i is in state $X \in O = \{A, B\}$, and node j is in state $Y \in O$.

Because the node set V is fixed in our model, we also make use of state densities $N_X(t)/N$, which give the fraction of nodes in a state X.

We compare our nonlinear RTR-CVM (Algorithm 1) to the RTR-CVM that we studied in Section 2.6.1. Although there are many possible variants, we henceforth write "the linear RTR-CVM" when referring to this specific model. In the linear RTR-CVM, the system updates in a way that is similar to our nonlinear RTR-CVM, except that the probability for which action to take is given by a parameter $\alpha \in [0,1]$. With probability α, a node performs a rewiring action; with probability $1-\alpha$, it adopts a neighbor's opinion. Therefore, in the linear RTR-CVM, until $E_D = \emptyset$, the count $N_A(t)$ increases by 1 with probability $\frac{1-\alpha}{2}$, decreases by 1 with probability $\frac{1-\alpha}{2}$, and does not change with probability α in each elementary time step. If we take $\alpha = 0$ (entailing that the network topology never changes) and begin with a finite, connected network (V, E), then $N_A(t)$ behaves equivalently to a simple, symmetric, one-dimensional (1D) random walk[3] (i.e., the two possibilities each have a probability of 1/2) [MPL17] on the integers with boundary $\{0, N\}$. This implies that for any finite, connected network, the probability at time t that opinion state A eventually becomes the consensus opinion state is $\frac{N_A(t)}{N}$. For $\alpha > 0$, the quantity $N_A(t)$ behaves like a symmetric, 1D random walk with step sizes $+1$ and -1 that each occur with equal probability $\frac{1-\alpha}{2}$ and a zero-move step that occurs with probability α.

In the linear RTR-CVM, it is equivalent to take the view that one is choosing the type of action (rewire or adoption) *before* choosing which node of the selected edge is the primary one. This makes it clearer that even when the number of nodes in state A is not equal to the number of nodes in state B (i.e., when $N_A(t) \neq N_B(t)$), a rewiring action causes the number

[3]In a 1D random walk, "simple" refers to the rule that, at each time step, a walker must move by either $+1$ or -1; "symmetric" refers to these outcomes being equally probable. In a symmetric random walk with possible moves of $+1$, -1, and 0, one requires only that the first two possibilities are equiprobable to each other.

$N_{AB}(t)$ of discordant edges to decrease by $1/2$ in expectation, regardless of the system state. By contrast, the effect of an adoption on the number $N_{AB}(t)$ of discordant edges does depend on the system state, and it is possible for an adoption to increase $N_{AB}(t)$ in expectation. As an extreme case, consider a star network with a hub node in state A and k leaf nodes, and suppose that one node is in state B but all others are in state A. An update step is guaranteed to select the network's single discordant edge; if an adoption occurs, $N_{AB}(t)$ increases by $\frac{k-3}{2}$ in expectation. However, for a system on a degree-regular network (i.e., a network in which each node has the same degree) that satisfies the conditions

$$\begin{aligned} \sigma_i &\approx N_A(t)/N \text{ for all nodes } i \text{ in state } A\,, \\ \sigma_j &\approx N_B(t)/N \text{ for all nodes } j \text{ in state } B\,, \end{aligned} \tag{3.1}$$

an adoption causes $N_{AB}(t)$ to decreases by 1 in expectation. We use the term *locally well-mixed* for a system on any network that satisfies the conditions in Equation (3.1), which entails that there are no correlations between the opinion states of nodes and the network topology. In a recent paper, Lee et al. [LKW19] defined a related quantity called *social perception bias* that measures the ratio of a node's perception of the fraction of nodes whose opinion state is in the minority to the true fraction of nodes in the minority, where a value of 1 implies perfect perception of the frequency of the minority state in a network. Using this terminology, one can alternatively characterize a system as locally well-mixed using the condition that all nodes have a social perception bias that is close to 1. One can give a mathematically precise definition of locally well-mixed in the limit that the number N of nodes becomes infinite. Specifically, a system is *locally well-mixed* if almost all nodes have a social perception bias of $1 - o(1)$ as $N \to \infty$.

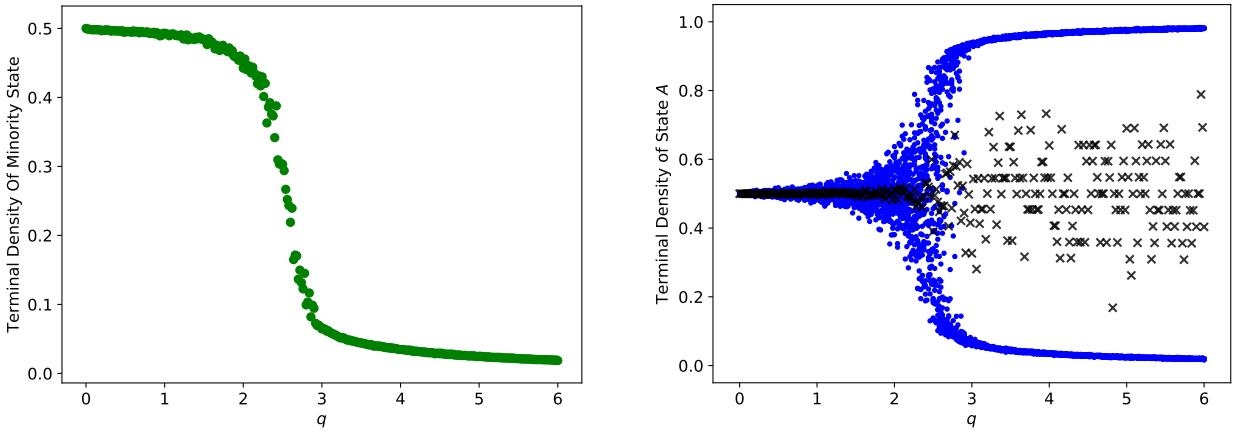
In our nonlinear CVM, there is no longer a symmetry between the two nodes that are incident to the same discordant edge (i, j). The local survey σ_i of node i can differ from the local survey σ_j of node j, so the probabilities of which action (rewiring or adoption) occurs depend on which node is the primary one. That is, in an update that involves node i in state A and node j in state B (i.e., after one selects the discordant edge (i, j), but before selecting which node is the primary one), $N_A(t)$ either (1) increases by 1 with probability $\dfrac{(1-\sigma_j^q)}{2}$,

corresponding to node j adopting node i's state; (2) decreases by 1 with probability $\frac{(1-\sigma_i^q)}{2}$, corresponding to node i adopting node j's state; or (3) remains the same with probability $\frac{\sigma_i^q + \sigma_j^q}{2}$, corresponding to either node i rewiring or node j rewiring. Therefore, although we can still view $N_A(t)$ as a 1D random walk, it is no longer symmetric, because the step probabilities can differ from each other.

The effect of this asymmetry on $N_{AB}(t)$ is more subtle. Consider a locally well-mixed system on a connected network. We also assume that one of the states, which we take to be B without loss of generality, is the majority [so $N_B(t) > N_A(t)$]. In an elementary time step, suppose that we select the discordant edge (i, j) with node i in state A and node j in state B. The local surveys then satisfy $\sigma_i < 1/2 < \sigma_j$. When $q = 1$, this implies that if i is the primary node, it is more likely to adopt than to rewire. If node i adopts state B, then $N_{AB}(t)$ decreases by more than 1 in expectation, because more of i's neighbors are in state B than in state A. If node j is the primary node, it is more likely to rewire than to adopt. If node j rewires to a node that we choose uniformly at random, then $N_{AB}(t)$ decreases by $N_B(t)/N > 1/2$ in expectation. Overall, we observe that the number $N_{AB}(t)$ of discordant edges decreases more rapidly in our nonlinear CVM than it does in the linear RTR-CVM under locally well-mixed conditions. Additionally, our nonlinear CVM has rather different dynamics when it is locally well-mixed than when it is not locally well-mixed. In Section 3.3.4, we explore how to construct systems with correlations between nodes' opinion states and network topology, and we investigate how their dynamics differ from situations in which a system is locally well-mixed.

3.3.2 Simulations on Erdős–Rényi Networks

We begin exploring our nonlinear RTR-CVM by simulating it on Erdős–Rényi (ER) $G(N, p)$ networks. We seed each realization with an ER network with $N = 50,000$ nodes, half of which begin in state A and the other half of which begin in state B. The edge probability p is $\frac{4}{N-1}$, so the mean degree is $\langle k \rangle \approx 4$. As we did in Section 2.6.1, we focus on studying properties of terminal states, so we are interested in what happens as $t \to \infty$. For example,



(a) Minority State

(b) State A

Figure 3.2: Terminal density for (a) the minority state and (b) state A in RTR simulations (see Algorithm 1) of our nonlinear CVM for $q \in [0, 6]$ with discretization $\Delta q = 0.02$. For each value of q, we simulate 20 realizations. We seed each realization with a different ER network with $N = 50,000$ nodes and edge probability $p = \frac{4}{N-1}$; we initialize half of the nodes in state A and the other half in state B. In panel (a), each green point is the mean over the 20 realizations of $\min\{N_A(t)/N, N_B(t)/N\}$ at the termination of the simulation. In panel (b), each small blue dot is $N_A(t)/N$ at the termination of the simulation. Each cross ($\times$) in (b) is the mean over the 20 realizations of $N_A(t)/N$ at the termination of the simulation.

we examine the terminal state densities $N_A(t)/N$ and $N_B(t)/N$ and the terminal minority-state density $\min\{N_A(t)/N, N_B(t)/N\}$.

In Figure 3.2a, we plot the terminal minority-state density from simulations for a range of values of q with discretization $\Delta q = 0.02$. When $q < 1$, we observe that the minority-state density is approximately 0.5. This implies that the network is *fragmented*, in the sense that it is separated into multiple components that are disconnected from one another, such that all nodes of a component are in the same opinion state. This behavior is similar to the fragmentation that was observed in Ref. [DGL12] for the linear RTR-CVM with sufficiently large rewiring rates. We interpret such fragmentation as individuals segregating into isolated communities, in which no pair of disagreeing individuals are neighbors of each other. This typically occurs when rewiring actions dominate the system, in the sense that the rewiring actions resolve most disagreements so the densities $N_X(t)/N$ do not change significantly. As we consider progressively larger values of q between 1 and 3.5, we observe a smooth transition (with an inverted 'S' shape) in the terminal minority-state density from approximately 0.5 to approximately 0.03. The small terminal minority-state densities for larger values of q suggest that adoption actions dominate the system, leading to large changes in the state densities. For expository convenience, we say that "almost every" node ends up in state X if 90 % or more of the nodes are in state X when a system terminates by reaching an absorbing state. (In these scenarios, the percentage of nodes in state X is often much larger than 90%, but we use this cautious phrasing because of particular examples.) Initially, $N_A(0)/N = N_B(0)/N = 1/2$. However, by the end of a simulation, one of the opinion states dominates, in the sense that $\max\{N_A(t)/N, N_B(t)/N\} \approx 1$. The other nearly vanishes, so $\min\{N_A(t)/N, N_B(t)/N\} \approx 0$, as one can see in Figure 3.2a.

In Figure 3.2b, we plot the terminal density of state A from simulations for a range of values of q. In this plot, each of the small blue dots is the terminal value of $N_A(t)/N$ for one of the 20 realizations that we simulate for each value of q. The crosses ($\times$) indicate the means over these realizations of the terminal quantity of $N_A(t)/N$. In the "branching" of the data points in this figure, we observe what appears to be a transition between a fragmentation regime (with no significant changes to state densities) and a regime with competition between

the adoption and rewiring mechanisms. Because our initial state densities are $N_A(0)/N = N_B(0)/N = 1/2$ in these simulations, it is equally likely for almost every node to terminate in state A as it is for almost every node to terminate in state B. We confirm this result in the means of the terminal state-A densities that we plot in Figure 3.2b.

With respect to coarse qualitative behavior, both our nonlinear RTR-CVM and the linear RTR-CVM of Ref. [DGL12] have a regime — when $q \lessapprox 1$ for our model, and when $1-\alpha \lessapprox 0.2$ for the linear RTR-CVM — in which rewiring dominates the system, as suggested by terminal minority-state densities that are close to the starting densities of 0.5. However, outside this regime, the two models differ significantly, as one can see by comparing Figure 3.2b with Figure 2.3a. The linear RTR-CVM appears to have a continuous (but non-smooth) phase transition from the rewiring-dominated regime to a regime in which rewiring and adoption are competitive [DGL12, SDH14, CM20]. In our nonlinear RTR-CVM, there seems to be a smooth transition between a regime in which rewiring dominates and a regime in which adoption dominates.

3.3.3 Approximations

Although mean-field approximations have been unable thus far to produce precise quantitative results in previous work on CVMs, they have been useful for exploring some of their qualitative behavior [KH08, PC09, BG11, GMW12, NKB08]. Writing a mean-field approximation of our model will require developing a mean-field analog of σ_i. We consider a state-heterogeneous mean, so we average the local surveys of nodes in state A and separately average the local surveys of nodes in state B. First, we consider the unweighted mean

$$\overline{\sigma_A} = \frac{\sum_{i:S[i]=A,\, k_i \neq 0} \sigma_i}{\sum_{i:S[i]=A,\, k_i \neq 0} 1} \,. \tag{3.2}$$

The unweighted mean $\overline{\sigma_A}$ is useful, because it has a simple interpretation and is well-approximated [4] by $\frac{N_{AA}(t)}{N_{AA}(t)+N_{AB}(t)}$. We will use the unweighted mean $\overline{\sigma_A}$ in Section 3.3.4.3, but it is not suitable as a mean-field approximation of our nonlinear RTR-CVM. In edge-based models such as ours and the linear RTR-CVM of Ref. [DGL12], nodes are not equally likely

[4] This approximation is exact for non-empty, degree-regular networks with at least one node in state A.

to be selected for an update. Instead, the probability to select a node increases with its number of neighbors in the other opinion state, as nodes with more such neighbors contribute more edges to the set E_D of discordant edges (from which we sample uniformly). Therefore, a suitable mean-field analog of σ_i should weight each node based on its probability of being selected. The probability that we select a particular node i to be the primary node in an update is

$$\frac{1}{2}\frac{k_i - s_i}{N_{AB}(t)} = \frac{1}{2}\frac{\bar{s_i}}{N_{AB}(t)}\,, \tag{3.3}$$

where $\bar{s_i}$ is the number of discordant edges that are incident to node i and the $1/2$ accounts for the random choice between the primary and secondary nodes. Therefore, our mean-field analogs of σ_i, which we denote by σ_X for $X \in \{A, B\}$, are

$$\sigma_A = \sum_{\{i:S[i]=A\,,\,k_i\neq 0\}} \sigma_i \frac{1}{2}\frac{k_i - s_i}{N_{AB}(t)} \tag{3.4}$$

$$= \frac{1}{2N_{AB}(t)}\left(N_{AA}(t) - \sum_{\{i:S[i]=A\,,\,k_i\neq 0\}} \frac{s_i^2}{k_i}\right)\,, \tag{3.5}$$

$$\sigma_B = \frac{1}{2N_{AB}(t)}\left(N_{BB}(t) - \sum_{\{i:S[i]=B\,,\,k_i\neq 0\}} \frac{s_i^2}{k_i}\right)\,. \tag{3.6}$$

The equation for the first moment, which we express in terms of state A, is

$$\frac{d[A]}{dt} = ([AB] + [BA])(\sigma_A^q - \sigma_B^q)\,. \tag{3.7}$$

Equation (3.7) arises from taking the difference of the "incoming rate" (i.e., nodes that change their state *to* A) minus the "outgoing rate" (i.e., nodes that change their state *from* A) to determine the net rate of opinion change into state A. Nodes in state B that are adjacent to a node in state A adopt state A at rate $1 - \sigma_B^q$. Nodes in state A that are adjacent to a node in state B adopt state B at rate $1 - \sigma_A^q$. Summing these rates over all nodes gives $([AB] + [BA])(1 - \sigma_A^q)$. Similarly, the rate at which nodes in state B adopt state A is $([AB] + [BA])(1 - \sigma_B^q)$.

Equation (3.7) indicates that the local surveys, which we capture in our mean-field approximation by σ_A and σ_B, have a global effect on the drift of opinion states, as they control

the sign of $\frac{d[A]}{dt}$. (By contrast, $\frac{d[A]}{dt} = 0$ in the linear RTR-CVM of Ref. [DGL12].) This suggests that network structure plays a more prominent role in how $[A]$ and $[B]$ evolve in our nonlinear RTR-CVM than in the linear RTR-CVM. For example, consider a network with two communities, where one community is densely connected and consists of C_A nodes in state A and the other community is sparsely connected and consists of C_B nodes in state B. We suppose that the second community is larger than the first (i.e., $C_B > C_A$). We also suppose that the two communities are linked to each other (in a way that we will make more precise in Section 3.3.4). When we select a discordant edge (i, j) with node i in state A and node j in state B, the local surveys satisfy $\sigma_i > \sigma_j$. On average, at least initially, we expect that more nodes in state B convert to state A than the reverse. However, the values of σ_A and σ_B can change rapidly in non-obvious ways as the system evolves, potentially reversing the sign of $\frac{d[A]}{dt}$. Therefore, it is not guaranteed that such a two-community network will terminate in a state with a large fraction of nodes in state A. In fact, as we will see in Section 3.3.4, whether this occurs depends on the nonlinearity parameter q.

3.3.4 Simulations on Stochastic Block Models

To explore how mesoscale network structures impact the dynamics of our nonlinear RTR-CVM, we simulate it on networks that we seed with such structures using a stochastic block model (SBM) [FH16, New18]. We assign each of the N nodes to one of two blocks, which we call "community" a and "community" b. Community a consists of cN nodes and community b consists of $(1 - c)N$ nodes, with $c < 1/2$. That is, community b has more nodes than community a. In our discussion, we seed all nodes in community a with opinion state A and all nodes in community b with opinion state B. We examine two types of mesoscale structures: "two-community structure" and "core–periphery structure".

In Section 3.3.3, we explored a mean-field approximation of our nonlinear CVM with an RTR scheme. This necessitated finding a suitable mean-field analog of the local surveys σ_i. We considered a state-heterogeneous approximation, in which we separately average over nodes in state A and state B. Denoting these approximations as σ_A and σ_B, we saw in

Equation (3.4) that

$$\sigma_A = \frac{1}{2N_{AB}} \left(N_{AA} - \sum_{i:S[i]=A,\, k_i \neq 0} \frac{s_i^2}{k_i} \right) \tag{3.8}$$

and

$$\sigma_B = \frac{1}{2N_{AB}} \left(N_{BB} - \sum_{i:S[i]=B,\, k_i \neq 0} \frac{s_i^2}{k_i} \right) . \tag{3.9}$$

Equivalently, we can write

$$\sigma_A = \sum_{i:S[i]=A,\, \bar{s}_i \neq 0} \frac{s_i \bar{s}_i}{(s_i + \bar{s}_i) \sum_{j:S[j]=A} \bar{s}_j} \tag{3.10}$$

and

$$\sigma_B = \sum_{i:S[i]=B,\, \bar{s}_i \neq 0} \frac{s_i \bar{s}_i}{(s_i + \bar{s}_i) \sum_{j:S[j]=B} \bar{s}_j} . \tag{3.11}$$

We seek to compute $\mathbb{E}[\sigma_A(0)]$ for systems that we seed with two-community SBM networks. (If we take all edge probabilities in the SBM to be equal, we obtain ER networks.) We calculate

$$\begin{aligned}
\mathbb{E}[\sigma_A(0)] &= \mathbb{E}\left[\sum_{i:S[i]=A,\, \bar{s}_i \neq 0} \frac{s_i \bar{s}_i}{(s_i + \bar{s}_i) \sum_{j:S[j]=A} \bar{s}_j} \right] \\
&= \sum_{i:S[i]=A,\, \bar{s}_i \neq 0} \mathbb{E}\left[\frac{s_i \bar{s}_i}{(s_i + \bar{s}_i) \sum_{j:S[j]=A} \bar{s}_j} \right] \\
&\approx N_A(0)[1 - \exp(P_{ab} N_B(0))] \times \mathbb{E}\left[\frac{s_i \bar{s}_i}{(s_i + \bar{s}_i) \sum_{j:S[j]=A} \bar{s}_j} \middle| \bar{s}_i > 0 \right] \\
&= N_A(0)[1 - \exp(P_{ab} N_B(0))] \times \mathbb{E}\left[\frac{s_i \bar{s}_i}{(s_i + \bar{s}_i) \left(\bar{s}_i + \sum_{j:S[j]=A,\, j \neq i} \bar{s}_j \right)} \middle| \bar{s}_i > 0 \right] .
\end{aligned} \tag{3.12}$$

The quantity s_i is the number of neighbors of node i that have the same opinion as i; it is distributed binomially with parameters $n = N_A(0) - 1$ and $p = P_{aa}$. The quantity $\bar{s}_i$ is the number of neighbors of node i that have a different opinion from i; it is distributed binomially with parameters $n = N_B(0)$ and $p = P_{ab}$. There are $N_A(0)$ nodes in state A; with probability

$$1 - (1 - P_{ab})^{N_B(0)},$$

each such node has at least one discordant edge. In the $N \to \infty$ limit, there are

$$N_A(0)[1 - \exp(P_{ab}N_B(0)]$$

nodes in state A with at least one discordant neighbor. We use this expression as a large-N approximation in Eq. 3.12. In this approximation, we replace summing over nodes that are in state A and are incident to at least one discordant edge by multiplying by the expected number of such nodes.

Edges in our SBM networks are independent of each other, so s_i, $\bar{s}_i$, and $\bar{s}_j$ are all independent random variables. For convenience, we define the notation

$$z := \sum_{j:S[j]=A,\, j\neq i} \bar{s}_j.$$

The sum of independent and identically distributed binomial random variables is another binomial random variable, so z is distributed binomially with parameter values

$$n = (N_A(0) - 1)N_B(0)$$

and $p = P_{ab}$. Using the law of the unconscious statistician, we obtain

$$\begin{aligned} &\mathbb{E}\left[\frac{s_i\bar{s}_i}{(s_i+\bar{s}_i)\,(\bar{s}_i+z)}\,\middle|\,\bar{s}_i > 0\right] \\ &= \frac{\sum_{t_i\geq 0,\,\bar{t}_i\geq 1,\,t\geq 0}\left(\frac{t_i\bar{t}_i}{(t_i+\bar{t}_i)(\bar{t}_i+z)}P(s_i=t_i)P(\bar{s}_i=\bar{t}_i)P(z=t)\right)}{1-P(\bar{s}_i=0)}. \end{aligned} \tag{3.13}$$

With Equation (3.13), we can numerically approximate $\mathbb{E}[\sigma_A(0)]$ for both ER and two-community SBM networks.

3.3.4.1 Two-Community SBM

To create a two-community network in which the smaller community (i.e., community a) is denser than the larger community (i.e., community b), the edge-probability matrix

$$P = \begin{pmatrix} P_{aa} & P_{ab} \\ P_{ba} & P_{bb} \end{pmatrix} \tag{3.14}$$

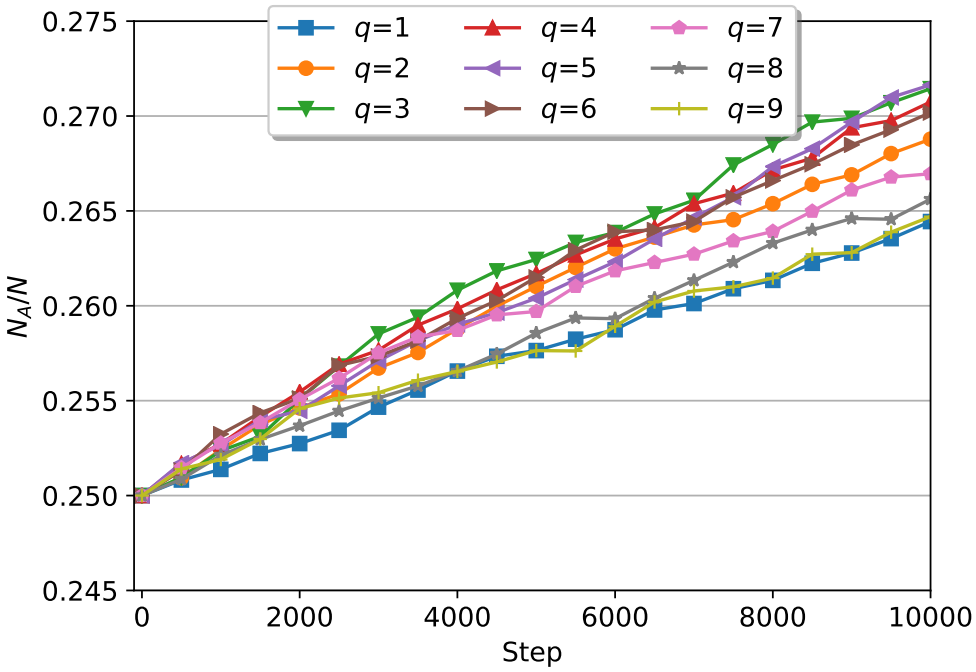


Figure 3.3: Density $N_A(t)/N$ of state A for the first $10,000$ elementary time steps for nine values of q in RTR simulations (see Algorithm 1) of our nonlinear CVM. We seed each of the nine realizations with two-community SBM networks, as described in Section 3.3.4.1. We show only initial and transient dynamics. For the terminal behavior of the simulations, see Figure 3.4a.

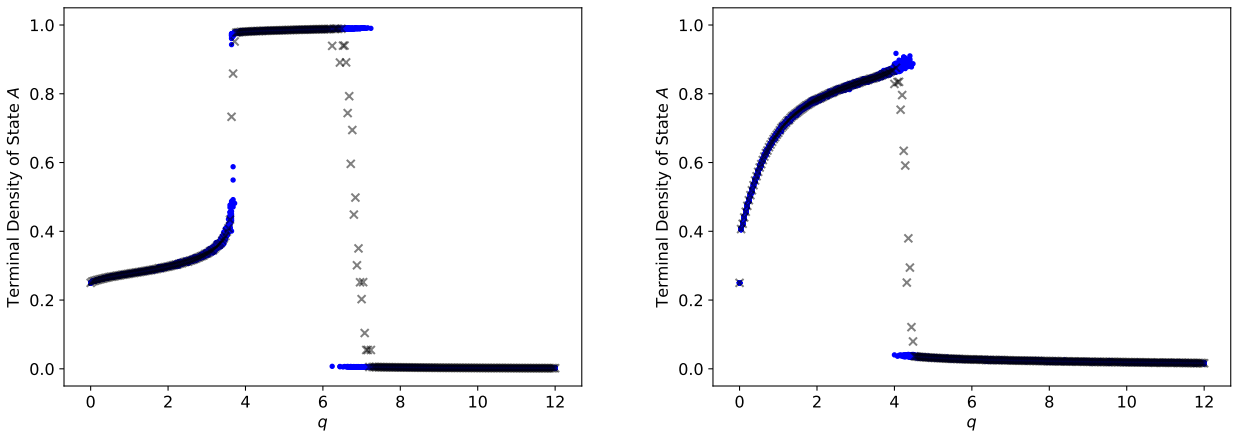
has probabilities that satisfy $P_{aa} > P_{bb} > P_{ab} = P_{ba}$. We initialize our simulations with networks with $c = 1/4$, and we set the SBM parameters to be $P_{aa} = \frac{12}{cN-1}$, $P_{bb} = \frac{4}{(1-c)N-1}$ (so that $N_{AA}(0) \approx N_{BB}(0)$), and $P_{ab} = 1/N$. In our simulations, we check that the expectations of σ_A and σ_B satisfy the inequality $\mathbb{E}[\sigma_A(0)] > \mathbb{E}[\sigma_B(0)]$. See Equation (3.13) for details. Accordingly, we expect that, at least initially, the density $N_A(t)/N$ of state A increases as the system evolves. In Figure 3.3, we plot $N_A(t)/N$ for the first $10,000$ elementary time steps of simulations for nine different values of q. In this plot, we show the initial (and transient) dynamics, rather than the full temporal evolution of our simulations to termination. The plot confirms the initial increase of $N_A(t)/N$. However, in Figure 3.4a, we observe that — despite this initial increase — the terminal value of $N_A(t)/N$ depends on the nonlinearity parameter q.

In Figure 3.4a, we plot the terminal state-A density from 20 simulations for each q. For $q \in [0, 3.8)$, between approximately 25 % and 60 % of the nodes terminate in state A, implying that the network fragments into multiple components. We observe what may be a hybrid phase transition [VES08, LCS16] at $q_{\mathrm{cr}} \approx 3.8$, where there appears to be a higher-order transition (i.e., at least second order) as $q \to (q_{\mathrm{cr}})^-$ and a first-order transition as $q \to (q_{\mathrm{cr}})^+$. For $q \in (q_{\mathrm{cr}}, 6.2)$, almost every node terminates in state A in all of our realizations, suggesting that rewiring dynamics are not fragmenting the system into multiple components.

For $q \in (6.2, 7)$, either almost every node terminates in state A or almost every node terminates in state B, and the latter occurs in progressively more realizations as we increase q. Although we do not show this in Figure 3.4a, the system has an additional regime for sufficiently large q. In this regime, the adoption action dominates and the system behaves like a voter model that does not coevolve with network structure (see Section 3.3.4.4). In this situation, state A becomes the consensus opinion with probability $N_A(0)/N = c = 1/4$.

In Figure 3.14a, we conduct simulations using the linear RTR-CVM seeded with a two-community SBM network with the same parameter values. We observe that Figure 3.14a resembles the outcome of initializing the linear RTR-CVM with an ER network (see Figure 2.3b). This suggests that, with respect to terminal state densities, the linear RTR-CVM may be less sensitive than our nonlinear RTR-CVM to initial community structure in a network.

As we show in Equation (3.13), we are able to numerically approximate the quantity $\mathbb{E}[\sigma_A(0)^q - \sigma_B(0)^q]$ for two-community SBM networks. We find that it depends on the parameters q, c, P_{aa}, P_{ab}, and P_{ba}. As the system evolves, however, it becomes challenging to track $\mathbb{E}[\sigma_A(t)^q - \sigma_B(t)^q]$ over time t. Nevertheless, from Figure 3.4a, we know that the temporal evolution is affected by the value of the nonlinearity parameter q.



(a) Two-Community Structure

(b) Core–Periphery Structure

Figure 3.4: Terminal density of state A from RTR simulations (see Algorithm 1) for our nonlinear CVM with $q \in [0, 12]$ and discretization $\Delta q = 0.04$. For each value of q, we simulate 20 realizations. In (a), we seed each realization with two-community SBM networks, as described in Section 3.3.4.1. In (b), we seed each realization with an SBM network with core–periphery structure, as described in Section 3.3.4.2. We plot individual realizations with blue dots and means with $\times$ symbols.

3.3.4.2 Core–Periphery Structure

We now investigate the dynamics of our nonlinear RTR-CVM with $\sigma_A(0) \gg \sigma_B(0)$ and $N_A(0) \ll N_B(0)$ using a core–periphery configuration of the SBM [CLW13, RPF17]. Our initial networks now have a small, densely connected core of nodes in state A and a large periphery of sparsely-connected nodes in state B. We label the core as block a and the periphery as block b. In this core–periphery network, the SBM probabilities in Equation (3.14) satisfy $P_{aa} > P_{ab} > P_{bb}$. In this scenario, a node i's local survey σ_i differs from the true global densities. For core nodes, $\sigma_i > 1/2$, so such nodes believe that their state (namely, state A) is the majority state, even though it is not. Conversely, for the peripheral nodes, $\sigma_j < 1/2$, so such nodes believe that their state (i.e., state B) is the minority state, even though it is not. When $q = 1$, if we select a core node as the primary node in an update, it is more likely to rewire than to adopt, at least initially. However, as most nodes are in

state B initially, the core node is likely to rewire to another node in state B. If we select a peripheral node as the primary node for an update, it is more likely to adopt than to rewire. At least initially, such adoptions convert peripheral nodes from the majority state B to the minority state A.

In our simulations of our nonlinear RTR-CVM on SBM networks with core–periphery structure, we take $3N_A(0) = N_B(0)$, and we set the probabilities to be $P_{aa} = \frac{20}{N_A(0)-1}$, $P_{bb} = \frac{1}{N_B(0)-1}$, and $P_{ab} = \frac{5}{N}$. In Figure 3.4b, we plot the terminal density of state A for various values of q from simulations on networks with $N = 50,000$ nodes. We observe a transition in the qualitative dynamics when q is in $(4, 4.5)$. As we increase q from 0 to 4, there are progressively more nodes that terminate in state A before the network fragments, with approximately 90 % of the nodes terminating in state A when $q = 4$. When $q \in (4, 4.5)$, the system appears to exhibit a phase transition that is similar to that of the ostensible hybrid phase transition of Figure 3.4a. In this case, however, the hybrid transition is discontinuous. In Section 3.3.4.4, we explore $q \in [0, 100]$. For $q \in (4.5, 39)$, we find that almost every node terminates in state B (see Figure 3.7).

3.3.4.3 Majority and Minority Illusions

Recent work by Lerman et al. [LYW16] on the majority illusion in social networks examined the phenomenon of distorted local observations when an opinion state that is globally rare in a network may be dramatically overrepresented in the local neighborhoods of many individuals. Using a model of threshold opinion dynamics, Lerman et al. illustrated that majority illusions can accelerate the spread of states that are initially rare. For our work with binary opinion states, we find it useful to distinguish between two different types of illusions. By a *majority illusion*, we mean the phenomenon of nodes in a minority state perceiving their state to be in the majority. Analogously, by a *minority illusion*, we mean the phenomenon of nodes in the majority state perceiving their state to be in a minority. In a model with binary opinion states, the minority illusion implies that nodes in the majority state incorrectly perceive that the minority state is held by the majority of nodes.

In our nonlinear CVM, a node i's local survey σ_i is based on a sample of the global population. In a locally well-mixed system (see Equation (3.1)), the sample leads to good estimates of the global densities $N_A(t)/N$ and $N_B(t)/N$. However, when seeding the system as in Section 3.3.4.1 and Section 3.3.4.2, the samples are biased initially. In Figure 3.5 and Figure 3.6, we plot the means of the local surveys $\overline{\sigma_A}$ (solid blue curves) and $\overline{\sigma_B}$ (solid red curves) versus elementary time steps and compare them to the true global densities, $N_A(t)/N$ (dashed blue curves) and $N_B(t)/N$ (dashed red curves), for simulations on systems that we seed with two-community structure and core–periphery structure, respectively. We calculate the unweighted means $\overline{\sigma_A}$ and $\overline{\sigma_B}$ from Equation (3.2), so we are treating all local surveys equally. In mathematical language, assuming that state A is in the minority (i.e., $N_A(t)/N < 1/2$), the majority illusion occurs when $\overline{\sigma_A} > 1/2$. Analogously, assuming that state B is in the majority (i.e., $N_B(t)/N > 1/2$), the minority illusion occurs when $\overline{\sigma_B} < 1/2$. In Table 3.1, we summarize how we seed networks with different types of illusions using an SBM network with $N_A(0) = cN$ and $c < 1/2$.

Illusion	Edge Probabilities
No Illusion	$P_{aa} = P_{ab} = P_{bb}$
Majority Illusion for A	$\frac{c}{1-c} P_{aa} \gg P_{ab}$
Minority Illusion for B	$\frac{1-c}{c} P_{bb} \ll P_{ab}$
Both Illusions	$\frac{1-c}{c} P_{bb} \ll P_{ab} \ll \frac{c}{1-c} P_{aa}$

Table 3.1: Summary of SBM parameters that we use to seed a network with a majority illusion, a minority illusion, both types of illusions, or neither illusion using an SBM network with $N_A(0) = cN$ nodes in state A and $c < 1/2$.

In Figure 3.5, we seed a network using the SBM two-community structure that we described in Section 3.3.4.1. Initially, the larger community (which has 3/4 of the nodes) consists of nodes in state B, and the smaller community (which has the remaining 1/4 of the nodes) consists of nodes in state A. However, the local surveys of the nodes in the smaller community lead them to perceive state A as the majority state and thus state B as the minority. Similarly, the local surveys of the nodes in the larger community lead them to

perceive state B as the majority state and thus state A as the minority. In other words, the larger community of nodes (which are in the majority state) correctly believe that their state is in the majority. However, the smaller community of nodes (which are in the minority state) experience a majority illusion, as they incorrectly believe that their state is in the majority.

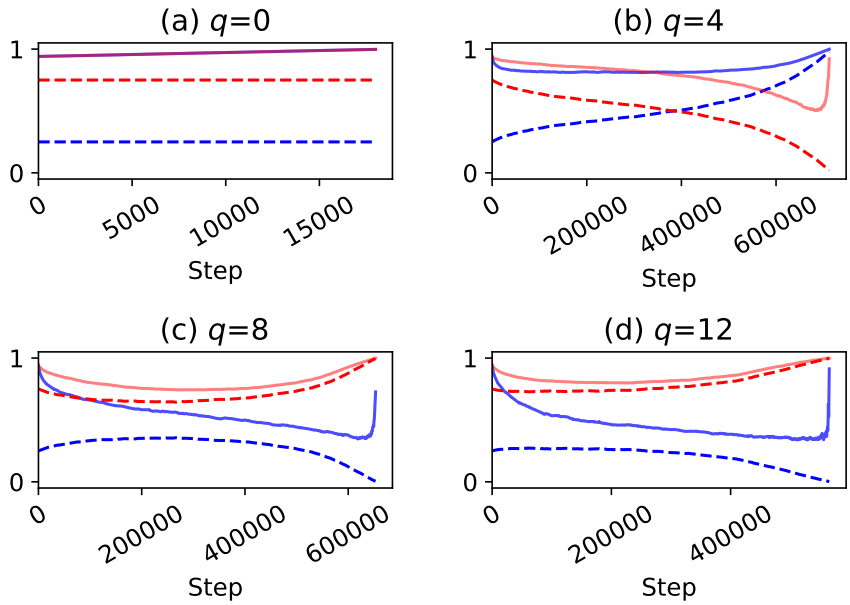


Figure 3.5: Unweighted means [$\overline{\sigma_A}$ (solid blue curve) and $\overline{\sigma_B}$ (solid red curve)] of local surveys for nodes in states A and B (see Equation (3.2) for details) and global state densities [N_A/N (dashed blue curve) and N_B/N (dashed red curve)] of states A and B versus elementary time steps in four RTR simulations (see Algorithm 1) of our nonlinear CVM. We seed each realization using the SBM two-community structure that we described in Section 3.3.4.1.

As we observed in Figure 3.4a, the effect on the terminal densities of initializing the system with a majority illusion depends on the value of the nonlinearity parameter q. For $q = 0$, in which only rewiring occurs, the state densities do not change, but the system fragments, such that each node only has neighbors that share its opinion state. Therefore, $\overline{\sigma_A}$ and $\overline{\sigma_B}$ increase to 1, and the majority illusion of state A increases in severity. For $q = 4$, the illusion becomes a reality, in the sense that N_A/N increases to match $\overline{\sigma_A}$. The system ultimately reaches an absorbing state with most nodes in state A, but there are still small

isolated clusters (i.e., separate components) of nodes that remain in state B, so $\overline{\sigma_B}$ increases to 1 near the end of a simulation (the mean that we take is over only these nodes). For $q = 8$ and $q = 12$, the nodes "wise up", in the sense that $\overline{\sigma_A}$ decreases to match N_A/N. The system ultimately reaches an absorbing state with most nodes in state B, but there are still small clusters of nodes in state A that cause $\overline{\sigma_A}$ to increase to 1 near the end of our simulations.

In Figure 3.6, we seed the system using the SBM core–periphery structure that we described in Section 3.3.4.2. Initially, all peripheral nodes (which constitute 3/4 of the nodes) are in state B, and the core nodes (which constitute the remaining 1/4 of the nodes) are in state A. The local surveys of the peripheral nodes lead them to perceive state B as the minority state and thus state A as the majority. Similarly, the local surveys of the core nodes lead them to perceive state A as the majority state and thus state B as the minority. The core nodes (which are in the minority state) experience a majority illusion, incorrectly believing that their state is in the majority. Conversely, the peripheral nodes (which are in the majority state) experience a minority illusion, incorrectly believing that their state is in the minority.

As we saw for two-community structure, the effect of the majority and minority illusions depends on the value of the nonlinearity parameter q in our networks with core–periphery structure. For $q = 0$, in which only rewiring occurs, state densities do not change, but the network fragments, such that each node only has neighbors that share its opinion state. Therefore, $\overline{\sigma_A}$ and $\overline{\sigma_B}$ increase to 1, and the majority illusion for state A increases in severity, but the minority illusion for state B dissipates. For $q = 4$, the majority illusion for state A becomes a reality, in the sense that N_A/N increases to match $\overline{\sigma_A}$. The minority illusion for state B also becomes a reality, in the sense that N_B/N decreases to match $\overline{\sigma_B}$ initially. However, near the end of a simulation, $\overline{\sigma_B}$ increases to 1 near the end, because there are still small clusters of nodes (i.e., separate components) in state B and $\overline{\sigma_B}$ is a mean over only these nodes. For $q = 8$ and $q = 12$, the nodes wise up to both illusions, as $\overline{\sigma_A}$ decreases to match N_A/N and $\overline{\sigma_B}$ increases to match N_B/N. The system ultimately reaches an absorbing state with most nodes in state B, but there are still small clusters of nodes in state A that cause $\overline{\sigma_A}$ to increase to 1 near the end of our simulations.

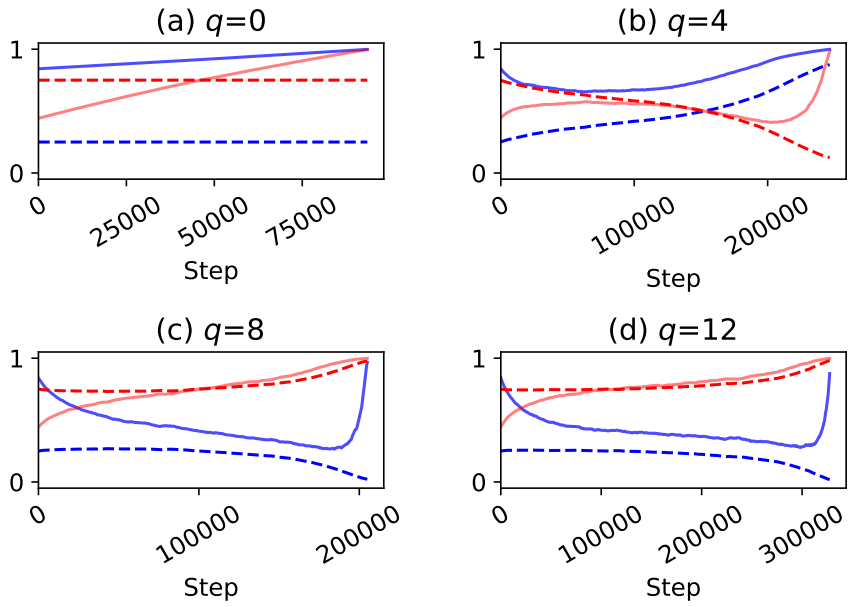


Figure 3.6: Unweighted means [$\overline{\sigma_A}$ (solid blue curve) and $\overline{\sigma_B}$ (solid red curve)] of local surveys for nodes in states A and B (see Equation (3.2) for details) and global state densities [N_A/N (dashed blue curve) and N_B/N (dashed red curve)] for states A and B versus elementary time steps in four RTR simulations (see Algorithm 1) of our nonlinear CVM. We seed each realization using the SBM core–periphery structure that we described in Section 3.3.4.2.

The examples in this subsection demonstrate that, under certain conditions, seeding our nonlinear RTR-CVM with illusions can lead to the spreading of initially rare opinion states. For instance, when we seed the system with an SBM two-community network such that there is a majority illusion but not a minority illusion and take the value of the nonlinearity parameter to be $q = 4$, almost every node adopts the initially rare opinion state in all realizations of our simulations. However, under other conditions, seeding the system with illusions can also stifle the spread of initially rare opinion states. For example, when we seed the system with core–periphery structure with both types of illusions and take the value of the nonlinearity parameter to be $q = 8$, the initially rare opinion state vanishes almost entirely in all realizations of our simulations. This behavior contrasts sharply with what we observe in the linear RTR-CVM, in which the probability that an initially rare opinion state spreads, conditioned on the event that an opinion state does indeed spread, is equal

to the initial fraction of nodes in the rare state. In other words, the rewiring rate α in the linear RTR-CVM affects whether some opinion state spreads to almost every node, but the initial state densities determine the probability of which state it will be. By contrast, the nonlinearity parameter q in our nonlinear RTR-CVM affects not only whether some opinion state spreads to almost every node but also the probabilities of which state it will be. See Section 3.6 for details.

3.3.4.4 Simulations of our Nonlinear CVM for Large Values of q

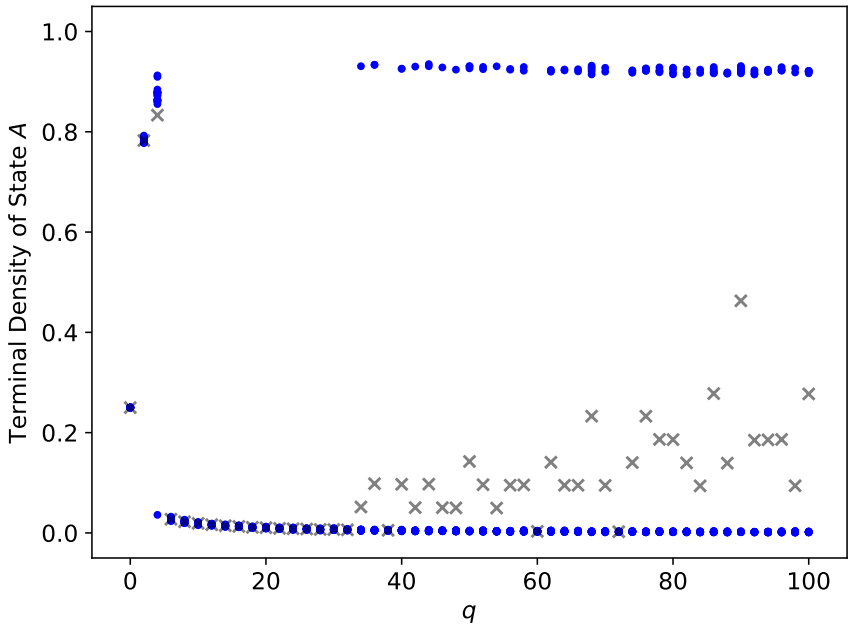


Figure 3.7: Terminal density of state A in simulations of our nonlinear RTR-CVM (see Algorithm 1) for $q \in [0, 100]$ with a step size of $\Delta q = 2$. For each value of q, we simulate 20 realizations. We seed each realization with a different SBM network with core–periphery structure (see Section 3.3.4.2), but now there are $N = 20,000$ nodes. We plot individual realizations with blue dots and means with $\times$ symbols.

In Figure 3.4, we observed regimes of q values in which almost every node terminates in state B, which initially has density 3/4, for all 20 realizations. This regime extends past $q = 12$, but we know that as $q \to \infty$, we must recover a voter model that does not coevolve

with network structure. We repeat one of our case studies for large values of q to improve our understanding of this limiting behavior.

We extend our case study from Section 3.3.4.2, in which we examined our nonlinear RTR-CVM seeded with core–periphery structure, by now considering $q \in [0, 100]$ for networks with $N = 20,000$ nodes. In Figure 3.7, we plot the terminal density of state A. In this case, we see that when $q \gtrapprox 38$, the adoption mechanism begins to dominate and state A becomes competitive with state B. In some trials, state A spreads to almost every node. For sufficiently large values of q, adoption actions should completely dominate and the system should behave like a voter model that does not coevolve with network structure; in such a scenario, almost every node terminates in the same state. Specifically, we expect that almost every node terminates in state A in $N_A(0)/N = 1/4$ of the realizations. In our computations, we observe this scenario for $q \gtrapprox 80$.

As we noted for Figure 3.14b, with our initial networks, approximately 10 % of nodes start in state B and are isolated. Therefore, when $q \gtrapprox 80$, the system terminates with these nodes still in state B (see Figure 3.7), even when state A has spread to every node in the largest connected component.

3.4 Rewire-to-Same

3.4.1 Model

In this section, we explore our nonlinear CVM with an RTS scheme. The key difference from the RTR scheme is that when rewiring occurs in the RTS scheme, the primary node deletes its discordant edge to the secondary node and then forms an edge with a node that we choose uniformly at random from the set of nodes in the same opinion state as the primary node. In Section 2.6.2, we compare our nonlinear RTS-CVM of Algorithm 2 to the linear RTS-CVM of Ref. [DGL12]. Based on previous work [BS17, SDH14], it seems that RTS schemes have been more difficult to analyze quantitatively than RTR schemes in linear CVMs.

To fully specify the RTS scheme, we start by examining a peculiarity of the RTS scheme.

What happens when a rewiring action cannot take place, because the primary node is already adjacent to all nodes in its opinion state (including the trivial case in which there are no other nodes in its state)? This situation is likely to arise if a system approaches consensus or if a network is densely connected (specifically, if the mean degree satisfies $\langle k \rangle \geq N/2$) [BS17, BDZ15]. There are several possible rules to employ, and the choice of rule may affect both the outcome and the analysis. Possible specifications include the following:

(i) stipulate that there is no replacement edge;
(ii) stipulate that we instead perform an RTR action;
(iii) stipulate that the recently deleted discordant edge reforms;
(iv) stipulate that the recently deleted discordant edge reforms and that the primary node instead performs an adoption action; and
(v) stipulate that we allow multi-edges, self-edges, or both.

Each of these choices either introduces a new mechanism, such as edge deletion, or changes the class of allowed networks. (Previously, we demanded that networks have neither self-edges nor multi-edges.) We choose to use option (i) of letting no replacement edge form, such that $|E(t)|$ is no longer a conserved quantity. By contrast, recent work on CVMs on dense random graphs allowed the formation of multi-edges [BS17]. We give a formal description of our nonlinear RTS-CVM in Algorithm 2.

Algorithm 2 Nonlinear Rewire-to-Same (RTS) Coevolving Voter Model

procedure FITTINGINVM(V, E, S, q) ▷ Input: Initial network and opinion states
  $E_D \leftarrow Discordant(V, E, S)$; $t \leftarrow 0$; $Record(V, E, S, t)$
  $V_A \leftarrow GetNodesByState(V, S, A)$; $V_B \leftarrow GetNodesByState(V, S, B)$
  while $E_D \neq \emptyset$ **do** ▷ While there are discordant neighbors
    $(i, j) \leftarrow RandomChoice(E_D)$
    $PrimaryNode, SecondaryNode \leftarrow RandomPermutation(i, j)$
    $\sigma \leftarrow LocalVote(PrimaryNode, V, E, S)$
    $u \leftarrow \mathcal{U}(0, 1)$
    $PotentialNewNeighbors \leftarrow V_{S[PrimaryNode]} \setminus \Gamma(PrimaryNode, E)$
    if $u \leq \sigma^q$ **then** ▷ Rewire
      $E.remove(PrimaryNode, SecondaryNode)$
      if $PotentialNewNeighbors \neq \emptyset$ **then**
        $NewNeighbor \leftarrow RandomChoice(PotentialNewNeighbors)$
        $E.add(PrimaryNode, NewNeighbor)$
    else ▷ Adopt
      $S[PrimaryNode] \leftarrow S[SecondaryNode]$
      $V_A \leftarrow GetNodesByState(V, S, A)$; $V_B \leftarrow GetNodesByState(V, S, B)$
    $E_D \leftarrow Discordant(V, E, S)$; $t \leftarrow t + 1$; $Record(V, E, S, t)$

3.4.2 Simulations

We first simulate our nonlinear RTS-CVM on ER networks. In Figure 3.8, we observe similar qualitative behavior as we did in our nonlinear RTR-CVM (see Figure 3.2b). Specifically, there seems to be a continuous transition between fragmentation and consensus regimes. One difference between these models is that the transition occurs at about $q \approx 4.5$ in the RTS variant of the model. The similarity of these two models contrasts starkly with results for the linear CVM of Ref. [DGL12]; the behavior of that CVM differs significantly under the two rewiring schemes (see Figure 2.3a and Figure 2.4). Prior research on linear CVMs suggests

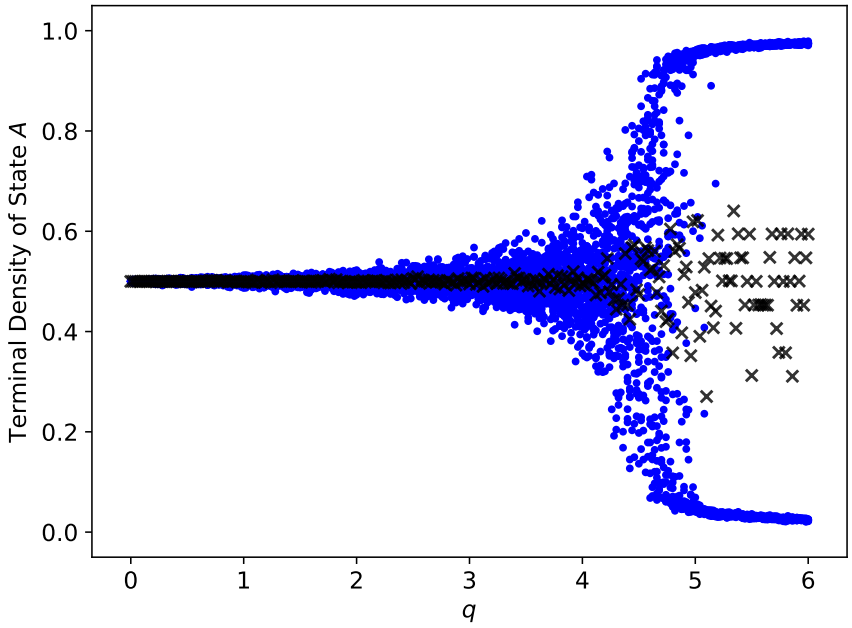


Figure 3.8: Terminal density of state A in RTS simulations (see Algorithm 2) for our non-linear CVM for $q \in [0, 6]$ with an increment of $\Delta q = 0.02$. For each value of q, we simulate 20 realizations. We seed each realization with a different ER network with $N = 50,000$ nodes and an edge probability of $p = \frac{4}{N-1}$, and we initialize half of the nodes in state A and the other half in state B. We plot individual realizations with blue dots and means with $\times$ symbols.

that the rewiring scheme affects linear CVMs significantly because of strong correlations between network structure and node states that arise from RTS actions, but not from RTR actions [DVB14]. By contrast, the similarity of results between the RTR and RTS variants for our nonlinear CVM suggests that such correlations play a less prominent role in our model than they do in the previously studied linear CVMs.

The sketch in Figure 3.9 illustrates what Demirel et al. [DVB14] reported as a "typical" configuration that is near fragmentation for a linear RTS-CVM. As the system evolves, nodes group into communities, which are connected to each other by only a few edges. Occasionally, there is an adoption that creates many discordant edges that are concentrated at one node. This leads to a disproportionately large number of ABA-triples that are concentrated on one node and constitutes a strong three-node correlation. However, in our nonlinear RTS-CVM, such situations occur much less frequently than they do in the linear RTS-CVM of Ref. [DGL12]. In fact, in our nonlinear CVM, the more ABA-triples that a node in state A would create by adopting state B, the less likely it is to adopt state B. To illustrate this observation, suppose that node i is in state A and has at least two neighbors in state A. If we select node i for an update, it adopts state B with probability $1-\sigma_i^q$, where σ_i is the fraction of nodes that are adjacent to node i and are also in state A. Consequently, the probability of node i creating ABA-triples by adopting state B decreases as the number of concordant edges that are incident to node i (i.e. the edges that would form the ABA-triples) increases.

In Figure 3.10, we plot the temporal evolution of the unweighted means, $\overline{\sigma_A}$ and $\overline{\sigma_B}$, of local surveys and compare them to global state densities (as in Section 3.3.4.3). We observe fragmentation in our simulations with $q = 0$, $q = 2$, and $q = 4$. Fragmentation causes the local surveys of nodes to become distorted, because nodes are in clusters and have no neighbors in a different opinion state. This leads to a weak form of a majority illusion for nodes in each opinion state. Although roughly half of the nodes are in state A and roughly half are in state B, almost every node perceives its own state to be in the majority. In our simulations with $q = 6$, we again initially observe a weak majority illusion for nodes in both states. As the system evolves, the density of state A decreases and $\overline{\sigma_A}$ decreases commensurately, but nodes in state A still experience a majority illusion. The

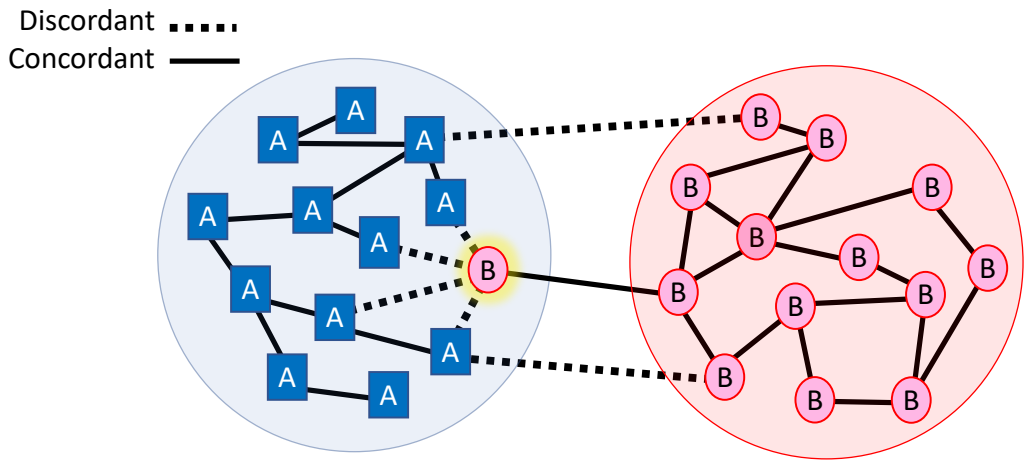


Figure 3.9: "Typical" configuration near fragmentation for a linear RTS-CVM. In this example, the highlighted node changes from state A to state B, and all of its previously concordant edges become discordant. Several discordant edges are now associated with a single node, thereby inducing ABA-triple correlations. [This illustration is our version of Figure 5 of Ref. [DVB14].]

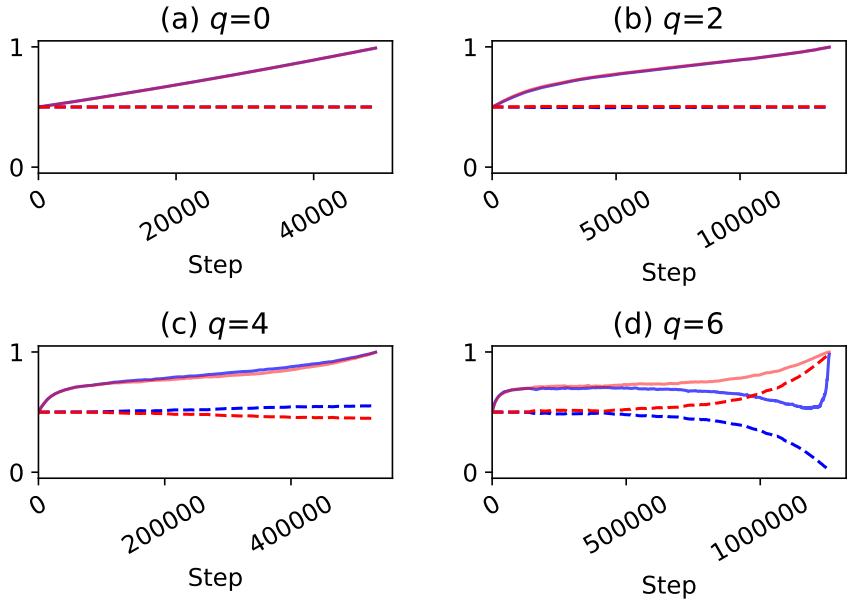
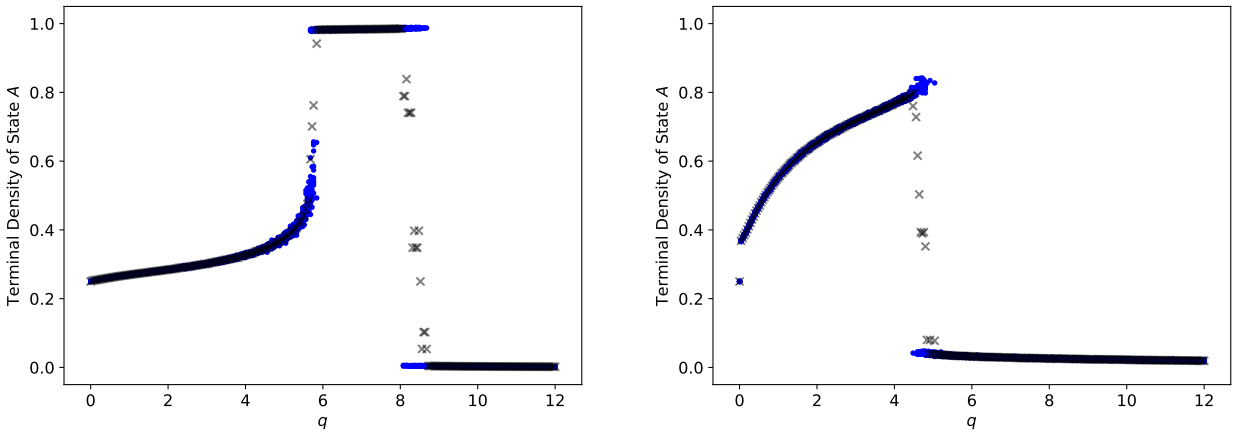


Figure 3.10: Unweighted means [$\overline{\sigma_A}$ (solid blue curve) and $\overline{\sigma_B}$ (solid red curve)] of local surveys for nodes in states A and B (see Equation (3.2) for details) and global state densities [N_A/N (dashed blue curve) and N_B/N (dashed red curve) for states A and B, respectively] versus elementary time steps in four RTS simulations (see Algorithm 2) in our nonlinear CVM. We seed each realization with an ER network with $N = 50,000$ nodes and edge probability $p = \frac{4}{N-1}$, and we initialize half of the nodes in state A and the other half in state B.

system ultimately reaches an absorbing state with most nodes in state B, but there are still small clusters of nodes in state A that cause $\overline{\sigma_A}$ to increase to 1 near the end of our simulations.



(a) Two-Community Structure

(b) Core–Periphery Structure

Figure 3.11: Terminal density of state A for RTS simulations (see Algorithm 2) of our nonlinear CVM for nonlinearity parameters $q \in [0, 12]$ with an increment of $\Delta q = 0.04$. For each value of q, we simulate 20 realizations. In (a), we seed our simulations with the two-community SBM networks that we described in Section 3.3.4.1. In (b), we seed our simulations with the SBM core–periphery structure that we described in Section 3.3.4.2. We plot individual realizations with blue dots and means with $\times$ symbols.

We conclude this section on our nonlinear RTS-CVM by conducting simulations that we seed with (1) two-community structure and (2) core–periphery structure using SBM networks (see Section 3.3.4). In Figure 3.11a, we observe that our nonlinear RTS-CVM seeded with two-community structure exhibits qualitatively similar long-time behavior as our nonlinear RTR-CVM seeded with the same two-community structure (see Figure 3.4a). Nevertheless, simulations using these two rewiring schemes do exhibit quantitative differences, such as in the locations of the transitions between regimes with qualitatively different terminal statistics. In Figure 3.11b, we see that the results of simulations of our nonlinear RTS-CVM that we seed with core–periphery structure are similar to those of our nonlinear RTR-CVM seeded with the same structure (see Figure 3.4b).

3.5 Rewire-to-None

3.5.1 Model

We now examine our nonlinear CVM with an RTN scheme. In this RTN-CVM, adoption occurs with probability $1 - \sigma_i^q$, and edge deletion occurs with probability σ_i^q; there are no replacement edges. We give a precise description of this model in Algorithm 3.

Algorithm 3 Nonlinear Rewire-to-None (RTN) Coevolving Voter Model

procedure FITTINGINVM(V, E, S, q) ▷ Input: Initial network and opinion states
  $E_D \leftarrow Discordant(V, E, S)$; $t \leftarrow 0$; $Record(V, E, S, t)$
  while $E_D \neq \emptyset$ **do** ▷ While there are discordant neighbors
    $(i, j) \leftarrow RandomChoice(E_D)$
    $PrimaryNode, SecondaryNode \leftarrow RandomPermutation(i, j)$
    $\sigma \leftarrow LocalVote(PrimaryNode, V, E, S)$
    $u \leftarrow \mathcal{U}(0, 1)$
    if $u \leq \sigma^q$ **then** ▷ Rewire
      $E.remove(PrimaryNode, SecondaryNode)$
    else ▷ Adopt
      $S[PrimaryNode] \leftarrow S[SecondaryNode]$
    $E_D \leftarrow Discordant(V, E, S)$; $t \leftarrow t + 1$; $Record(V, E, S, t)$

There have been some studies of RTN schemes in opinion models. For example, Refs. [GZ06, ZG06] examined an RTN scheme in a linear CVM. Additionally, a bounded-confidence opinion model with edge deletion (to model unfollowing on social media) was studied recently in Ref. [SCP19]. In Section 2.6.3, we investigated a linear CVM with our RTN scheme both analytically and computationally. There are many reasons to study an RTN mechanism in opinion models. In some sense, the edge-deletion mechanism is simpler than mechanisms that require additional parameters and specification of a rewiring rule [5]. Moreover, edge

[5] However, edge deletion can pose a mathematical challenge, because the number of edges and the mean degree are no longer constant [KMS17].

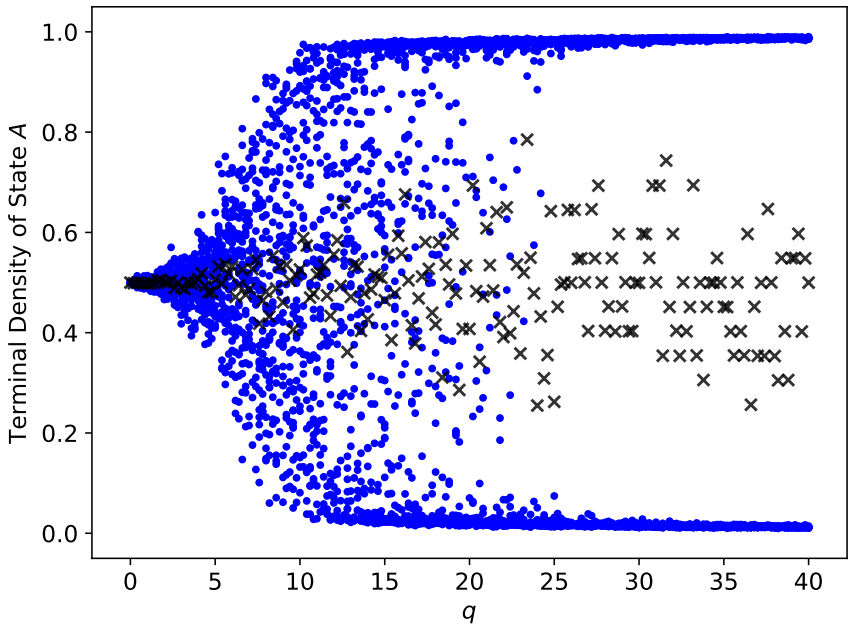
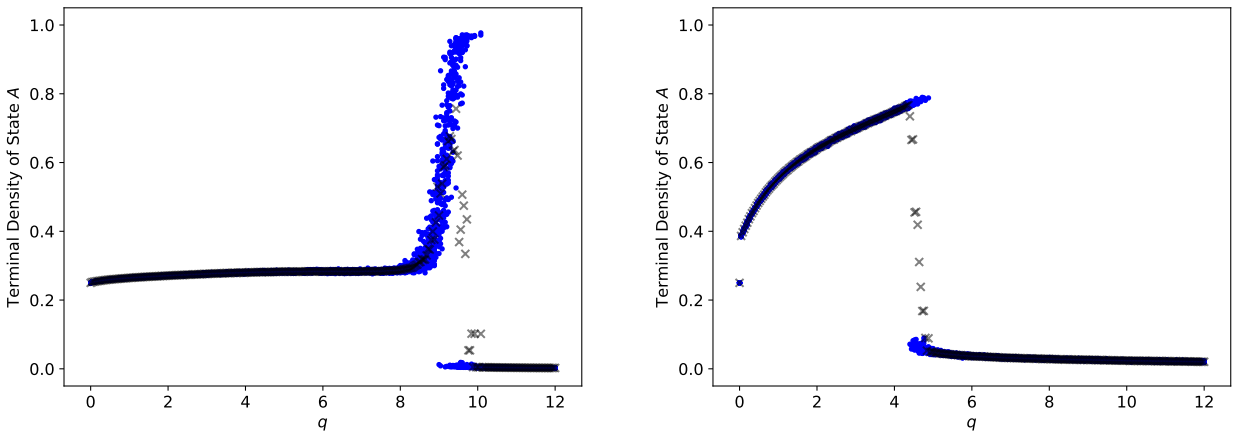


Figure 3.12: Terminal density of state A in RTN simulations (see Algorithm 3) of our nonlinear CVM for $q \in [0, 40]$ with an increment of $\Delta q = 0.2$. For each value of q, we simulate 20 realizations. We seed each realization with a different ER network with $N = 25,000$ nodes and an edge probability of $p = \frac{4}{N-1}$, and we initialize half of the nodes in state A and the other half in state B. We plot individual realizations with blue dots and means with $\times$ symbols.

deletion may also be more relevant than rewiring for internet social dynamics, because individuals perform actions such as unfriending or unfollowing without necessarily friending or following another account [JD15, GLH19, SZL18]. Edge deletion is also an important network mechanism in the structural evolution of social networks [FWG17, SRU08].

3.5.2 Simulations

We seed our nonlinear RTN-CVM with ER $G(N, p)$ networks with $N = 25,000$ nodes and an edge probability of $p = \frac{4}{N-1}$, and we initialize half of the nodes in state A and the other half of the nodes in state B. We plot the terminal state density of A in Figure 3.12. When $q = 0$, nodes do not adopt new opinions, so state densities do not change before the network



(a) Two-Community Structure

(b) Core–Periphery Structure

Figure 3.13: Terminal density of state A in simulations of our nonlinear RTN-CVM for $q \in [0, 12]$ with an increment of $\Delta q = 0.04$. For each value of q, we simulate 20 realizations. In (a), we seed the system with the two-community SBM networks that we described in Section 3.3.4.1. In (b), we seed the system with the SBM core–periphery networks that we described in Section 3.3.4.2. We plot individual realizations with blue dots and means with $\times$ symbols.

fragments. With edge deletion (i.e., the RTN mechanism), fragmentation of a network into disconnected components can occur for a wide range of q values (up to at least $q = 20$). This is a larger range than what we observed for the RTR and RTS schemes for our nonlinear CVM. For the RTR scheme, we did not observe fragmentation for $q \gtrapprox 3$; for the RTS scheme, we did not observe fragmentation for $q \gtrapprox 5.5$.

We conduct simulations using two-community SBM networks (see Section 3.3.4.1) to seed the system. We plot the terminal density of state A in Figure 3.13a. For $q \in (0, 8.5)$, we observe that the two communities separate from each other and that there are no signficant changes to the densities of opinion states. In contrast to the RTR scheme that we illustrated in Figure 3.4a and the RTS scheme in Figure 3.11a, for the nonlinear RTN-CVM, we do not observe any values of q in which almost every node terminates in state A for every realization of a simulation. Near $q \approx 9.52$, state A spreads to most nodes in most realizations, with a mean terminal density of approximately 0.75 over the 20 realizations. However, in some

realizations near $q \approx 9.52$, almost every node terminates in state B.

Finally, we simulate our nonlinear RTN-CVM model using SBM core–periphery networks (see Section 3.3.4.2) to seed the system. In Figure 3.13b, we observe that the RTN scheme produces terminal behavior that is qualitatively very similar to what we observed with the RTR (see Figure 3.4b) and RTS (see Figure 3.11b) schemes.

3.6 Linear Coevolving Voter Models on Stochastic Block Models

We compare the simulation results for our nonlinear CVM to the linear CVM that was studied in Ref. [DGL12]. In Section 2.6, we presented the linear model and studied it through simulations on ER networks. Now we study it on SBM networks.

3.6.1 Linear Rewire-to-Random CVM on SBMs

First, we compare the simulation results for our nonlinear CVM with an RTR scheme to the linear RTR-CVM that was studied in Ref. [DGL12]. We seed the system with two-community structure using SBMs. We use the same parameter values as in Section 3.3.4.1, so $N_A(0)/N = 1/4$ of the nodes are in state A. In Figure 3.14a, we plot the terminal density of state A from our simulations. The plot has roughly the same shape as when we seeded the linear CVM with an ER network (see Figure 2.3b). When $1 - \alpha \lessapprox 0.2$, rewiring actions dominate. The terminal minority-state densities are constant, with a value of 0.25. Adoption actions compete with rewiring actions when $1 - \alpha \gtrapprox 0.2$, and the plot appears to branch, with one branch decreasing to 0 and the other increasing to 1 as $1 - \alpha \to 1$. This illustrates that, by the time the system terminates, there are progressively larger changes to the state densities of the system for progressively larger values of $1 - \alpha$. Because the system begins with $N_A(0)/N = 1/4$, terminating along the upper branch (i.e., there is a positive change for $N_A(t)/N$) occurs in approximately $1/4$ of the realizations, as indicated by the values of the means of the terminal state-A densities.

In Figure 3.14b, we plot simulations of the linear RTR-CVM in which we seed the system

with core–periphery structure. We use the same parameter values as in Section 3.3.4.2. The plot's similarity to Figure 3.14a illustrates an insensitivity of this linear RTR-CVM to some types of initial network structure. With the parameter values of our seed core–periphery networks, approximately 10 % of the nodes begin in state B and are isolated. When $1-\alpha = 1$, there is no rewiring, so these nodes remain isolated and thus do not change their opinion states. Therefore, in the realizations with $1 - \alpha = 1$ in which opinion state A ultimately dominates the network, only approximately 90% of the nodes adopt state A.

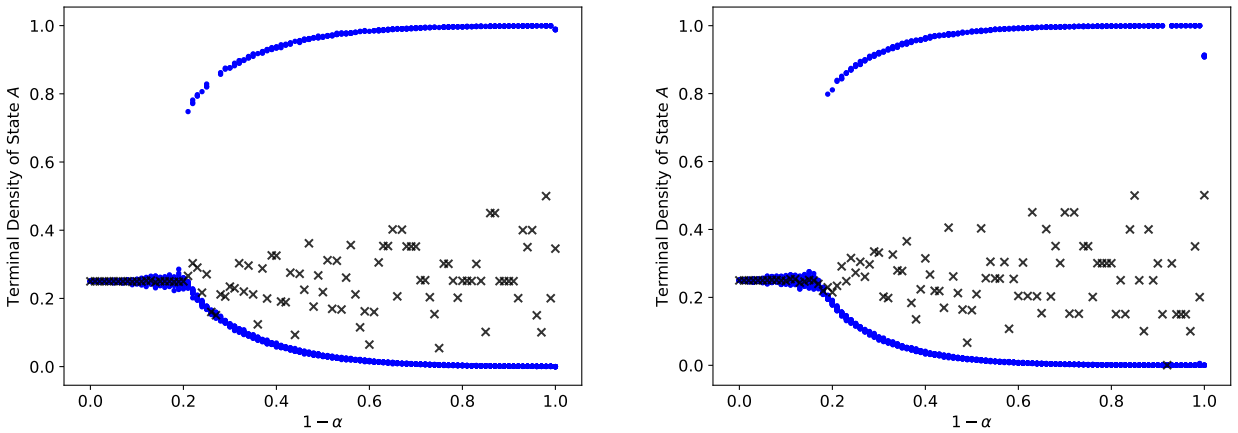
There are values of α in Figure 2.3b and Figure 3.14 for which none of our realizations terminate with almost all nodes in state A. The probability that a realization terminates along the top branch (signifying a net positive change in the density of state A) is $N_A(0)/N = 1/4$, so we expect on rare occasions (specifically, with probability $(3/4)^{20} \approx 0.0032$) that all 20 realizations for a particular value of α terminate along the bottom branch (i.e., with a net negative change in the density of state A).

3.6.2 Linear Rewire-to-Same CVM on SBMs

Next, we compare the results of simulations of our nonlinear RTS-CVM to simulations of the linear RTS-CVM from Ref. [DGL12]. In Figure 3.15, we plot the terminal state-A densities from simulations on networks that we seed with two-community structure (see Section 3.3.4.1) and on networks that we seed with core–periphery structure (see Section 3.3.4.1). In both cases, we observe qualitatively similar results as in Figure 2.4, which again suggests that this linear CVM is less sensitive than our nonlinear CVM to some initial network structures. It also suggests that this linear CVM is more sensitive than our nonlinear CVM to the choice of rewiring mechanism.

3.7 Conclusions and Discussion

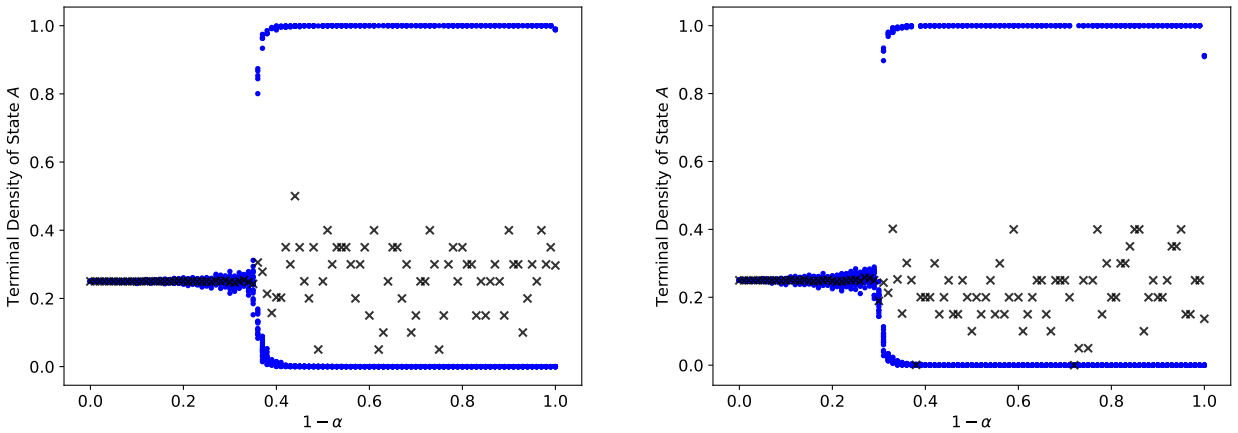
We explored a novel nonlinear coevolving voter model in which nodes take local information into consideration for their update actions, and we examined variants of our model with three different rewiring schemes: rewire-to-same, rewire-to-random, and rewire-to-none (i.e.,



(a) Two-Community Structure

(b) Core–Periphery Structure

Figure 3.14: Terminal density of state A in simulations of the linear RTR-CVM from Ref. [DGL12] for $\alpha \in [0, 1]$ with a step size of $\Delta\alpha = 0.01$. For each value of α, we simulate 20 realizations. In (a), we seed each realization with a different two-community SBM network (as described in Section 3.3.4.1). In (b), we seed each realization with a different core–periphery structure using SBM networks (as described in Section 3.3.4.1). We plot individual realizations with blue dots and means with $\times$ symbols.



(a) Two-Community Structure

(b) Core–Periphery Structure

Figure 3.15: Terminal density of state A in simulations of the linear RTS-CVM from Ref. [DGL12] for $\alpha \in [0, 1]$ with a step size of $\Delta\alpha = 0.01$. For each value of α, we simulate 20 realizations. In (a), we seed each realization with a different two-community SBM network (as described in Section 3.3.4.1). In (b), we seed each realization with a different core–periphery structure using SBM networks (as described in Section 3.3.4.1). We plot individual realizations with blue dots and means with $\times$ symbols.

unfriending). In our nonlinear CVM, each node can be in one of two opinion states. Additionally, updates are edge-based and occur asynchronously. An updating node i surveys its neighbor and records the fraction σ_i that share its state. With probability σ_i^q, for a nonlinearity parameter q, the node rewires a selected discordant connection; otherwise, with complementary probability $1 - \sigma_i^q$, it adopts a new opinion state.

By conducting extensive numerical simulations, we observed that our nonlinear CVM exhibits qualitatively similar characteristics as the linear CVM of Ref. [DGL12] with respect to terminal state densities when both models are initialized on ER networks with equal state densities $N_A(0)/N = N_B(0)/N = 1/2$. For example, both types of models possess a regime with rapid fragmentation into communities of different opinion states and a regime in which the system reaches a consensus. However, when we seed our nonlinear CVM with more complicated network structures, such as ones with community structure or core–periphery structure, we observed striking differences between our nonlinear CVM and the aforementioned linear CVM. In these scenarios, when the nodes have distorted views of local densities — such that they believe that they are in the majority or minority when the opposite is true — the value of the nonlinearity parameter q has a major effect on terminal state densities. For certain values of q and certain initial network topologies, the initially minority state consistently became the consensus in our simulations; for other values of q, the initially majority state consistently became the consensus. Although further analysis (especially of finite-size effects) is necessary, our work suggests that, on certain networks, our nonlinear CVM exhibits a rich assortment of phase transitions. The impact of initial network topology on terminal state densities distinguishes our nonlinear CVM from the linear CVM of Ref. [DGL12]. We also demonstrated that unlike the linear CVM of Ref. [DGL12], which is very sensitive to the choice of rewiring mechanism, our nonlinear CVM yields qualitatively similar behavior with both the RTR and RTS mechanisms.

Our nonlinear CVM also exhibits fascinating manifestations of both majority and minority illusions. For example, we observed that majority illusions can arise as a system evolves, and we also found that such illusions can resolve in different ways (e.g., by becoming true or by nodes wising up) for different values of the nonlinearity parameter q.

Our investigation of our nonlinear CVM raises several interesting questions. For example, we noted in Section 3.4 and Section 3.5 that the differences in model behavior from rewiring schemes in our nonlinear CVM are far less pronounced than they are in the linear CVM of Ref. [DGL12], and it is desirable to develop a mechanistic understanding of this qualitative difference between these families of models. It will also be interesting to develop precise conditions that determine when majority and minority illusions arise in our nonlinear CVM. Such illusions can either accelerate or stifle the spread of rare opinion states, so it is worthwhile to develop an understanding of the mechanisms that lead to these effects. To examine these ideas further, it will be interesting to explore the dynamics of our models on a larger variety of networks, such as those that were developed recently by Stewart et al. [SMD19]. It is also desirable to extend tools for approximating the dynamics of linear CVMs (such as approximate master equations [Gle13] and pair approximations [Jed17]) to nonlinear CVMs.

There are also many fascinating ways to extend our nonlinear CVM (see Section 2.4). We anticipate that it will be particularly interesting to incorporate ideas from recent efforts that have examined the effects of noise (e.g., random state mutations) [JXC13], hipsters (in the form of nodes that try to be in a minority) [JP19], and zealots (in the form of nodes that do not change states) [KWD17].

Opinions and social networks are coupled to each other intimately in a complex way. Developing and refining models for coevolving opinions and social networks can help improve understanding not only of their relationships with each other but also of their impact on political and social polarization, echo chambers, and other social phenomena.

CHAPTER 4

A Polyadic Extension of Bounded-Confidence Models

In this chapter, I present an ongoing project on opinions models that is a collaboration with Heather Z. Brooks (UCLA), Michelle Feng (UCLA), and Mason A. Porter. We are exploring extensions of bounded-confidence models to hypergraphs.

Thus far, we have focused on voter models, a single (but very popular) class of opinion models. We now consider bounded-confidence models (BCM), which differ from voter models in two key ways. First, the opinion space for BCMs typically is continuous. This is useful for issues that occur on a continuum, such as beliefs about minimum wages or taxes. Real-valued intervals are also suitable for representing a probability of choosing one of two choices. Second, in the BCMs that have traditionally been studied, individuals behave deterministically. (There is a layer of randomness if one uses asynchronous updates.) Using deterministic, instead of stochastic, dynamics is primarily a modeling decision, but it also affects what analytical techniques one uses.

4.1 Background

Bounded-confidence models (BCM) are continuous-opinion state models in which an individual only interacts with neighbors who hold opinions that are sufficiently close to their own [Lor07]. The two most common BCMs are the (asynchronous) Deffuant–Weisbuch model [DNA00, WDA02] and the (synchronous) Hegselmann–Krause model [HK02].[1] For a review of the two models, their similarities, and their differences, see Ref. [Lor07].

[1]See Section 2.2 for a discussion about asynchronous versus synchronouns updates in the context of voter models

For concreteness, let us consider the Deffuant–Weisbuch (DW) model. In the DW model on a complete (all-to-all) graph, at each time step t, one chooses a pair (i, j) of nodes uniformly at random. If the difference $|x_i(t) - x_j(t)|$ of opinions is below some threshold $\epsilon_{i,j}$, then they adjust their opinions by

$$x_i(t+1) = x_i(t) + \mu_{i,j}(x_j(t) - x_i(t))$$

and

$$x_j(t+1) = x_j(t) + \mu_{i,j}(x_i(t) - x_j(t))\,;$$

otherwise $x_i(t+1) = x_i(t)$ and $x_j(t+1) = x_j(t)$. The opinions of all other nodes remains fixed. With this type of averaging, the mean opinion of the population is a conserved quantity of the system. The parameters $\epsilon_{i,j} \in [0, \infty)$ are the confidence bounds and they model how open-minded or closed-minded individuals are to each other's opinions [PC86]. The parameters $\mu_{i,j} \in [0, 0.5]$ are called the "multipliers" and they control the rate at which individuals adjust their opinion. Using a single value ϵ and μ for all pairs leads to what is sometimes called the "homogeneous" DW model. For a DW model on a network, rather than select a pair of nodes uniformly at random, one typically selects an edge from the edge set uniformly at random.

4.2 A BCM on Hypergraphs

We now define our BCM for a *hypergraph* as an extension of the homogeneous DW model. Hypergraphs are a polyadic generalization of graphs [New18, SBH20]. In addition to dyadic (i.e., pairwise) edges, we allow hyperedges, which encode connections between an arbitrary number of nodes. The space $\mathcal{E}$ of possible hyperedges is the power set $\mathcal{P}(V)$ of the set V of nodes. Let $N := |V|$ denote the number of nodes. For the hypergraphs (V, E) that we consider in the context of our BCM, we restrict the hyperedge set $E \subset \{e \in \mathcal{E} |\ |e| \geq 2\}$ so that each hyperedge is incident to at least two nodes. In addition to a hypergraph, we have the time-dependent opinion configuration $\boldsymbol{x}(t) \in O^N$, where we take the opinion space O to be the the real line $\mathbb{R}$.

To generalize the notion of confidence to hyperedges, we define a *discordance* function $d : E \times O^N \to \mathbb{R}_{\geq 0}$ that maps a hyperedge and opinion configuration to a real number. This function measures the lack of agreement in a hyperedge; we use it to determine whether the members of the hyperedge update their opinions. We present a class of discordance functions

$$d_\alpha(e, \boldsymbol{x}) = \left(\frac{1}{|e| - 1}\right)^\alpha \sum_{i \in e} (x_i - \bar{x}_e)^2 , \tag{4.1}$$

parameterized by α, where $\bar{x}_e = 1/|e| \sum_{i \in e} x_i$. We focus on two specific cases, $\alpha = 0$ and $\alpha = 1$. In the former case, $d_0(e, \boldsymbol{x})$ penalizes large hyperedges, in the sense that

$$d(e, \boldsymbol{x}) \leq d(e', \boldsymbol{x}) \text{ if } e \subset e', \tag{4.2}$$

with equality holding if and only if $x_i = \bar{x}_e$ for all $i \in e' \setminus e$. We call a discordance function that satisfies Equation (4.2) *hyperedge monotonic*. This models a situation in which large groups tend to be less effective than small groups at changing opinions. In the latter case, $d_1(e, \boldsymbol{x})$ coincides with the unbiased sample variance of the opinions of the nodes in e. The scaling by $1/(|e|-1)$ prevents hyperedges with many nodes from being disadvantaged relative to hyperedges with few nodes. Specifically, if the opinions are independent and identically distributed, then the expected d_1-discordance for any subset of nodes is

$$\mathbb{E}[d_1(e, \boldsymbol{x})] = \mathbb{E}[d_1(e', \boldsymbol{x})] , \quad \text{for all } e, e' \in E . \tag{4.3}$$

Unlike d_0-discordance, d_1-discordance is not hyperedge monotonic. For example, let $\boldsymbol{x} = (0, 1, 0.5)$ and consider the hyperedges $e = \{1, 2\}$ and $e' = \{1, 2, 3\}$. We have that $d^1(e, \boldsymbol{x}) = 0.5 > 0.25 = d^1(e', \boldsymbol{x})$, even though $e \subset e'$. One can interpret node 3's role in the interaction as that of a mediator who reduces the amount of discordance, thereby potentially allowing an update to take place that otherwise could not.

We employ DW-type (i.e., asynchronous) updates. In each step, we randomly select a hyperedge e from E according to some probability distribution. For mathematical convenience, we use the uniform distribution over E. If the discordance $d(e, \boldsymbol{x})$ is less than the confidence bound c, then the nodes $i \in e$ update their opinions x_i to the mean opinion $\bar{x}_e$; otherwise, their opinions do not change. That is, if we select hyperedge e at time t, then for

all $i \in e$, we have that

$$x_i(t+1) = \begin{cases} \bar{x}_e(t)\,, & \text{if } d(e, \boldsymbol{x}) < c \\ x_i(t)\,, & \text{otherwise} \end{cases} \tag{4.4}$$

and $x_j(t+1) = x_j(t)$ for all $j \notin e$.

If the hypergraph is a simple graph (i.e., if $|e| = 2$ for all $e \in E$), this generalized BCM reduces to a standard DW model with a rescaled confidence bound c. This rescaling is due to the difference in discordance functions: the standard DW model uses the absolute value $|x_i - x_j|$ of the difference of opinions, whereas our model uses $\frac{1}{2}(x_i - x_j)^2$ from Equation (4.1).

4.2.1 Preliminary Analysis

The sequence $\boldsymbol{x}(0), \boldsymbol{x}(1), \boldsymbol{x}(2), \ldots$ of opinion configurations is a discrete-time continuous-state-space Markov chain, where the evolution process is defined by the hypergraph (V, E), the confidence bound c, and the discordance function d. An opinion configuration $\boldsymbol{x}$ is an *absorbing state* if $\boldsymbol{x}(t') = \boldsymbol{x}(t)$ for all $t' \geq t$. For a configuration $\boldsymbol{x}$ to be an absorbing state, $d(e, \boldsymbol{x}) \in \{0\} \cup [c, \infty)$ for all $e \in E$, so each hyperedge is either in consensus or is too discordant.

Henceforth, We use the discordance function $d = d_1$ from Equation (4.1), set the confidence bound $c = 1$, and focus on the case in which our hypergraph is a *complete hypergraph* $E = \{e \in \mathcal{E} |\ |e| \geq 2\}$. A peculiar property of our model is that if a hyperedge e non-trivially updates at time t, then the discordance of each hyperedge $e' \supset e$ decreases (i.e., $d(e', \boldsymbol{x}(t+1)) < d(e', \boldsymbol{x}(t))$). This follows from the following lemma.

Lemma 1. *Let C be a collection of n real values $\{x_1, x_2, \ldots, x_n\}$, and let $C' = \{x_{i_1}, x_{i_2}, \ldots, x_{i_\ell}\}$ be some subcollection of C. Construct a new collection D by taking the union of $C \setminus C'$ and ℓ copies of the mean $\bar{x}_{C'} = \frac{1}{\ell}(\sum_{j=1}^{\ell} x_{i_j})$ of C'. The sample variances satisfy $s^2(D) \leq s^2(C)$, where equality holds if and only if $C = D$.*

Proof. The collections C and D have the same mean $\bar{x}_C$. We have

$$(n-1)(s^2(C) - s^2(D)) = \sum_{j=1}^{\ell} (x_{i_j} - \bar{x})^2 - \ell(\bar{x}_{C'} - \bar{x})^2.$$

Expanding the the right-hand side's second term, we get

$$\sum_{j=1}^{\ell}(x_{i_j}-\bar{x})^2-\frac{1}{\ell}\left(\sum_{j=1}^{\ell}(x_{i_j}-\bar{x})\right)^2.$$

Define $y_j := x_{i_j}-\bar{x}$ to simplify the notation and write

$$\sum_{j=1}^{\ell}y_j^2-\frac{1}{\ell}\left(\sum_{j=1}^{\ell}y_j\right)^2.$$

Expanding the second term further and simplifying yields

$$\frac{1}{\ell}\left((\ell-1)\sum_{j=1}^{\ell}y_j^2-2\sum_{j=1}^{\ell}\sum_{k=j+1}^{\ell}y_jy_k\right)=\frac{1}{\ell}\left(\sum_{j=1}^{\ell}\sum_{k=j+1}^{\ell}(y_j-y_k)^2\right)\geq 0\,.$$

We have equality if and only if $y_1=\ldots=y_\ell$, which proves the lemma. □

As a direct consequence of Lemma 1, the discordance for the hyperedge $e=V$ decreases monotonically as the system evolves. Suppose that we independently seed each individual's opinion with a random number that we draw from a normal distribution with mean μ and variance σ^2. The distribution of the discordance of hyperedges follows a mixture of scaled chi-squared distribution, and the expected discordance is σ^2 [Coc34]. If $\sigma^2<\gamma$ for some constant $\gamma\approx\left(1-\frac{4}{9N}\right)^{-3}$ that relates the mean and median of the mixed scaled chi-squared distribution, then most hyperedges can update on average. However, if $\sigma^2>\gamma$, then most hyperedges cannot update on average.

In the $N\to\infty$ limit, one can approximate the distribution of hyperedges' initial discordance by a normal distribution [HM13] with a mean of about σ^2 and a variance of about $\frac{2\sigma^4}{n}$. The exact distribution of initial discordance has a slight positive (right) skew. Therefore, for the complete hypergraph with N nodes and opinions seeded independently by taking $x_i(0)\sim\mathcal{N}(0,\sigma^2)$, we expect a phase transition in convergence time at $\sigma=1$ as $N\to\infty$. In Section 4.2.2, we numerically examine the effect that σ^2 has on convergence time.

One can also consider the "continuum" formalism from Ref. [BKR03]. Consider a hypergraph that has every hyperedge of size $\ell\in L\subset\{2,\ldots,n\}$. Let $P(x,t)dx$ be the fraction of nodes that have opinions in the interval $[x,x+dx]$ at time t. The distribution $P(x,t)$ evolves

according to the rate equation

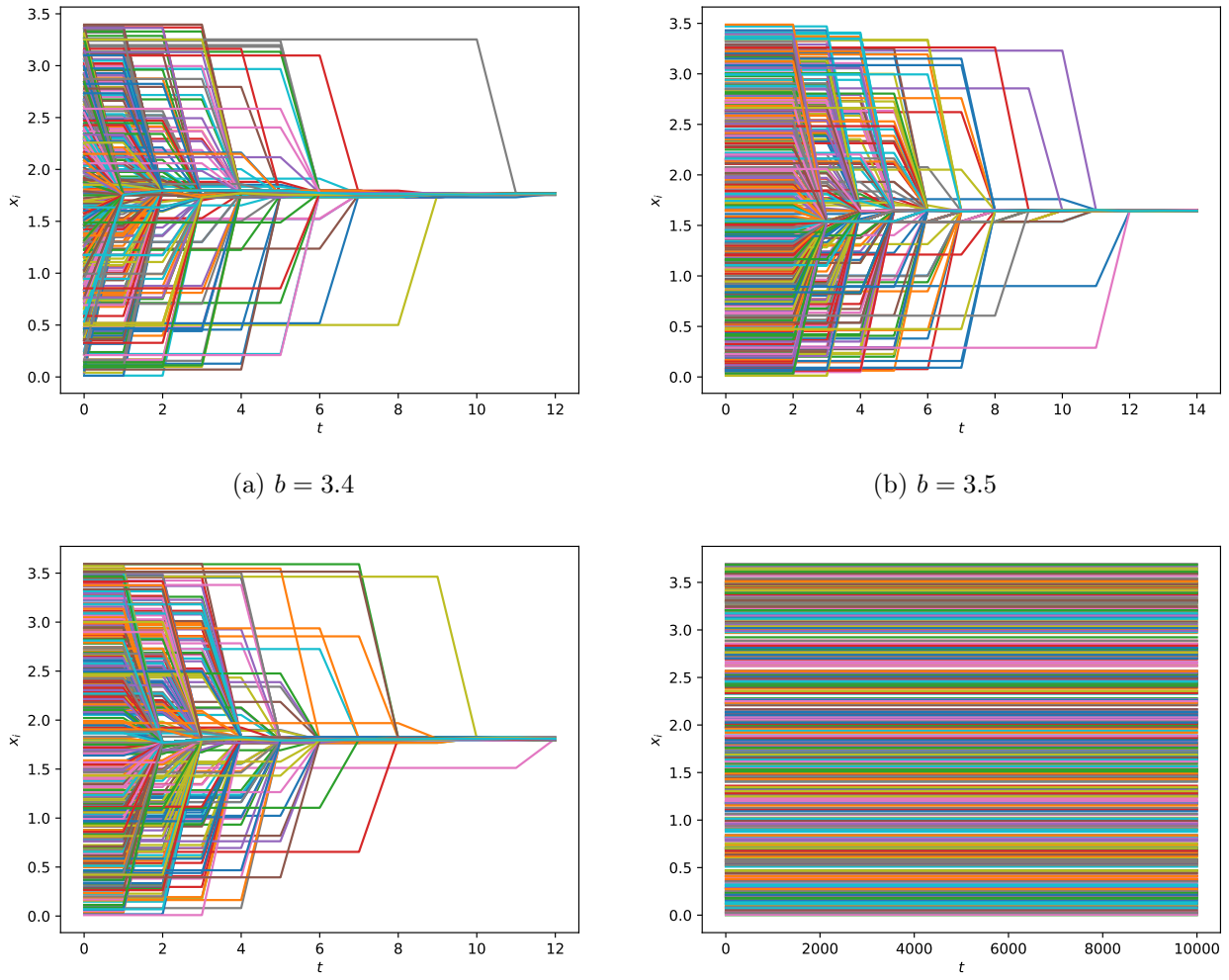
$$\frac{\partial}{\partial t}P(x,t) = \sum_{\ell \in L} \int \cdots \int_{\sum_{j=1}^{\ell}(y_j - \bar{y})^2 < \ell - 1} dy_1 \cdots dy_\ell P(y_1,t) \cdots P(y_\ell,t) \left[\delta(x-\bar{y}) - \delta(x-y_1)\right]. \tag{4.5}$$

The ℓ-fold integrals in the summand are over all ℓ-tuples of points whose sample variance is less than 1. The δ functions reflect the gains (from nodes that update their opinion to $\bar{y}$) and losses (from nodes that update their opinions away from their current values) from the update process. One of our current goals is to study special cases of this integro-differential equation.

4.2.2 Simulations

In our ongoing work, we are also conducting Monte Carlo simulations of our model. We use a complete hypergraph, the discordance function $d = d_1$ from Equation (4.1), and the confidence bound $c = 1$. In Figure 4.1, we show some simulations of the evolution of opinions for a complete hypergraph with 500 nodes and opinions seeded independently using $x_i(0) \sim \mathcal{U}(0,b)$. We allow the system to run until $d(V,\boldsymbol{x}) < 10^{-5}$ or until $t = 10^4$. The variance of the uniform distribution $\mathcal{U}(0,b)$ is $\frac{1}{12}b^2$, so we expect that if $b \gg \sqrt{12} \approx 3.5$, the system takes much longer to converge than if $b \lessapprox 3.5$. Because we are drawing a hyperedge uniformly over E in each update, the size the hyperedge (i.e., $|e|$) in each update approximately follows the binomial distribution $B(n, 1/2)$.

In Figure 4.2, we plot the empirical convergence time t^*, which is the earliest time that the system satisfies $d(V,\boldsymbol{x}) < 10^{-5}$. If the system does not reach such an opinion configuration by $t = 10^4$, we record t^* as 10^4. We simulate a system on a complete hypergraph with $50,000$ nodes and opinions seeded independently using $x_i(0) \sim \mathcal{N}(0,\sigma^2)$ for $\sigma \in [0.9, 1.1]$ with a step size of $\Delta\sigma = 0.004$. For each value of σ, we perform 20 trials. The black curve indicates the mean of t^* over the 20 trials, and the blue area depicts one standard deviation from the mean. The plot is consistent with the possibility of a phase transition in convergence time as a function of σ.



(a) $b = 3.4$

(b) $b = 3.5$

(c) $b = 3.6$

(d) $b = 3.7$

Figure 4.1: Simulations of our BCM on a complete hypergraph with 500 nodes and opinions seeded independently using $x_i(0) \sim \mathcal{U}(0, b)$. The curves indicate the evolution of opinions in time. We stop the simulation when either $d(V, \boldsymbol{x}) < 10^{-5}$ or $t > 10^4$. In panels (a)–(c), the system converges in the sense that $d(V, \boldsymbol{x}) < 10^{-5}$; in panel (d), the system does not converge before $t = 10^4$.

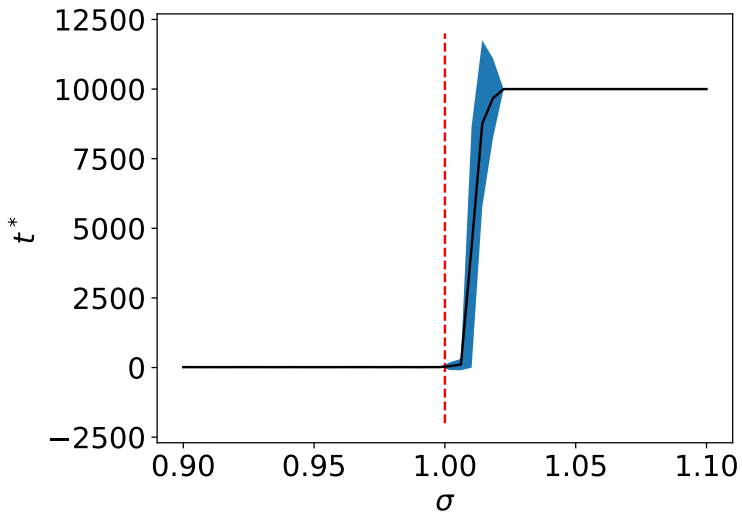


Figure 4.2: Empirical convergence time for our BCM on a complete hypergraph with $50,000$ nodes and opinions seeded independently using $x_i(0) \sim \mathcal{N}(0, \sigma^2)$ for $\sigma \in [0.9, 1.1]$ with a step size of $\Delta\sigma = 0.004$. The empirical convergence time t^* is the first time that the system reaches an opinion configuration satisfies $d(V, \boldsymbol{x}) < 10^{-5}$. If the system does not reach such an opinion configuration by $t = 10^4$, we record t^* as 10^4. We simulate 20 trials for each σ. The black curve indicates the mean of t^* over the trials, and the blue area depicts one standard deviation from the mean. We include a dashed red line at $\sigma = \left(1 - \frac{4}{9n}\right)^{-3/2} \approx 1.00$ for reference.

4.2.3 Current Plans

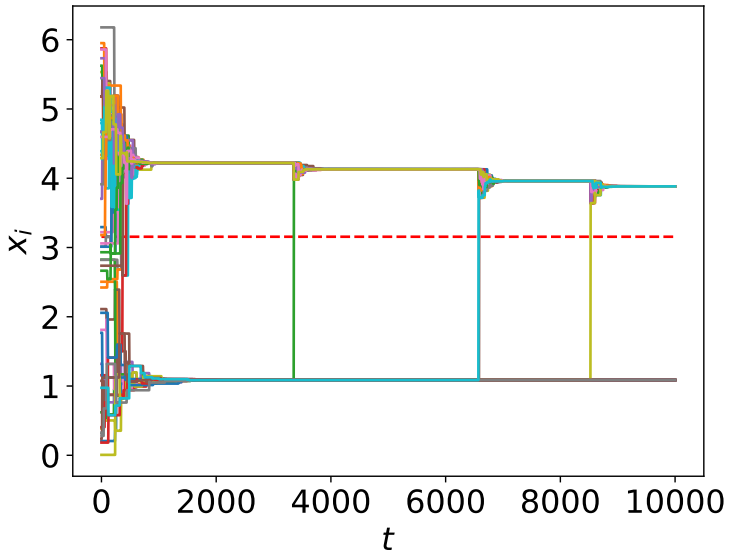
We expect that BCMs on hypergraphs with nontrivial architectures (e.g., heterogeneous degree distribution) may yield a richer landscape of behaviors than those on complete hypergraphs. To pursue this, we will examine BCMs seeded with random hypergraph models such as the hypergraph configuration model defined in Ref. [Cho19].

We also plan to investigate different mechanisms for hyperedge selection in each time step. In Section 4.2.2, we used a uniform distribution over the hyperedge set. This mechanism favors interaction that comprise nearly half of the population. A possibly more realistic distribution is one in which the sizes of the selected hyperedges follow a power-law distribution. In Figure 4.3, we show three case studies that are similar to those in Figure 4.1, but rather than choosing a hyperedge uniformly at random from E, we first select a hyperedge size ℓ using a power-law distribution

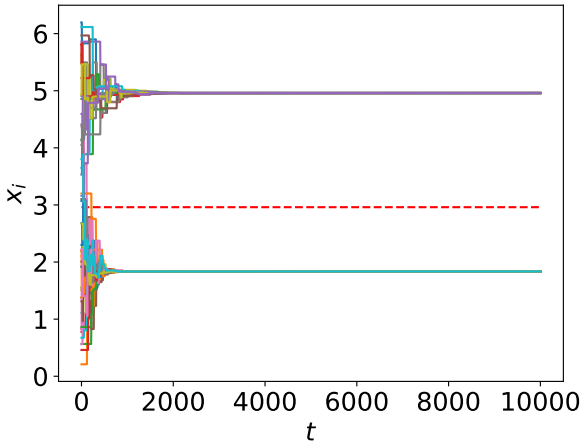
$$P(\ell = 1 + k) = \frac{1}{\zeta(\alpha + 1)k^{\alpha+1}} \text{ for } k = 1, 2, \ldots \tag{4.6}$$

with $\alpha = 4$ and then choose a hyperedge uniformly at random from $\{\, e \in E \mid |e| = \ell \,\}$.[2] In our three case studies, we observe that opinions converge into distinct clusters. In Figure 4.3a, we observe that individuals "jump" from one opinion cluster to another. This type of phenomenon cannot happen in the original DW model, because opinion clusters are separated by a distance that is larger than the confidence bound [Lor07]. However, in our hypergraph extension of BCMs, such jumps occur if one chooses to update a hyperedge that has sufficiently many nodes from one cluster and sufficiently few (but at least one) from another such that the discordance is smaller than the confidence bound.

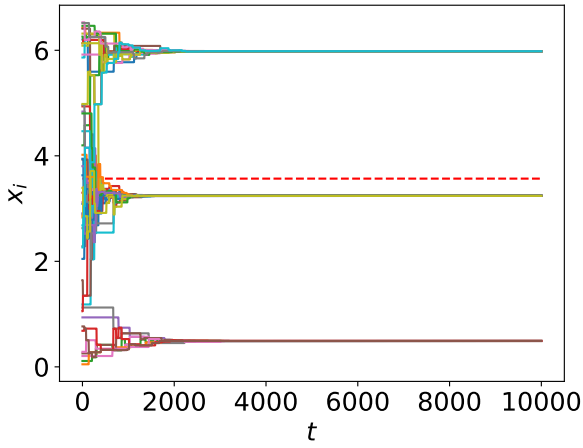
[2]If we sample $\ell > N$, we replace it with $\ell = N$.



(a) $b = 6.2$



(b) $b = 6.4$



(c) $b = 6.6$

Figure 4.3: Simulations of our BCM with power-law-distributed ($\alpha = 4$) meeting size. We use a complete hypergraph with 50 nodes and opinions seeded independently using $x_i(0) \sim \mathcal{U}(0, b)$. We simulate until $t = 10^4$. In each plot, the dash red line indicates the population's mean opinion.

CHAPTER 5

Applications of Topological Data Analysis to IC Layouts

In this chapter, I give an extended presentation of original research that was published as Ref. [KDC19], which I co-authored with Vito Dai and Luigi Capodieci of Motivo, Inc. All of the data sets used in this chapter are owned by Motivo, Inc.

5.1 Background

5.1.1 A Challenge in Integrated-Circuit Design

Integrated circuit (IC) manufacturing yield is the result of the interaction of complex, high-dimensional layout configurations with process parameters, materials, tool settings, and random events such as photon shot noise and particle defects. When yield-limiting layout configurations enter an IC fabrication plant, engineers may observe physical defects and electrical failures of the IC. These defects generally occur near the tolerance limits of photolithography systems that 'print' the IC using light. An example of a tolerance limit is the minimum separation distance between adjacent electronic components, such as transistors or interconnect wires. However, for an IC with a complex design with many potential causes of failure, merely observing the minimum separation of electronic components is insufficient to ensure that the IC is manufacturable.

An IC (also called a 'chip') is a set of electronic circuits on a small, flat piece of semiconductor material. In the last 60 years, there has been an immense increase in computing power due in no small part to advances in IC technology [Cer03]. An observation known as Moore's law (named after engineer Gordon Moore) describes the doubling that occurs nearly

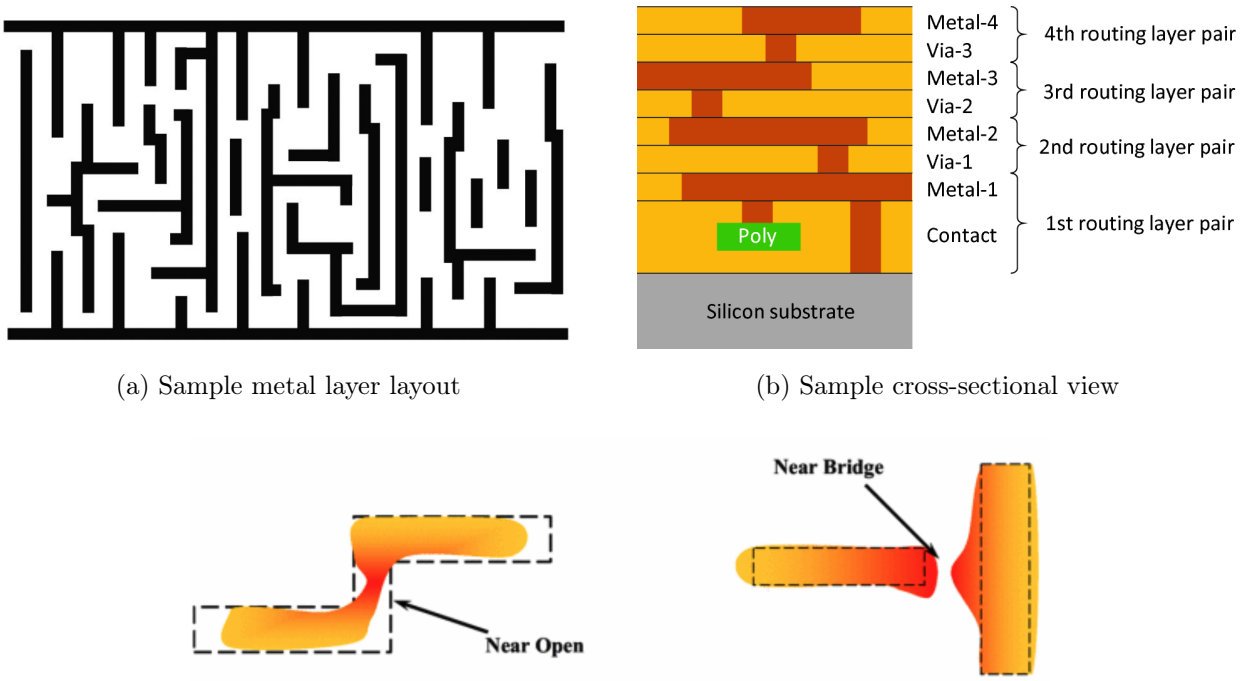
every two years in the number of electronic components per integrated circuit [Sha20].[1] To keep pace with Moore's law, IC manufacturers strive to place a progressively larger number of transistors on progressively smaller chips.

Manufacturing a chip is a long, multi-stage process that involves expertise in many scientific and engineering fields. The specific problem that we explore occurs at the interface between the design stage and fabrication. In the design stage of IC, a layout engineer places and connects all of the components that make up a chip in a kind of blueprint (see Figure 5.1a). This blueprint, called an *IC layout*, represents an IC using planar geometric shapes that correspond to the patterns of metal, oxide, or semiconductor layers that make up the components of a chip. An IC layout must meet certain criteria, including ones based on size, performance, and manufacturability [LGS99, LS20]. An IC layout meets the manufacturability criterion if it can be accurately and reliably fabricated, given the expected variability of the photolithography process. Manufacturability is a quality that can be measured through *yield*, which is the ratio of the number of functioning chips to the total number of fabricated chips. For simplicity, we sometimes treat manufacturability as a binary quality in the sense that we call an IC layout 'manufacturable' if it meets some desired yield threshold.

Because modern chips have billions of electronic components, IC layout engineers use computer-aided design (CAD) systems to automate parts of the design process [YK82, RSS85, MCF96]. However, for the manufacturability criterion, traditional CAD systems check against design rules that only specify simple conditions, such as having a minimal separation between adjacent electronic components [LS20].[2] These design rules account only for 2–15% percent of the IC layout [DTX17]. Therefore, when an IC layout moves from the design stage to the testing stage, the manufacturability of 85–98% of the IC layout is unknown. The testing stage involves printing a portion (5–50%) of a chip to check for proper

[1]Moore's initial 1965 observation was a doubling every year, but he revised it in 1975 to a doubling every two years [Moo06].

[2]Some CAD systems' design rules check for more complicated rules, but these involve compound conditions that reduce the speed of the CAD system and are difficult for humans to understand [TDC14].



(a) Sample metal layer layout

(b) Sample cross-sectional view

(c) Illustration of defects

Figure 5.1: Sample layout, cross-sectional view, and potential defects in an IC. In (a), we show a sample of one layer (specifically, one metal layer), in an IC layout. In (b), we show a sample cross-sectional view of an IC layout. In (c), we show an illustration of potential defects that can occur. The dotted lines represent the intended shape from an IC layout, and the reddish-orange shapes indicate potential output in manufacturing. A 'near open' is a type of defect in which a feature nearly disconnects and a 'near bridge' is a type of defect in which disconnected features nearly connect. Panel (a) is reproduced from *Principles of Lithography*, Figure 10.24 [Lev05], with permission from SPIE. Panel (b) is reproduced from *Technology Know-How: From Silicon to Devices*, Figure 2.24 [LS20], with permission from Springer. Panel (c) is reproduced from "Modeling manufacturing process variation for design and test", Figure 2 [KS11], with permission from IEEE © 2011 .

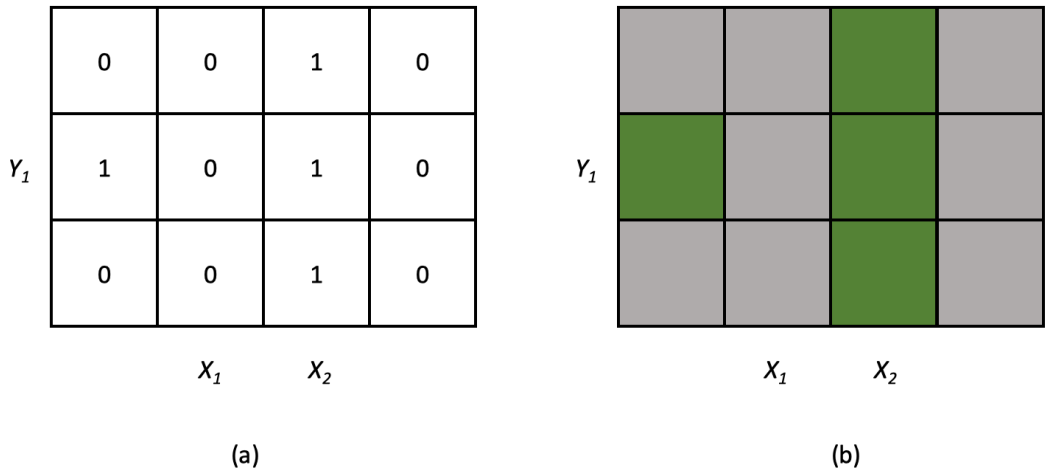


Figure 5.2: An example of a pattern that is used in pattern matching. In (a), we show the bitmap representation of the pattern. In this representation, we denote an 'occupied' cell by 1 and an 'empty' cell by 0. In (b), we show the color-coded representation of the same pattern.

functionality [DTX17]. This provides additional information about manufacturability. However, this does not guarantee manufacturability of the complete layout, and the process of iterating back and forth between the testing stage and the design stage is both costly and slow.

To augment traditional design rules, some CAD systems have begun to incorporate a technique called *pattern matching* [CFP17, LS20]. Pattern matching is a geometric approach that involves searching an IC layout for snippets of the layout that have a certain arrangement of electronic components. A *pattern* is an abstract arrangement of electronic components, and pattern matching finds all configurations in an IC layout that fit the pattern [YPM15, TDC14]. In Figure 5.2, we give an example of a pattern, which we represent in two different ways. In the bitmap (i.e., a binary matrix) representation, each cell is annotated either with a 1 to indicate an occupied cell or with a 0 to indicate an empty cell. Neither the heights nor the widths of the cells are specified. We call a specific realization of a pattern with all of the heights and all of the widths specified a *configuration*.

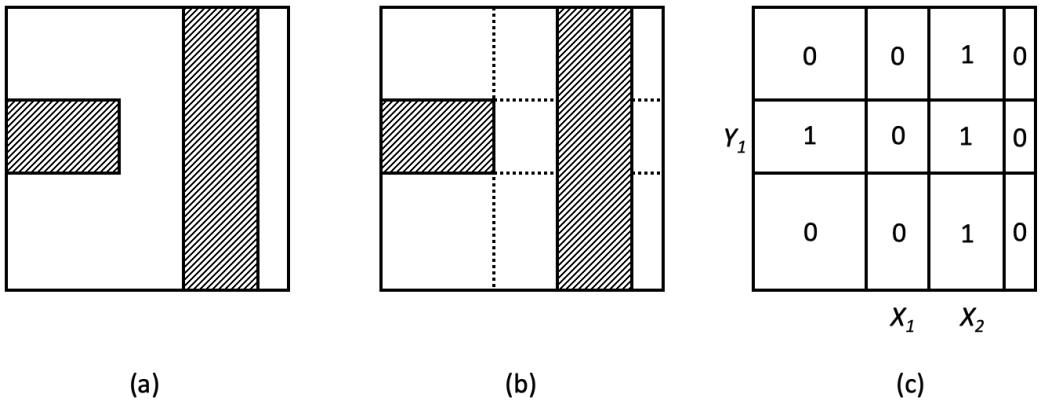


Figure 5.3: Identifying snippets of an IC layout with a pattern. In (a), we show a snippet of the IC layout. We decompose the snippet into cells in (b). In (c), we convert this decomposition into a bitmap that encapsulates the pattern.

In Figure 5.3, we show a how a snippet of an IC layout is matched to a pattern. We can convert a snippet (i.e., a rectangular patch) of layout to its bitmap representation by first decomposing the layout representation into rectangular cells that are aligned with the edges of the shaded regions (see Figure 5.3b). In this example, there are 3×4 rectangular cells, which we then convert into a 3×4 bitmap in which a shaded 'occupied' cell is recorded as a 1 and an unshaded 'empty' cell is recorded as a 0. Because we use a rectangular decomposition, within each row of cells, all of the cell heights are the same. Likewise, within each column of cells, all of the cell widths are the same. Thus, for a pattern with $n \times m$ cells, one can specify n heights and m widths. However, as we illustrate in Figure 5.4, there are only $(n-2)+(m-2)$ *critical dimensions*; that is, there are only $(n-2)+(m-2)$ dimensions that meaningfully describe the snippet. Specifically, there are $n-2$ critical heights and $m-2$ critical widths, because the heights of the cells in the top row and the bottom row and the widths of the cells in the leftmost column and the rightmost column can be shrunk arbitrarily small and still match a given pattern (see Figure 5.4). Therefore, the critical dimensions represent the parameters of interest in pattern matching. For our purposes, when a configuration in an IC layout matches a pattern that we seek, we record only the critical dimensions $(X_1, \ldots, X_{n-2}, Y_1, \ldots, Y_{m-2})$, where the first $n-2$ coordinates

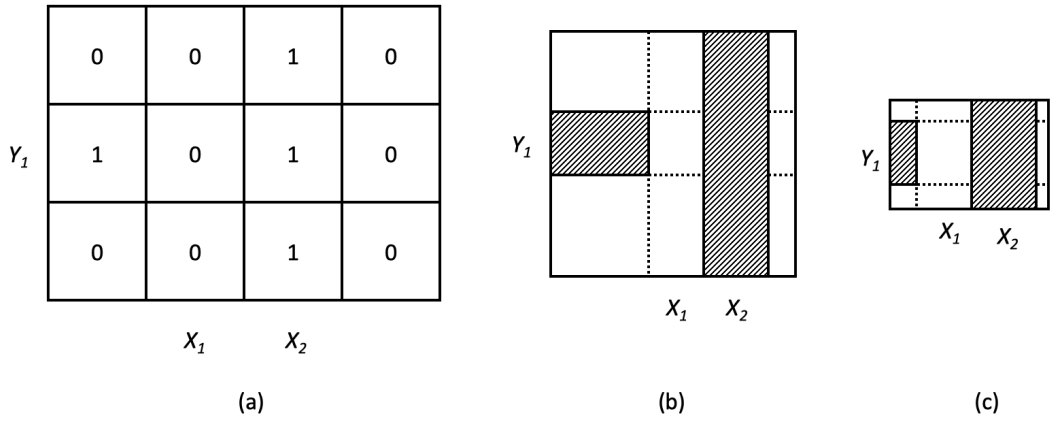


Figure 5.4: Critical dimensions of a pattern. We match the pattern in (a) to the configuration in (b). If we zoom in on the configuration as in (c), we see that the pattern still matches the zoomed-in configuration. To avoid counting essentially the same configuration multiple times, we record only the critical dimensions (X_1, X_2, Y_1).

are the critical widths and the last $m - 2$ coordinates are the critical heights. Therefore, configurations for patterns with different numbers of critical dimensions are embedded in spaces of different dimensions.

For each layer in an IC layout and for each pattern, we obtain a multiset of k-tuples, where k is the number of critical dimensions of the pattern. Each k-tuple in the multiset corresponds to a configuration in the IC layout, and the k-tuple records the critical dimensions of the configuration. The reason that we obtain a multiset in general is that the same configuration may occur in multiple places in an IC layout. We refer to the multiset of k-tuples as the *coverage* of the pattern. Although the coverage data form a multiset of points rather than a set, we still use the term *point cloud* (a term that typically refers to sets) to describe it.

To understand how pattern matching is used to aid in the design process, consider a simple example. Suppose that we collect data sets from an IC layout that we know to be manufacturable. Looking specifically at the coverage data for the two patterns that

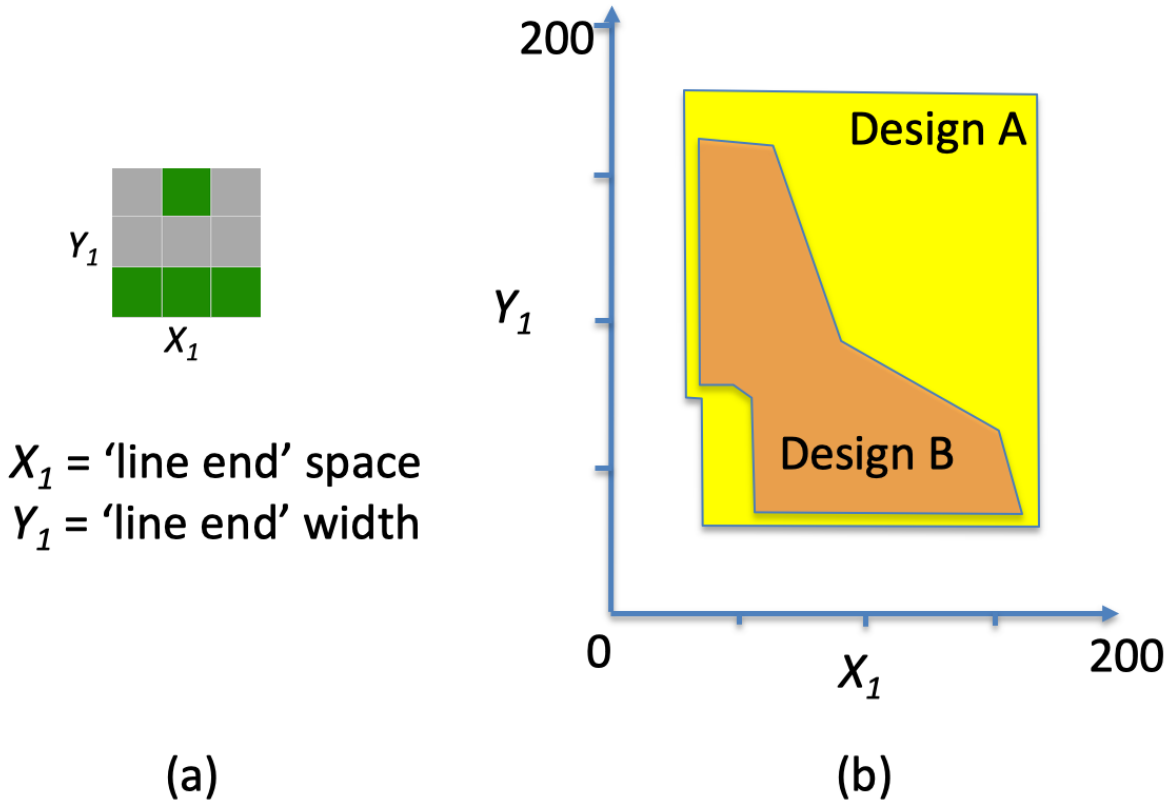


Figure 5.5: Comparing coverage data of a pattern between IC layouts. In (a), we show the 'line ends' pattern. It has two critical dimensions, X_1 and Y_1. In (b), we show a cartoon plot of the coverage of the pattern in two toy example IC layouts, 'Design A' and 'Design B'.

correspond to the matrices

$$\begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},$$

we have a record of all measurements of horizontal and vertical separation between adjacent electronic components. By taking the minimum of all of these measurements (and knowing that the data come from a manufacturable IC layout) allows us to infer an upper bound on the minimal separation distance needed between adjacent components. Another way in which pattern matching can be useful in the design process is by comparing the coverage of patterns between multiple IC layouts. In Figure 5.5, we show a cartoon plot of the coverage of the 'line ends' pattern for two toy examples of IC layouts. If Design B's coverage is properly contained in Design A's coverage for all patterns (as illustrated in the figure), we can reasonably conclude that Design B is manufacturable if Design A is manufacturable.

Unfortunately, coverage data is more complicated than in the toy example of Figure 5.5. The data for a pattern with k critical dimensions in an IC layout is a finite multiset of points in a k-dimensional space. We depict this in Figure 5.6, in which we indicate the multiplicity of a point by the size of its marker. We use this convention of scaling markers according to multiplicity for all subsequent scatter plots in this chapter. Because of the size and complexity of IC layouts, it is unlikely that the coverage of one pattern in one layout is a proper subset of the coverage of the same pattern in a different layout. In practice, one asks if the configurations that appear in one IC layout are similar to the ones that appear in another. For concreteness, we consider the following problem: suppose that we know the coverage data C for a pattern from a manufacturable IC layout and we want to know whether a configuration c' that we do not observe in that layout (i.e., $c' \notin C$) is *yield-limiting*, meaning that if it were to be included in an IC layout, then the layout would have strictly smaller yield. It is impossible to know with certainty whether c' is yield-limiting based on knowledge of C. However, we can perform some probabilistic classification, in which we infer that if c' is near some $\{c_i\} \subset C$ (viewing c' and the elements of C as points in a k-dimensional space equipped with some metric), then it is more likely to not be yield-limiting than if c' were not

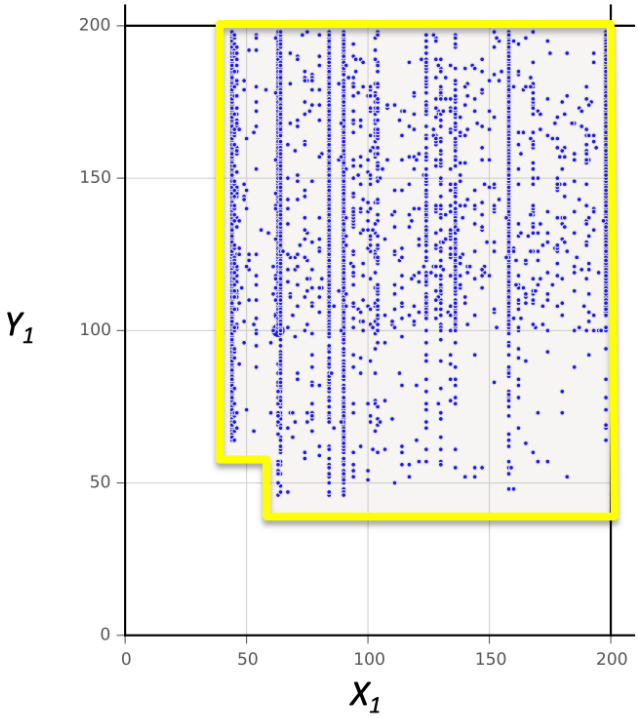


Figure 5.6: Scatter plot of the coverage data for the 'line ends' pattern from a manufacturable IC layout. We use blue dots to represent points and we scale the dot size in proportion to the point's multiplicity. The region outlined yellow indicates what we infer to be 'safe' configurations based on their proximity to blue points.

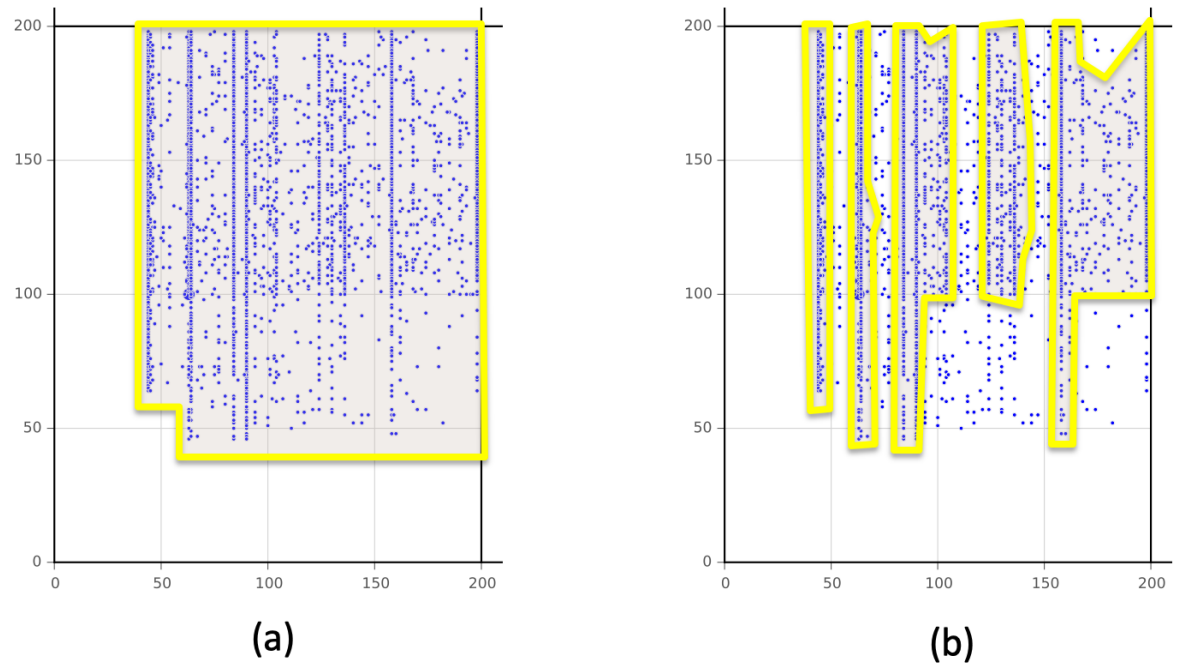


Figure 5.7: The choice of risk tolerance affects the inferred safe region from coverage data. Panels (a) and (b) show the same coverage (in blue) but different safe regions (outlined in yellow). In (a), we show a more risk-tolerant, larger safe region. In (b), we show a less risk-tolerant, smaller safe region.

near any $c \in C$. The region outlined in yellow of Figure 5.6 depicts what we may infer to be a 'safe' region. That is, we may suspect configurations in this region to be manufacturable with some sufficiently high probability.

Depending on the risk tolerance, one may infer different safe regions (see Figure 5.7). It is desirable to use a 'multiscale' approach that can convert coverage data into safe regions efficiently without a pre-specified risk tolerance. Further, for patterns with three or more critical dimensions, it is difficult for humans to visualize scatter plots of the coverage. Two-dimensional projections may reveal 'holes' in a safe region as in Figure 5.8, but they cannot reveal higher-dimensional 'voids'. Consequently, we need an approach that also works in three or more dimensions.

In this chapter, we use a data set that was generated by Motivo, Inc. from a manufac-

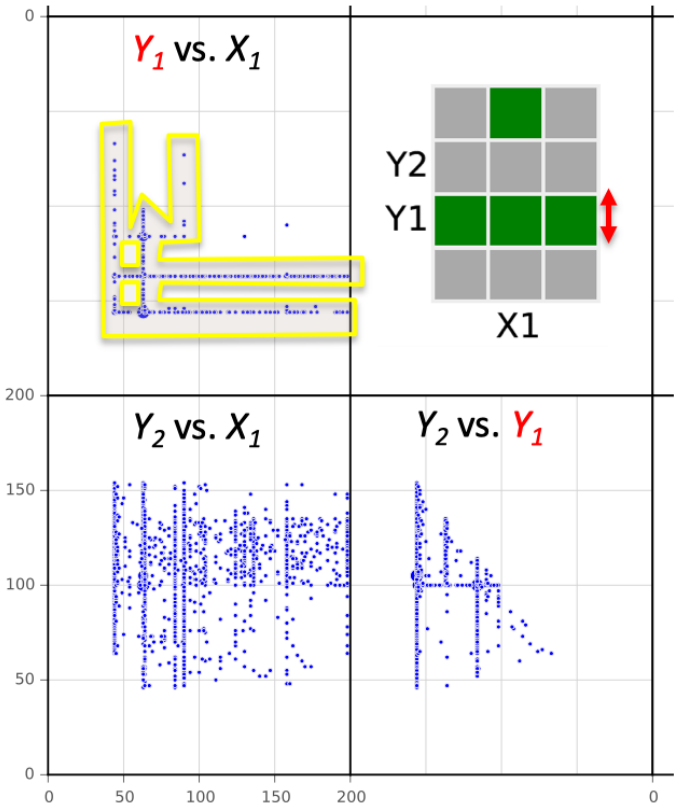


Figure 5.8: A pattern with three critical dimensions (X_1, Y_1, Y_2). We show three projected scatter plots of its coverage data. For the Y_1 versus X_1 scatter plot, we outline a potential safe region in yellow; it appears to have two 'holes'.

turable IC layout that consists of multiple metal layers (see Figure 5.1b). We focus on layers 'metal3', 'metal4', and 'metal5'.[3]

5.1.2 A Potential Solution from Topological Data Analysis

Homology provides a framework to study 'shapes' — in particular, the structure of their holes [Hat05, Ghr14]. However, a shape is an idealized mathematical entity, whereas the coverage data from IC layouts are finite multisets of points in a k-dimensional space. To study the homology of these multisets, we turn to *persistent homology* [EH08], which is a method from topological data analysis that allows one to study the homological structures, such as connected components and holes, of a point cloud at a range of spatial resolutions. In our analysis, rather than use the full power of persistent homology, we focus solely on the existence (but not persistence) of homological structures as we vary the spatial resolution. That is, we study how the number of i-dimensional holes, known as the ith *Betti number* β_i, changes as we change a spatial resolution parameter ϵ. In Section 5.3, we show that plotting the Betti numbers of coverage data against ϵ, in what are called *Betti plots* [CHY15], provides a useful summary of coverage data for an IC layout. We also show that Betti plots allow one to quantitatively compare coverage data between multiple patterns within a single IC layout, as well as to compare coverage data of a single pattern between multiple IC layouts.

Applications of this topological approach include physical design optimization and comparative yield-risk analysis. In Section 5.2, we give a brief introduction to topological data analysis and persistent homology. This approach enables the efficient discovery of homological features in high-dimensional data. In Section 5.3, we describe an application of persistent homology to coverage data from IC layouts. In Section 5.4, we discuss results and future work.

[3]For more details on the metal layers, see Ref. [LS20].

5.2 Topological Data Analysis And Persistent Homology

The goal of topological data analysis (TDA) is to examine the structure (i.e., 'shape') of data using techniques from topology in a computationally efficient way [ZC05]. An important tool in TDA is persistent homology (PH) [OPT17]. Loosely speaking, homology is concerned with coarse-grained classification of spaces based on their 'holes'. One aspect of this classification involves identifying and counting certain homological features of a space. One may ask how many pieces (i.e., connected components) does a space consist of? One may also ask how many holes or cavities are in a space? The answers to these homological questions are given by the *Betti numbers* of the space. The ith Betti number $\beta_i(S)$ of a space S is the number of i-dimensional holes in that space, except for the 0th Betti number $\beta_0(S)$, which counts the number of connected components. For example, a torus T consists of one connected component, two one-dimensional holes, a single (two-dimensional) cavity that is enclosed by the surface, and no higher-dimensional holes. Thus, the Betti numbers for a torus are $\beta_0(T) = 1$, $\beta_1(T) = 2$, $\beta_2(T) = 1$, and $\beta_j(T) = 0$ for $j \geq 3$. One computes the ith Betti number β_i of a space S by finding the rank of the ith homology group [Hat05].

It is not obvious what one can gain by studying the Betti numbers of data sets that consist of finitely many individual observation points that are recorded in a high-dimensional space, especially ones in which we do not know an appropriate scale of resolution. The problem is even more challenging if there are multiple scales or if the data are corrupted by noise. The pipeline of PH provides a way of transforming finite data sets into a sequence of spaces that one can then study using homology at all resolution scales simultaneously and in a way that is robust with respect to small perturbations in input data [OPT17].

5.2.1 Mathematics of Persistent Homology

We present a mathematical formulation of PH in a way that is geared to the analysis of coverage data of IC layouts. We begin with a set S of points in some d-dimensional Euclidean space. We refer to S as point-cloud data". The set has a rather uninteresting topological structure; the homology of S reveals that S consists of $|S|$ connected components, with

each point in its own component, and has no higher-dimensional homological features such as holes or cavities. Therefore, rather than examining the homology of S, we study the homology of a *filtered simplicial complex* of S. A filtered simplicial complex of S is a sequence $S_{\epsilon_1} \subseteq S_{\epsilon_2} \subseteq \ldots \subseteq S_{\epsilon_N}$ of nested spaces, where each space is a 'thickening' of S [OPT17]; we explain this process of thickening in detail in Section 5.2.2. This sequence of spaces allows one to examine point-cloud data at different resolutions.

We construct each S_ϵ by connecting nearby points in the point cloud according to some rule parameterized by the *resolution parameter* ϵ. Specifically, we form *simplices* — such as edges, triangles, tetrahedra, and their higher-dimensional generalizations — between the points in the point cloud in a way that depends on the proximity of points and rules that we specify in a filtered simplicial complex. There are many filtered simplicial complex [OPT17], each with their own rules for forming simplices.

Throughout this chapter, we use a filtered Vietoris–Rips complex. In such a complex, the space S_ϵ is composed of the points of S, edges between each pair of points in S that are a distance at most ϵ apart, triangles that connects any triple of points in S that have a pairwise distance of at most ϵ, and, in general, an $(n-1)$-simplex between any n-tuple of points whose points are a distance of at most ϵ apart pairwise. An equivalent interpretation for a filtered Vietoris–Rips complex is that we thicken the point cloud by placing balls of radius $\epsilon/2$ around each point and add a $(n-1)$-simplex for each subset of n balls that intersect pairwise; see Figure 5.10 for an example. As we increase the resolution parameter ϵ from ϵ_1 to ϵ_2, where $\epsilon_1 \leq \epsilon_2$, the space grows monotonically, such that S_{ϵ_1} is contained in S_{ϵ_2}. Additionally, note that $S_0 = S$. In PH, one seeks to compute and analyze the homology of S_ϵ for a range of ϵ values, without a priori choosing a specific resolution ϵ to examine.

Persistent homology involves converting the filtered simplicial complex into a finite persistence module, which records the 'birth' and 'death' of homological features such as connected components, holes, and cavities as one increases the resolution parameter ϵ [OPT17, ELZ02, ZC05]. The original algorithm to compute PH relies on Gaussian elimination and has a worst-case time complexity that is cubic in the number of simplices [MMS11, MN13]. The problem of combinatorial explosion in the number of simplices is an active area of research [SC04].

For example, if we set ϵ equal to the diameter of a point cloud, then constructing a filtered Vietoris–Rips complex entails generating $2^n - 1$ simplices, where n is the number of data points.[4] Some approaches for reducing the computational complexity of such complexes are to perform preprocessing or to construct exponentially smaller filtered simplicial complexes that produce equivalent results [MN13].

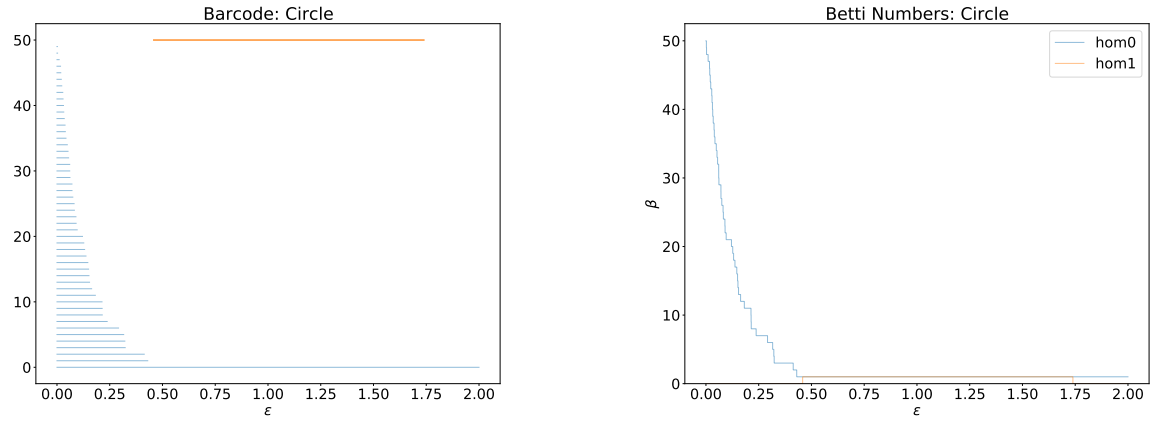
5.2.2 PH Pipeline

We briefly summarize the pipeline of PH. Beginning with a set S of data points, the first step is to convert S into a filtered simplicial complex $(S_{\epsilon_j})_{j=1}^{N}$. In our analysis of IC layouts, we use a filtered Vietoris–Rips complex with Euclidean distance. We then convert the filtered simplicial complex S_ϵ into a persistence module, which yields the births and deaths of homological features. Finally, we visualize the feature lifetimes. There are many software packages for computing PH. See Ref. [OPT17] for a comprehensive review and comparison of PH packages. We use DIONYSUS 2 [Mor] by Dmitriy Morozov of Lawrence Berkeley National Laboratory.

Two prominent ways of visualizing feature lifetimes are persistence diagrams [CEH07] and barcodes [Ghr08]. In a barcode, one represents the lifetimes of features as intervals. Each left endpoint is the birth time of a feature, and the associated right endpoint is the death time of that feature.[5] In Figure 5.9a, we show an example of a barcode for a point cloud that we generate by sampling 50 points uniformly on the unit circle. The blue intervals represent the lifetimes of 0-dimensional homological features (i.e., connected components). The orange interval represents the lifetime of a 1-dimensional homological feature (i.e., a hole). In barcodes, the intervals are stacked on top of each other with different types of features (such as connected components, holes, and cavities) separated by color or into different plots. Barcodes are unique up to reordering of the intervals within each feature

[4]The diameter of a point clouds equals the maximum distance over all pairs of its points.

[5]Because one typically computes a filtered simplicial complex on a finite range $[0, \epsilon_{\max})$, features that persist past $\epsilon_{\max}$ are recorded as having no death time. In a barcode, one usually records features that have no death time by marking the right endpoint of its interval with an arrow.



(a) Barcode

(b) Betti Plot

Figure 5.9: Barcode and Betti plot for a point cloud that we generate by sampling 50 points uniformly on the unit circle. (a) The intervals in the barcode represent lifetimes of connected components (blue) and a hole (orange). (b) The Betti plot of the same point cloud. See Figure 5.10 for a detailed explanation of how to read Betti plots.

type.

Barcodes are a convenient way of visualizing the lifetime of all features. However, they can be difficult to interpret and to compare, especially when the number of features is large.[6] Instead, we visualize the output of the PH computation using a *Betti plot*. Recall that the ith Betti number $\beta_i(S)$ of a space S is the number of i-dimensional holes in S. For a filtered simplicial complex, we have a sequence $(\beta_i(S_{\epsilon_j}))_{j=1}^{N}$, which records the number of i-dimensional holes for each space in the sequence. Equivalently, for a fixed S and for each i, we can construe $\beta_i(S_\epsilon)$ as a piecewise-constant function $\beta_i(\epsilon)$ from $[\epsilon_1, \epsilon_N] \subset \mathbb{R}_{\geq 0}$ to $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$. These functions are constant on the subintervals (i.e., *subdomains*) induced by partitioning $[\epsilon_1, \epsilon_N]$ at the points ϵ_j. That is, the subdomains of our functions are the half-open intervals $([\epsilon_j, \epsilon_{j+1}))_{j=1}^{N-1}$. A Betti plot is the graph of these piecewise-constant functions. Because of computational limitations, we compute only the first two functions in

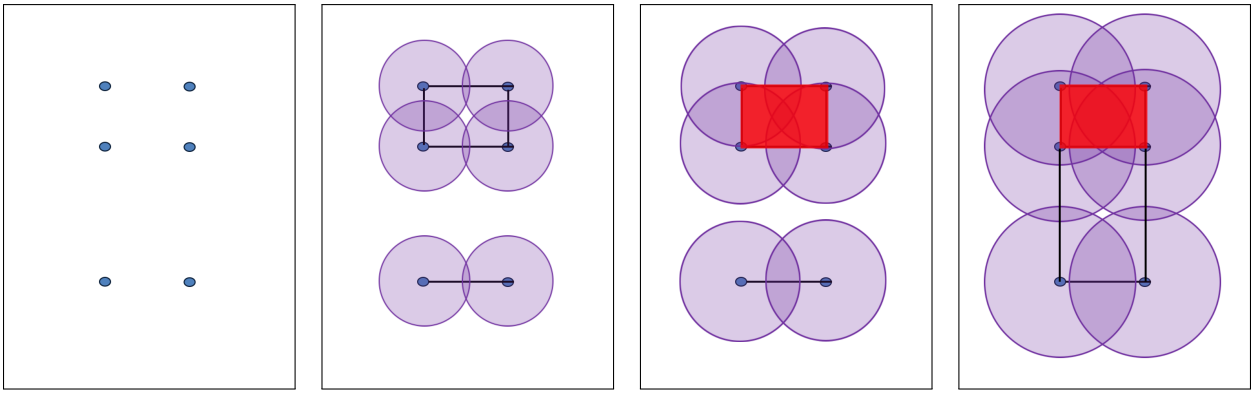
[6]Comparison of barcodes (and persistence diagrams) is an active area of research in the PH literature [KMN17, MGL18].

the sequence; they correspond, respectively, to counts of the number of components and the number of holes. For comparison, in Figure 5.9b, we show the Betti plot corresponding to the barcode in Figure 5.9a. We give a detailed explanation of how to construct a Betti plot in Figure 5.10.

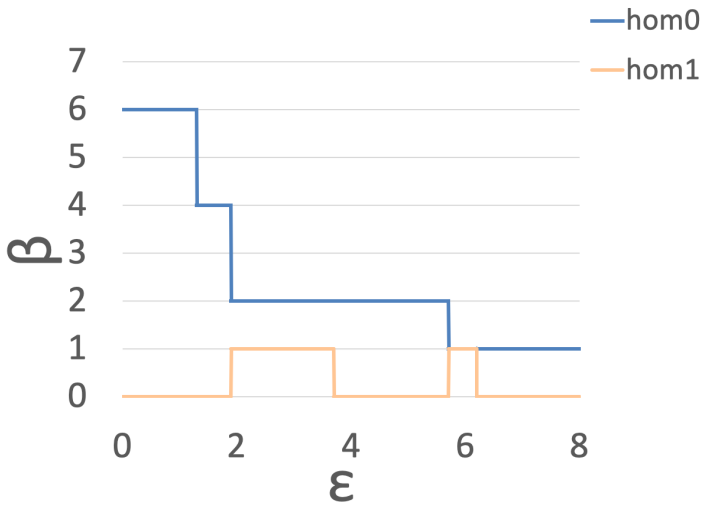
Using a point cloud of six points, we illustrate the process of thickening points using a filtered Vietoris–Rips complex and the resulting Betti plot in Figure 5.10. In Figure 5.10a, the resolution parameter is $\epsilon = 0$. Without any thickening, the six points are separated, and each constitutes its own connected component. There are no holes at this resolution. In Figure 5.10b, we set $\epsilon = 2$. We convey the thickening of the space by placing balls around the points (see the shaded purple areas). When two balls intersect, the points to which they correspond become connected by an edge. With $\epsilon = 2$, the six points are connected in such a way that there are two connected components and one hole. In Figure 5.10c, we further thicken the points by setting $\epsilon = 4$. At this resolution, the four topmost thickened points intersect pairwise (i.e., their purple shaded regions intersect pairwise), so the hole is filled in. Finally, in Figure 5.10d, we thicken the points to a resolution of $\epsilon = 6$. At this resolution, there is only one component and one hole.

In Figure 5.10e, we show the corresponding Betti plot for this filtered simplicial complex. The 0th Betti number β_0 (blue), which is the rank of the 0th homology group and corresponds to the number of components, is 6 at $\epsilon = 0$. The 1st Betti number β_1 (orange), which is the rank of the 1st homology group and corresponds to the number of holes, is 0 at $\epsilon = 0$. We observe that as ϵ increases (i.e., as points thicken, causing components to merge), β_0 decreases. We also observe that β_1 (orange curve) increases from 0 to 1 at $\epsilon \approx 1.9$ when a hole is born and decreases to 0 when that hole dies at $\epsilon \approx 3.7$. We observe a similar increase and decrease for the hole that is born at $\epsilon \approx 5.8$ and dies at $\epsilon \approx 6.2$.

The information in a Betti plot is a subset of the information in a barcode. For 0-dimensional homological features, the two types of visualization contain equivalent information. However, for higher-dimensional homological features, the information from a PH computation may not be present in a Betti plot. A simple example involves one i-dimensional feature dying at exactly the same resolution at which another i-dimensional feature is born.



(a) $\epsilon = 0$ (b) $\epsilon = 2$ (c) $\epsilon = 4$ (d) $\epsilon = 6$



(e) Betti Plot

Figure 5.10: An illustration of thickening a point cloud S that consists of six points (blue) and the resulting Betti plot. Panels (a)–(d) show different stages of the thickening process, corresponding to different resolution-parameter values: (a) $\epsilon = 0$, (b) $\epsilon = 2$, (c) $\epsilon = 4$, and (d) $\epsilon = 6$. We show the radius-$(\epsilon/2)$ balls as purple shaded disks, the 1-simplices as black lines, and the 2-simplices as red shaded areas. In panel (e), we show the Betti plot that corresponds to S. The blue and orange curves correspond to 0th and 1st Betti numbers, respectively. As a shorthand for the legends of all of our Betti plots, we use 'hom0' and 'hom1' to indicate 0-dimensional and 1-dimensional homological features.

A Betti plot does not distinguish between (1) this co-occurrence of birth and death of i-dimensional features and (2) no change in the homological structure of the space. This is unlike a barcode, which does distinguish between these.

In our analysis of IC layout data, we find Betti plots to be useful for multiple reasons. As we demonstrate in Section 5.3.2, Betti plots are robust to a certain type of subsampling, in the sense that as one takes progressively smaller samples, the general shapes of the curves are preserved but the curves becomes progressively lower-resolution approximations of the original curves. As we demonstrate in Section 5.3.3, we can use Betti plots to determine which pairs of patterns are similar and dissimilar based on their respective coverage data. Moreover, we are able to quantify the similarity of patterns by defining a distance function on Betti plots.

Betti plots are the graphs of a sequence of piecewise-constant functions $\beta_i(\epsilon)$, each corresponding to the number of i-dimensional homological features. Therefore, defining a distance between two Betti plots is equivalent to defining a distance between two sequences of functions. There are many choices for how to define a distance between functions. Through trial and error, we found the following distance function d to be suitable for our purposes. The only requirement is that the domain for both Betti plots must be the same. Let $B^1 = (\beta_0^1, \beta_1^1)$ be the two functions in the first Betti plot, let $B^2 = (\beta_0^2, \beta_1^2)$ be the two functions in the second, and let $[\epsilon, \epsilon_N]$ be the domain for all four functions. First, we convert each function f into a vector $[f(x_1), f(x_2), \ldots, f(x_M)]$ of real values, where the x_i are equispaced in $[\epsilon_1, \epsilon_N]$. In essence, we are taking M uniformly-spaced samples from our function. Ideally, we want to be reasonably confident that we sample from each subdomain of our piecewise-constant functions. Accordingly, we set $M = 1000$, which is an order-of-magnitude larger than the number of subdomains that we observe in Betti plots of coverage data. This results in two pairs of M-dimensional vectors, $(\vec{\beta_0^1}, \vec{\beta_1^1})$ and $(\vec{\beta_0^2}, \vec{\beta_1^2})$. We then take the discrete cosine transform (specifically, the orthogonal DCT-II) of each vector and then normalize them. If a vector is the 0 vector, we leave it as the 0 vector. We denote the transformed pairs of M-dimensional unit vectors by $(\vec{b_0^1}, \vec{b_1^1})$ and $(\vec{b_0^2}, \vec{b_1^2})$, respectively. We compute the distance

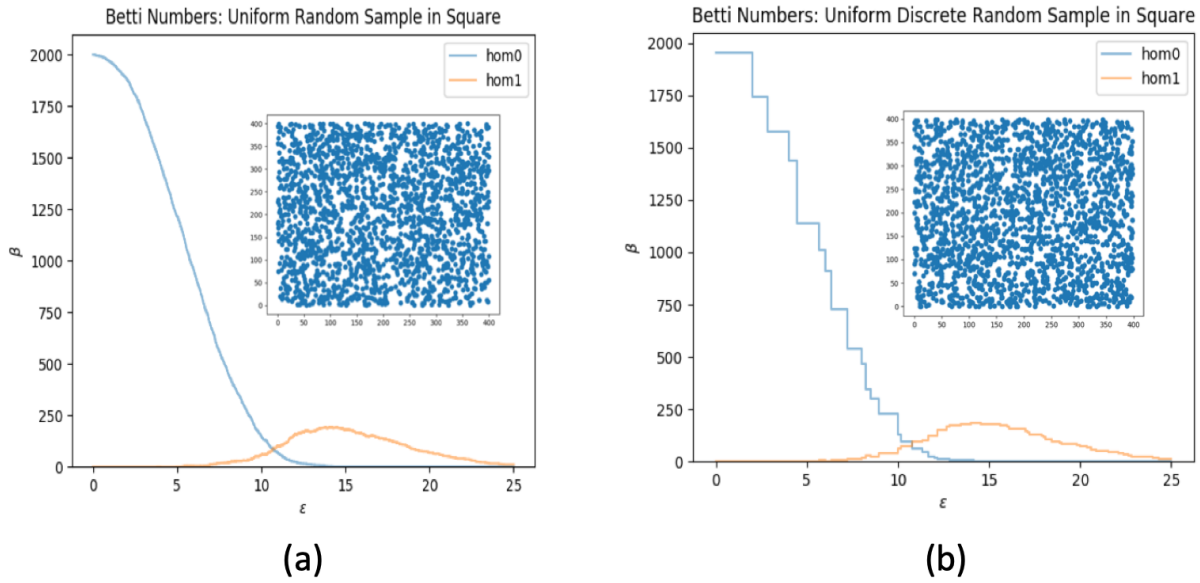


Figure 5.11: Betti plots of uniformly random point clouds on (a) a 'continuous' grid and (b) a 'discretized' grid in the region $[0, 400] \times [0, 400]$. We show scatter plots of the point clouds in the insets. The difference between the two point clouds is difficult to notice in the scatter plot, but it is discernible in the Betti plots.

between the two Betti plots as

$$d(B^1, B^2) = \frac{1}{2\sqrt{M}} \left(\left| b_0^2 - b_0^1 \right|_2 + \left| b_1^2 - b_1^1 \right|_2 \right). \tag{5.1}$$

We use this Betti-plot distance function d for all of our subsequent comparisons of Betti plots.

5.2.3 Tests on Synthetic Data

We now examine and compare the Betti plots of synthetically generated data to calibrate our understanding of how a Betti plot reflects the characteristics of a point cloud.

In Figure 5.11, we show the Betti plots and corresponding scatter plots in the inset of two synthetically generated point clouds. For Figure 5.11a, we generate a point cloud by sampling 2000 points uniformly at random with replacement in the region $[0, 400] \times [0, 400]$ using a 'continuous' floating-point grid. There are approximately 10^{18} points in this 'continuous'

grid, but the grid points are not uniformly spaced [Smi97]. This implies that the grid points are not equiprobable, even though we are sampling uniformly. The probability that we draw 2000 unique points is greater than $1 - 2 \times 10^{-8}$. For Figure 5.11b, we generate a point cloud by sampling 2000 points uniformly at random with repetition in the same region $[0, 400] \times [0, 400]$, but now we use a 'discretized' grid by rounding to the nearest integer multiple of two. In other words, we sample from the lattice $(2\mathbb{Z}/400\mathbb{Z})^2$, which has $200^2 = 40000$ equiprobable points. With this many available points to sample from, we expect approximately 1950 unique points in our sample.

One can see that the Betti curves in Figure 5.11b resemble those of Figure 5.11a qualitatively. However, one can also see that the subdomains of the piecewise-constant curves in Figure 5.11b are generally much larger than those in Figure 5.11a. To understand why this is the case, we must consider the possible pairwise distances that arise in our 'continuous' and 'discretized' grids. In our discretized grid, points cannot connect to form edges until $\epsilon \geq 2$. One can observe this phenomenon in the Betti plot of Figure 5.11b, in which, at a resolution of $\epsilon = 2$, the β_0 curve (blue) has its first 'step'. Contrast this to the continuous grid, in which points can connect to form an edge as early as $\epsilon \approx 10^{-35}$ [Smi97]. In general, the discretized grid has fewer distinct distances available than the continuous grid; this manifests as larger subdomains in Figure 5.11b than in Figure 5.11a.

We next compare two point clouds that we generate using different mechanisms. For the first point cloud, we sample points from one of three bivariate Gaussian distributions. We sample from the three distributions with equal probability. All three of the distributions have the same covariance, but we generate the centers of the three distributions uniformly at random. For the second point cloud, we simulate the paths of four Lévy-flight random walkers in the plane [SK86]. All four walkers start a point in the plane that we choose uniformly at random, and all of the walkers use the same transition probabilities that we bias in favor of horizontal movement by a factor of 5. We then 'discretize' the two point clouds as in Figure 5.11b so that the coordinates are in $(2\mathbb{Z}/400\mathbb{Z})^2$ with periodic boundary conditions. We denote the first point cloud by G for Gaussian and the second point cloud by W for walker.

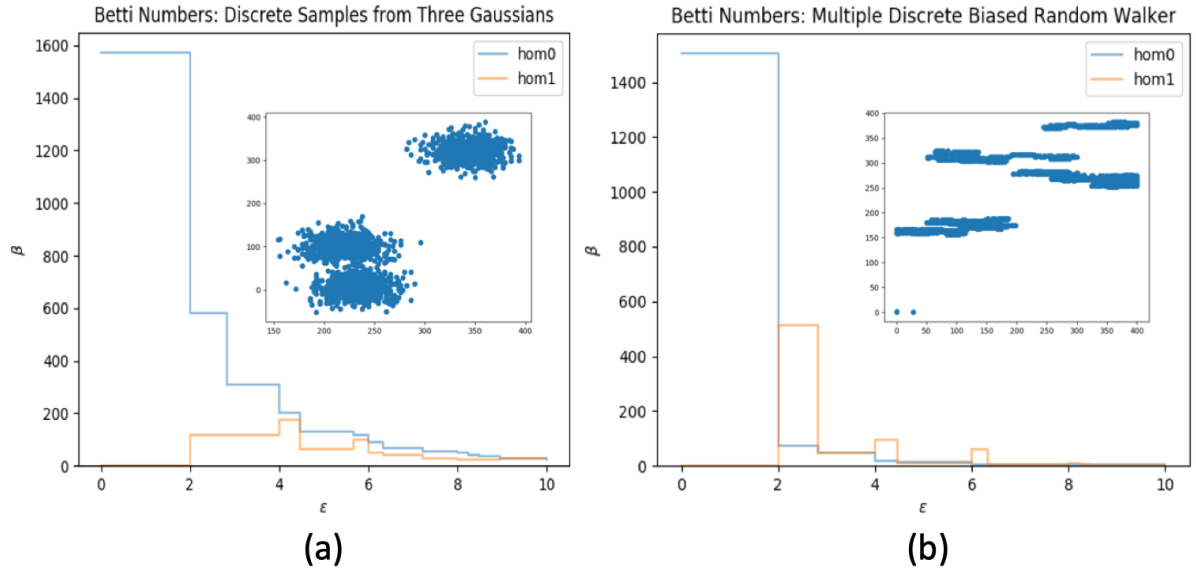


Figure 5.12: Betti plots of (a) multivariate Gaussian random points and (b) multiple random-walk paths. Both (a) and (b) are projected onto a discretized grid in the region $[0, 400] \times [0, 400]$. We show scatter plots of the point clouds in the insets.

In Figure 5.12, we show the Betti plots of the point clouds G and W; we show the corresponding scatter plots in the inset. In both Betti plots, we observe that using the discretized grid leads to large subdomains for the curves that are similar to those of 5.11b. Let us examine the relative sizes of the drop in the first 'step' of β_0 function. That is, consider the quantity

$$(\beta_0(0) - \beta_0(2))/\beta_0(0)\,. \tag{5.2}$$

For the Gaussian-generated point cloud G, the first drop is approximately 0.63; for the random-walker point cloud W, the first drop is approximately 0.93. These drops are larger than the first drop in 5.11b, which is approximately 0.10. We can interpret this quantity in Equation (5.2) as measuring a type of 'clustering' in the point clouds, according to which the point cloud W is more clustered than G, which in turn is more clustered than the uniform point cloud of 5.11b.

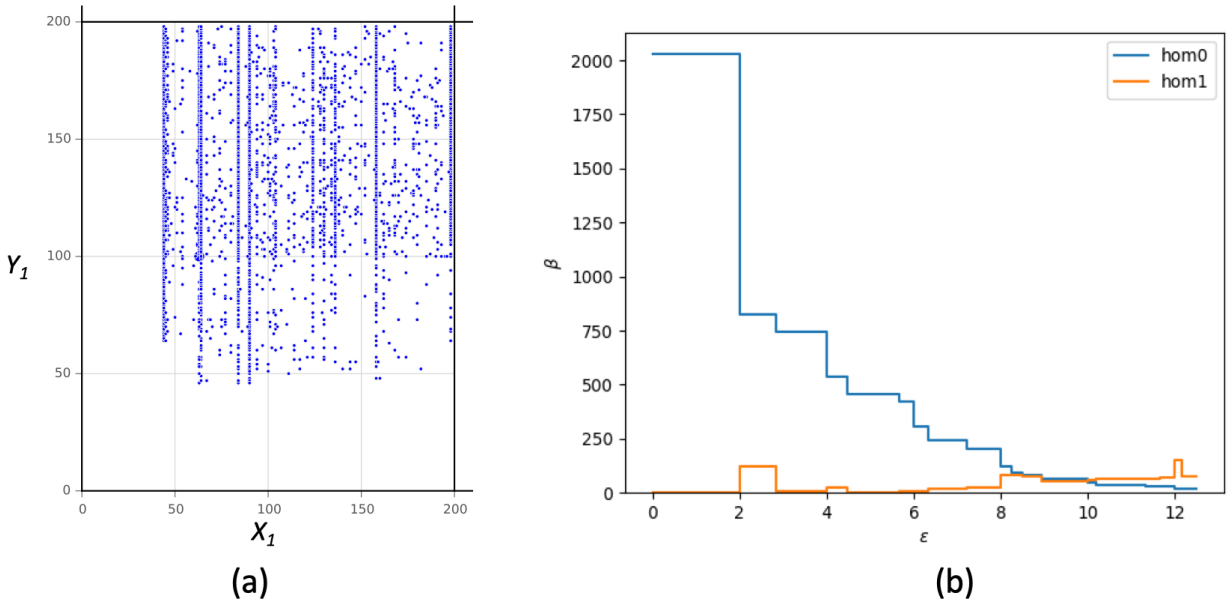


Figure 5.13: Coverage data of the 'line ends' pattern in the 'metal5' layer from an IC layout. In (a), we show this data as a scatter plot; in (b), we show the corresponding Betti plot of this data.

5.3 Applications To Coverage Data from IC Layouts

We now examine coverage data from IC layouts using PH. In Section 5.3.1, we examine the Betti plot of the 'line ends' pattern, which has two critical dimensions, as a simple case study. In Section 5.3.2, we study the effect that subsampling has on Betti plots. In Section 5.3.3, we compare coverage data between several patterns that each have two critical dimensions, so that we can also visualize the coverage data using scatter plots to help build our understanding. In Section 5.3.4, we apply our approach to a large variety of patterns, including patterns with 3–14 critical dimensions.

5.3.1 Betti Plot of the 'Line Ends' Pattern

In Figure 5.13, we consider the coverage data for the 'line ends' pattern in the 'metal5' layer from an IC layout. In Figure 5.13a, we show the scatter plot of the point cloud; in Figure 5.13b, we show the corresponding Betti plot. Comparing the Betti plot in Figure 5.13b

to the Betti plot in Figure 5.11b and the two Betti plots in Figure 5.12, we observe that the curves have similar subdomains. This suggests that the coverage data also conforms to a discretized grid with similar spacing, as points do not connect until $\epsilon = 2$. We also observe that the relative size of the drop in the first step is 0.59 for this coverage data, which suggests that it has a similar level of 'clustering' to that of synthetic Gaussian data in Figure 5.12a. This similarity is not apparent when comparing the corresponding scatter plots of the point clouds.

Recall from Section 5.1.1 that coverage data is a multiset in general, because the same configuration can appear multiple times in an IC layout. We have not yet made use of the multiplicity of points in our Betti plots. In the example of Figure 5.13, we effectively ignored multiplicities because coincident points (i.e., points with multiplicity greater than 1) form a single connected component at a resolution of $\epsilon = 0$.

5.3.2 Subsampling Sensitivity

In the testing phase of IC layouts, a common practice to acquire a representative sample of the configurations that occur in an entire chip is *clipping*. Clipping is a form of subsampling that involves extracting a contiguous sublayout from an IC layout, analogous to how one can extract a patch from an image [Ma09, WDC12, LS20]. This produces a representative sample under the assumption that configurations are not spatially correlated within an IC layout. Working under this same assumption, we use a form of subsampling coverage that we call 'clip' subsampling to produce samples that are statistically similar to clipping. In clip subsampling, one samples configurations from the coverage data without replacement according to a probability distribution that weights points in proportion to their multiplicities. Consequently, we test the sensitivity of Betti plots to clip subsampling using the coverage data of the 'line ends' pattern of the 'metal5' layer in an IC layout. In Figure 5.14, we show the Betti plots and scatter plots (inset) of coverage data that we clip subsample at rates of 100%, 80%, 50%, and 20%. We observe that the qualitative shape of the β_0 (blue) and β_1 (orange) curves in the Betti plots are well-preserved under this clip subsampling. This

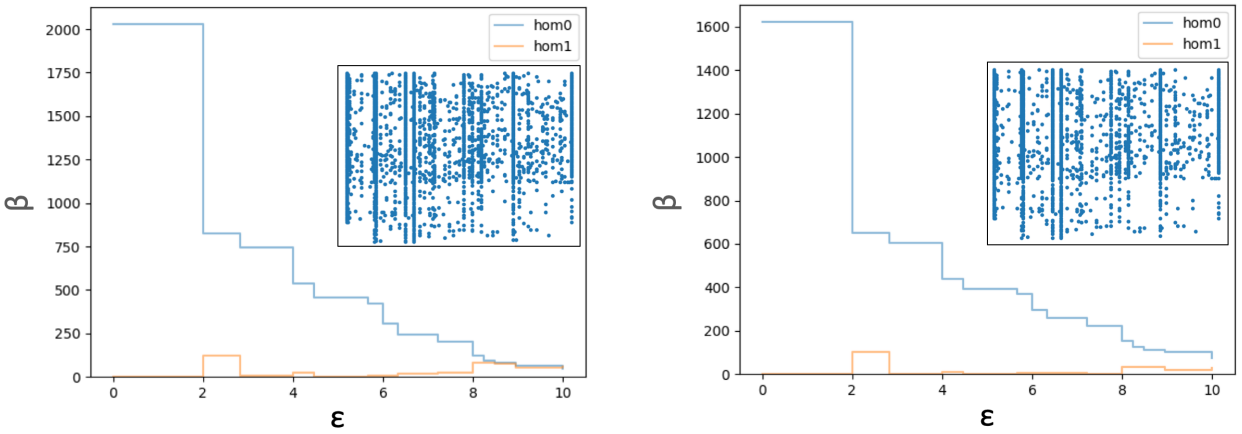
suggests that Betti plots may be robust to the type of sampling that occurs in clipping when testing chips.

We next use the distance function that we defined in Equation (5.1) to compute the distance between the Betti plot of the complete coverage data and the Betti plots of the clip-subsampled coverage data. The distances of the 80%, 50%, and 20% clip-subsampled data to the full data are approximately 0.001464, 0.00335, and 0.005143, respectively. The small yet monotonic increases in distance as we clip subsample progressively smaller fractions of the data are consistent with our observation that the clip subsampled Betti plots have lower resolution but retain the general shape of the Betti plot of the full point cloud.

5.3.3 Comparing Coverage Data of Different Patterns

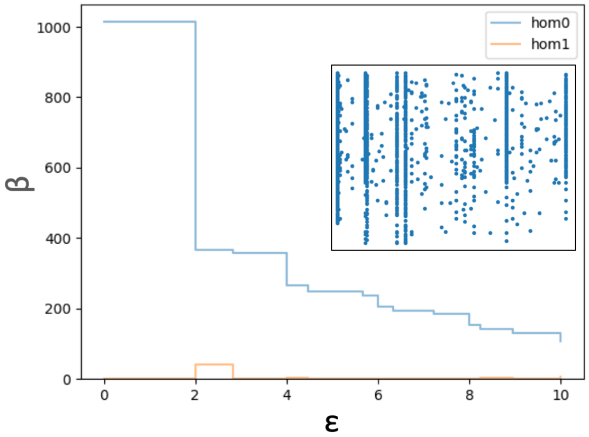
Understanding the relationship between the coverage data of different patterns can reveal insights into the IC design-manufacturing process [DTX17]. In Figure 5.15, we compare the coverage data of two patterns in the 'metal5' layer: the 'left line ends' pattern and the 'right line ends' pattern. These patterns are X-mirrors of each other, so one may expect their dimensional coverage data to be similar. The inset scatter plots show that they are similar, but this visual comparison of scatter plots is already cumbersome, even in this setting with only two critical dimensions. By contrast, one can readily observe in the Betti plots that the coverage data are similar. A Betti plot summarizes essential features of the coverage data at multiple scales without requiring additional inputs from domain knowledge of the data. Using Equation (5.1), we find that the distance between the Betti plots is approximately 0.000143.

In Figure 5.16, we compare the coverage data of two dissimilar patterns in the 'metal5' layer. The two patterns are the 'left line ends' pattern and the 'line ends' pattern, which are related by diagonal-transposition (or, equivalently, by a 90-degree rotation). These patterns have different characteristics because of asymmetries in the design-manufacturing process [Lev05, LS20]. Consequently, we expect their coverage to be dissimilar, which we observe in both the Betti plots and the inset scatter plots. Using Equation (5.1), we measure the

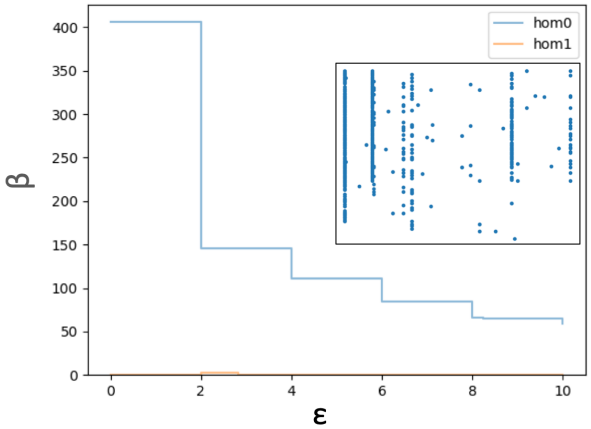


(a) 100%

(b) 80%



(c) 50%



(d) 20%

Figure 5.14: Betti plots and scatter plots (inset) of coverage data for the 'line ends' pattern in the 'metal5' layer of an IC layout at various clip subsampling rates: (a) 100%, (b) 80%, (c) 50%, and (d) 20%. The Betti plots of the clip-subsampled data have similar shape, but with progressively lower resolution as we include progressively less data.

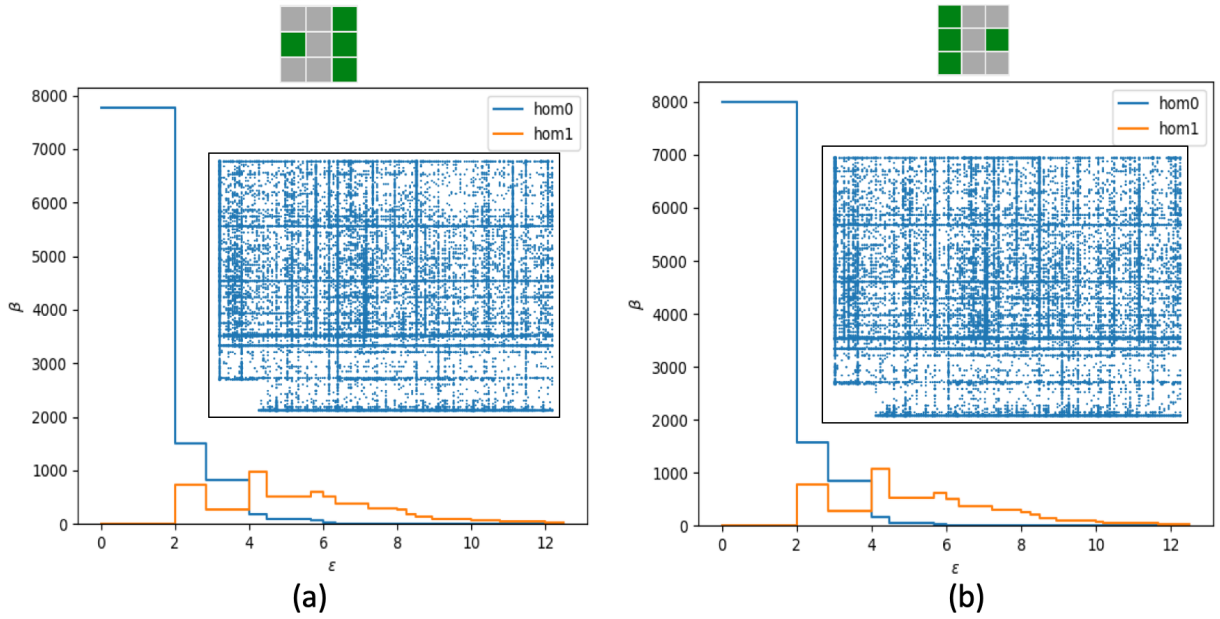


Figure 5.15: Betti plots with inset scatter plots of coverage data for (a) the 'left line ends' pattern and (b) the 'right line ends' pattern in the 'metal5' layer.

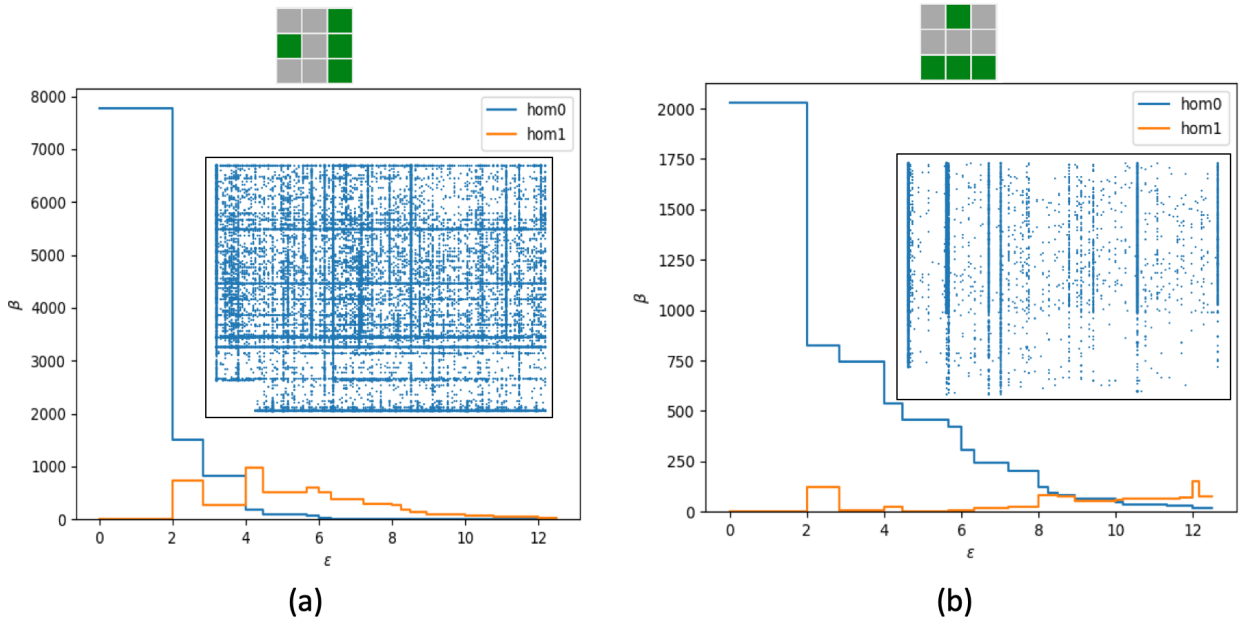


Figure 5.16: Betti plots with inset scatter plots of coverage data for (a) the 'left line ends' pattern and (b) the 'line ends' pattern in the 'metal5' layer.

distance between the Betti plots to be approximately .01, which is two orders-of-magnitude greater than the distance between the Betti plots in Figure 5.15.

The examples in this subsection suggest that the Betti-plot distance in Equation (5.1) is successfully capturing similarities and differences in coverage data that are relevant to IC design rules. Indeed, by doing a comprehensive comparison of each pattern and its X-mirror pattern in the 'metal5' layer, we consistently observe small distances between Betti plots. We thereby learn that the IC layout, and by inference the underlying design-manufacturing process, has some level of X-mirror symmetry in the 'metal5' layer [YK82, RSS85, CGC94]. Similarly, by comparing each pattern to its 90-degree rotation pattern in the 'metal5' layer, we consistently observe large distances between Betti plots. This suggests that the underlying design-manufacturing process is not symmetric with respect to 90-degree rotations [YK82, HSV90, KC95].

5.3.4 Systematic Comparisons

One of the benefits of PH is that it allows one to directly compare data sets that are embedded in spaces of different dimensions. In the context of IC layouts, this allows us to compare the coverage data of patterns that have a different number of critical dimensions. In this vein, we now perform a systematic comparison of a focal point cloud to an assortment of point clouds. For our example, we use the coverage data of the 'left line ends' pattern in the 'metal5' layer as the focal point cloud. In Figure 5.17, we show distance (see Equation (5.1)) between the Betti plot of the focal point cloud to the Betti plots of point clouds that come from clip subsampling, coverage data from related patterns in the same layer and in other layers, coverage data of randomly chosen patterns, and synthetically generated data.

We divide our comparisons into groups that are separated by red lines in Figure 5.17, with each group's label in green text. In the 'Clip Subsample' group, we compare the focal point cloud to its clip subsampling. As we saw in Section 5.3.2 when we studied clip subsampling for the 'line ends' pattern, we again see that the distance increases as we take progressively smaller samples. In the 'Child Patterns' group, we consider what we call 'child' patterns

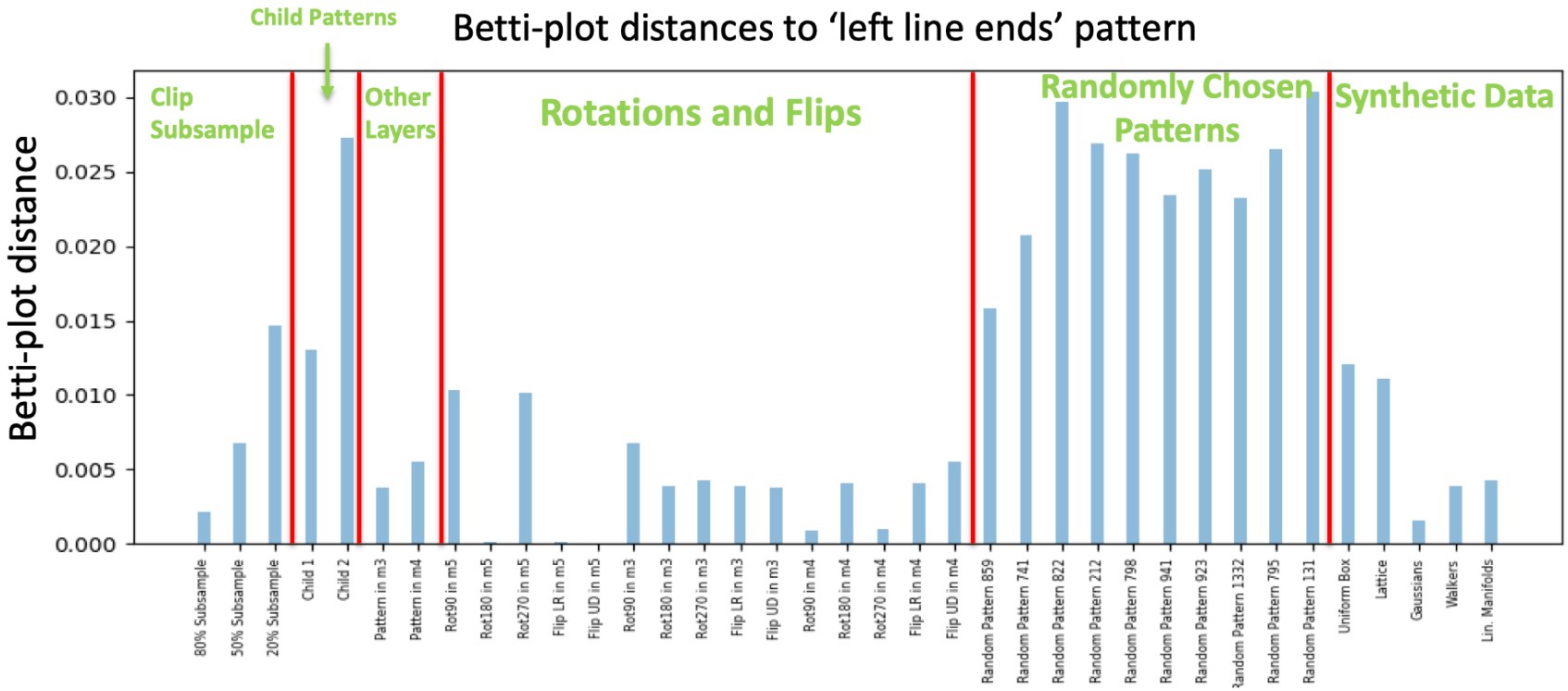


Figure 5.17: Betti-plot distances of the coverage data for the 'left line ends' pattern in the 'metal5' layer to several other point clouds.

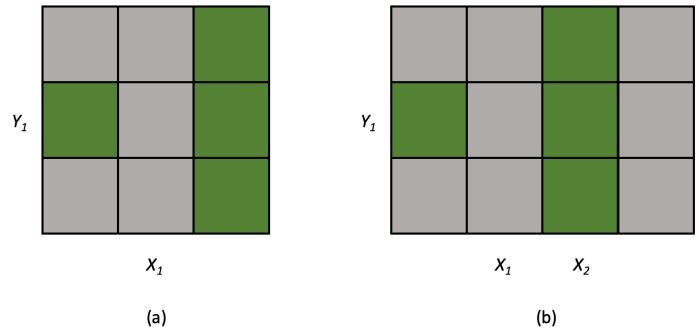


Figure 5.18: An example of parent and child patterns. In (a), we show a parent pattern with two critical dimensions, X_1 and Y_1. In (b), we show one of its child patterns, which has the additional critical dimension X_2. Child patterns always have at least one additional critical dimension than the associated parent pattern.

of the 'left line ends' pattern. By a child pattern, we mean a pattern that contains the original pattern (i.e., the 'parent' pattern) as a subpattern (see Figure 5.18). We observe that the Betti-plot distance between the parent pattern's coverage data and the coverage data of 'Child1' is approximately 40% smaller than the distance between the parent pattern and 'Child2'. Although we do not pursue this here, one can use this quantitative relationship between parent and child patterns in design tools. For example, Dai et al. introduced a graph-based approach to relate patterns that appear in designs [DTX17]. This approach involves creating an unweighted graph in which the nodes are patterns that appear in a layout and edges represent parent–child relationships. One can augment this graph by adding weights to the edges that are inversely proportional to the Betti-plot distances of the coverage data for each pattern. This may provide statistical insights or to improve search efficiency.

In the 'Other Layers' group, we perform inter-layer comparisons. We observe that the 'left line ends' pattern in the 'metal3' layer is more similar to the 'metal5' layer than is the same pattern in the 'metal4' layer, as expected based on the design-manufacturing process [YK82, RSS85, CGC94]. In the 'Rotations and Flips' group, we examine coverage data of patterns related to the 'left line ends' pattern by a rotation or flip. As we observed in Figure 5.15, in the 'metal5' layer, the 'left line ends' pattern is similar to its X-mirror-symmetry pattern

(labeled in the plot as 'Flip LR in m5'), the 'right line ends' pattern. The 'left line ends' pattern is equivalent to its Y-mirror-symmetry pattern (labeled in the plot as 'Flip UD in m5'), so the distance between the two is necessarily 0. The 'left line ends' pattern is dissimilar to its diagonal-transpose-symmetric pattern (labeled in the plot as 'Rot90 in m5'), as we observed in Figure 5.16. We also see that the coverage data from the 'metal4' layer of the diagonal-transpose pattern (labeled in the plot as 'Rot90 in m4') is similar to that of our focal point cloud. This is consistent with expectations based on the design-manufacturing process [YK82, HSV90, KC95]. In the 'Randomly Chosen Patterns' group, we give some baseline comparisons to the coverage data from the 'metal5' layer of patterns that we choose uniformly at random among all patterns in the IC layout.

In the 'Synthetic Data' group, we compare the focal point cloud to synthetic point clouds. In order from left to right, we use the continuous uniformly random point cloud from Figure 5.11a (labeled in the plot as 'Uniform Box'), the discretized uniformly random point cloud from Figure 5.11b (labeled in the plot as 'Lattice'), the Gaussian random point cloud from Figure 5.12a (labeled in the plot as 'Gaussian'), the random-walk-path point cloud from Figure 5.12b (labeled in the plot as 'Walkers'), and a point cloud that we generate by sampling points uniformly at random from three lines and then discretizing (labeled in the plot as 'Lin. Manifolds'). As we noted in Section 5.2.3, synthetic Gaussian data can closely resemble coverage data from IC layouts. This observation may give insight into how to generate realistic synthetic test data [DTX17].

Finally, we perform a systematic comparison of the coverage data of the 'left line ends' pattern in the 'metal5' layer to the coverage data of all other patterns in that layer. We compare coverage data of approximately 2000 patterns, some of which have 14 critical dimensions. In Figure 5.19, we extract the twenty most-similar patterns according to the Betti-plot distance of Equation (5.1). We call attention to a few of these patterns. We note that the nearest match is the 'right line ends' pattern that that we studied in Figure 5.15. Additionally, a child pattern (see Figure 5.18) appears as the 6th-most similar pattern.[7] We

[7]The patterns that ranked 2–5th in terms of similarity are also of interest. However, the reasons why they are interesting are not within the scope of this thesis.

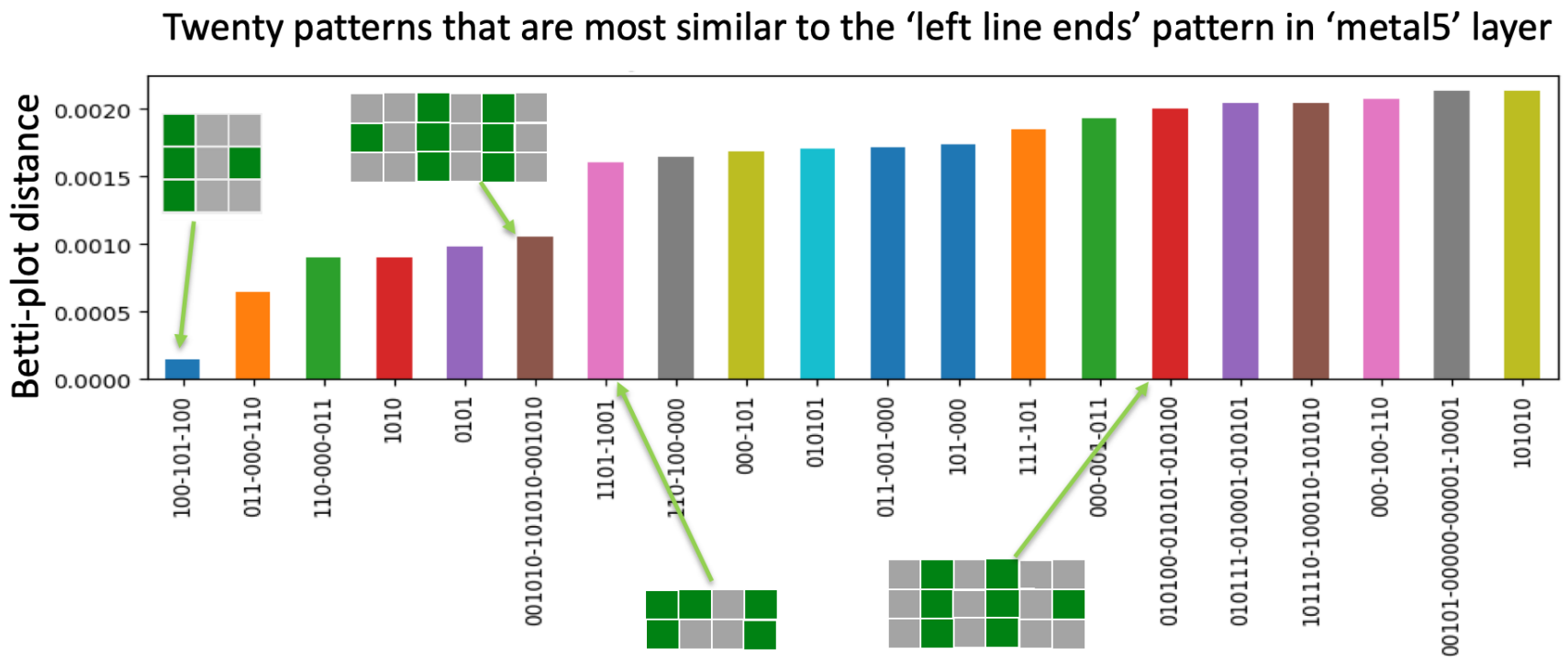


Figure 5.19: The twenty most-similar patterns based on Betti-plot distances to the 'left line ends' pattern in the 'metal5' layer. The bar chart shows the Betti-plot distances.

obtained these quantitative relationships between patterns by comparing point clouds that are embedded in spaces of different dimensions (i.e., patterns with different numbers of critical dimensions) without providing additional input or domain knowledge. Specifically, we do not identify which two of the five critical dimensions for the child pattern correspond to the critical dimensions of the parent pattern.

5.4 Conclusions and Future Work

Analyzing IC layouts for manufacturability is very challenging. However, tools from topological data analysis — specifically, persistent homology — provide helpful ways to examine high-dimensional coverage data at multiple scales. In particular, Betti plots allow one to compute useful summaries of coverage data sets. We demonstrated that Betti plots are robust to a certain type of subsampling that is similar to clipping in chip testing. We then discussed how Betti plots enable quantitative comparisons between coverage data sets, and we used these comparisons to determine pattern symmetries and inter-layer relations that arise in the photolithography process. We were also able to find similar and dissimilar patterns using only coverage data, without the need to incorporate domain knowledge.

We propose two possible directions for future work. First, it would be interesting to use recent advances in persistent homology that allow one to extract the locations of homological features [Oba18] to detect boundaries in the coverage data. For example, we anticipate that one can use boundary information in probabilistic classification of yield-limiting configurations. Second, it seems worthwhile to perform a broader exploration of coverage data from thousands of IC layouts using persistent homology. One could then incorporate commonalities into the design rules of computer-aided design systems.

CHAPTER 6

Applications of Network Dictionary Learning

In this chapter, I present an ongoing project that entails the analysis of networks. This project is a collaboration with Hanbaek Lyu (UCLA), Mason A. Porter (UCLA), and Josh Vendrow (UCLA) on applications of a recently developed method called 'network dictionary learning' (NDL). We discuss problems of network reconstruction and denoising using NDL on both synthetic and real-world networks.

6.1 Background

Lyu et al. [LMS19, LNB19] recently introduced an algorithm for the analysis of networks that combines Markov chain Monte Carlo (MCMC) motif sampling and non-negative dictionary learning. We begin by discussing their motif-sampling method to generate data from networks and then discuss how to use such data to learn dictionaries of 'parts' of networks.

MCMC motif sampling of a simple graph $G = (V_1, E_1)$ for some motif $K = (V_2, E_2)$ involves exploring the space of subgraphs of G that are induced by homomorphic copies of K in a way that converges to some target distribution, which we take to be the uniform distribution. One does this by finding a random sequence $(x_t : K \to G)_{t\in\mathbb{N}_0}$ of graph homomorphisms[1] whose stationary distribution is the uniform distribution.

Under certain constraints on G and K, one way to construct such a sequence is via Glauber-chain sampling [LMS19]. We describe Glauber-chain sampling in the case that K

[1] A graph homomorphism $f : H \to H'$, where $H = (V, E)$ and $H' = (V', E')$, is a map from the nodes of H to the nodes of H' that preserves adjacency $(i, j) \in E \implies (f(i), f(j)) \in E'$.

is a path graph[2] of length κ (with nodes labeled from 0 to $\kappa - 1$), and we start with some initial graph homomorphism $x_0 : K \to G$. To obtain x_{t+1} from x_t, we perform the following steps:

1. Select a node i of K uniformly at random.

2. For each node $j \neq i$, map it to the same node from the previous step: $x_{t+1}(j) = x_t(j)$.

3. Map node i randomly to one of the nodes of G such that it makes x_{t+1} a graph homomorphism. Specifically, when $i \in \{1, \ldots, \kappa - 2\}$, set $x_{t+1}(i) \in V_1$ equal to a node that we choose uniformly at random from the intersection of the neighborhoods of $x_{t+1}(i-1)$ and $x_{t+1}(i+1)$. When $i = 0$, we use the neighborhood of $x_{t+1}(i+1)$; when $i = \kappa - 1$, we use the neighborhood of $x_{t+1}(i-1)$.

Using the above Glauber-chain sampling, we record a sequence $(A_t)_{t \in \mathbb{N}_0}$ of adjacency matrices of dimension $\kappa \times \kappa$ that correspond to the subgraphs of G that are induced by $(x_t(K))_{t \in \mathbb{N}_0}$. The sequence $(A_t)_{t \in \mathbb{N}_0}$ gives the data that we want to analyze using non-negative matrix factorization (NMF). Because these data take the form of an infinite sequence, we desire an NMF algorithm that is *online*, in the sense that one can stream the data incrementally to produce progressively more accurate approximations.

Each adjacency matrix in the sequence $(A_t)_{t \in \mathbb{N}_0}$ is an observation from the network G. Our goal is to learn the 'parts' that combine to form these observations (and consequently the network G), in the sense that we want to find a set $\{B_1, \ldots, B_r\}$ of matrices that can approximate the observations using only *conical combinations* (i.e., sums of the form $\sum_{i=1}^{r} a_i B_i$ where each $a_i \geq 0$). Consider truncating the sequence at some $t = T - 1$. Flattening the T adjacency matrices into κ^2-dimensional column vectors and concatenating these vectors, we obtain a data matrix D of dimension $\kappa^2 \times T$. We can express parts-based

[2]A path graph is a connected graph, with at least two nodes, in which exactly two nodes have degree 1 and the remaining nodes have degree 2. One can draw a path graph so that all of its nodes and edges lie on a line.

learning through NMF, which is the constrained minimization problem

$$\underset{W \in \mathbb{R}_{\geq 0}^{\kappa^2 \times r},\, H \in \mathbb{R}_{\geq 0}^{r \times T}}{\arg\min} \|D - WH\|_F^2 \,. \tag{6.1}$$

One typically chooses r so that $(\kappa^2 + T)r < \kappa^2 T$ and can thus regard the product WH as a low-rank approximation of D [LS99]. The matrix W is often called the 'dictionary' matrix [MBP10, TD12]. This terminology arises from the interpretation that the r columns of W are the parts (also called the 'atoms') that combine conically to approximate the data. In settings where the data become available in a sequential order, under certain independence assumptions, one can use an online non-negative matrix factorization (ONMF) algorithm that solves Equation (6.1) iteratively [MBP10].

Lyu et al. introduced a new ONMF method that requires that data form a Markov chain [LNB19], rather than the stronger assumption that the data are independent. That is, their method works for sequential data in which each data point is independent, conditioned on the previous data point. Combining their method with MCMC motif sampling, they introduced network dictionary learning (NDL). See Algorithm 1 in Ref. [LNB19] for details.

6.2 NDL on Synthetic Networks

One particularly interesting application is *reconstruction* [ZLL18], in which we attempt to rebuild a network using only (non-negative) linear combinations of the atoms from some dictionary W. We refer to this process as *self-reconstruction* if we learn W from the network that we are attempting to reconstruct and *cross-reconstruction* otherwise. When testing the accuracy of reconstruction, we use the Jaccard distance

$$d_J(E_1, E_2) = 1 - \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|} \tag{6.2}$$

between the original network's edge set E_1 and the reconstructed network's edge set E_2. We obtain the same qualitative results when we compute the Rand index [Ran71].

To gain a better understanding of reconstruction using NDL, we apply it to a variety of synthetic networks. We are currently considering the ER $G(N, p)$ random graph model,

the Watts–Strogatz (WS) model [WS98], and the Barabási–Albert (BA) model [BA99]. To construct a WS network, we begin with a regular ring lattice with N nodes, where each node is connected to its k nearest neighbors (with $k/2$ on each side). For each node i and for each of the $k/2$ edges to its right, with some probability, we rewire the edge from (i, u) to (i, v), where we choose v uniformly at random from nodes $j \neq i$. To construct a BA network, we begin with τ isolated nodes. In each step, we add a new node with τ edges that preferentially connect to the existing nodes with a probability that is proportional to the existing nodes' degrees.

In Figure 6.1(a)–(c), we show the $r = 25$ atoms (i.e., the columns of W that we reshape into square matrices) that we learn from three synthetic networks: (a) an ER $G(N, p)$ graph with $N = 1000$ and $p = 0.05$; (b) a WS network with $N = 1000$ nodes, 50 nearest-neighbor connections, and rewiring probability $p = 0.05$; and (c) a BA network with $N = 1000$ nodes and parameter value $\tau = 10$. We use Glauber-chain sampling in which K is a path graph of length $\kappa = 21$.

In Figure 6.1(d)–(f), we show our attempts at reconstructing synthetic networks. We indicate the reconstruction accuracy [using Equation (6.2)] on the vertical axis. Each curve corresponds to reconstruction using a dictionary that we learn from some network; see the legend in Figure 6.1(f). We vary the dictionary size r, which we indicate on the horizontal axis. For each combination of (1) synthetic network to reconstruct, (2) synthetic network from which we learn a dictionary, and (3) size r of the dictionary that we learn, we conduct 5 trials, each of which uses a different realization of the network models. Each marker in Figure 6.1(d)–(f) indicates the mean accuracy over these 5 trials, and the error bars indicate the standard deviations.

We observe that self-reconstruction accuracy is above 75% and that cross-reconstruction using the same network model but with (slightly) different parameters also has an accuracy above 75% in each experiment. We also observe that dictionaries that we learn from ER networks with $p = 0.05$ perform poorly at cross-reconstructing the WS networks. Additionally, we observe that dictionaries that we learn from BA networks with $\tau = 10$ tend to perform well at cross-reconstructing the ER and WS networks. These experiments also illustrate the

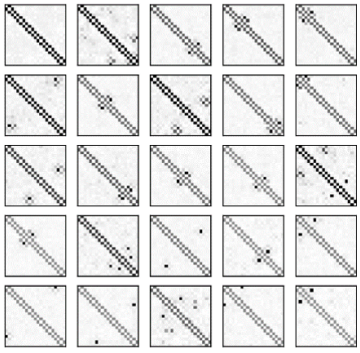
asymmetry in cross-reconstruction accuracy; one network's dictionary may cross-reconstruct another network well, but not vice versa.

6.3 NDL on Real-World Networks

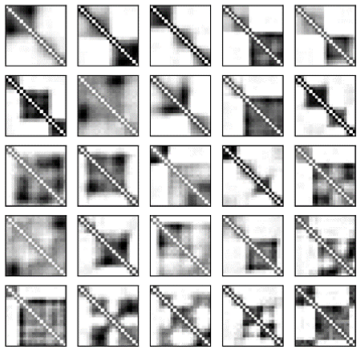
We now apply NDL to real-world networks. We are currently studying the 'Facebook100' data set from Ref. [TMP12]. This data set consists of Facebook social networks at one hundred American colleges and universities at a single point in time in Fall 2005. In Figure 6.2, we show the self-reconstruction and cross-reconstruction accuracies of networks from four universities: Caltech, Harvard, MIT, and UCLA. We use $T = 10^8$ observations to train each dictionary. An unexpected result is that the cross-reconstruction of MIT's network by Harvard's or UCLA's network outperforms MIT's self-reconstruction. One possible explanation for this is that the mixing time for motif sampling of the MIT network using a Glauber chain may be significantly longer than 10^8 steps. Although there are bounds for the mixing time of motif sampling of dense networks [LMS19], we are unaware of good bounds for motif-sampling mixing time for sparse networks.

6.4 Current plans

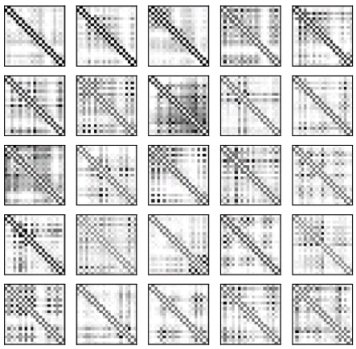
We also are studying brain networks that encode connectivity between different locations of human brains. Specifically, we are examining a data set that is based on functional magnetic resonance imaging (fMRI) studies on patients who were diagnosed with schizophrenia and controls who were deemed to be healthy from the Center for Biomedical Research Excellence [ABB17]. By comparing the dictionary matrices that we learn from the brain networks, we consider the following classification problem: Can we identify brain networks of schizophrenic patients versus healthy controls based on atoms? Another problem that we are studying is a network analog of 'inpainting' [MBP10]: we corrupt a network (by removing and/or adding edges), learn a dictionary from the corrupted network, and then self-reconstruct to remove the corruptions.



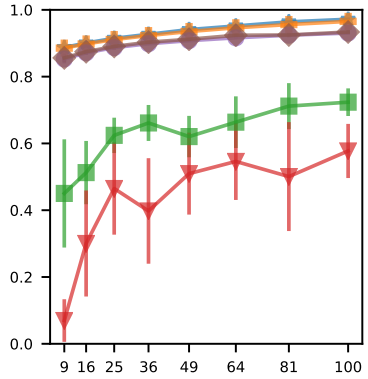
(a) Atoms from ER Network



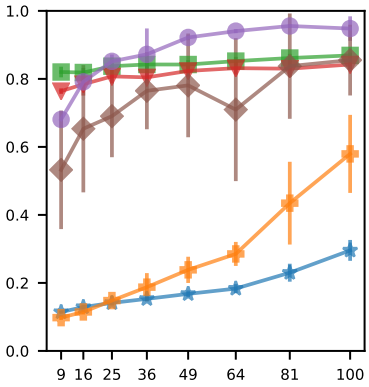
(b) Atoms from WS Network



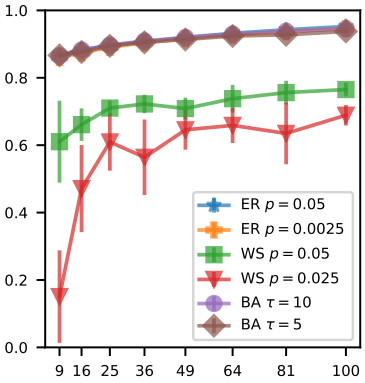
(c) Atoms from BA Network



(d) Reconstructing ER Networks



(e) Reconstructing WS Networks



(f) Reconstructing BA Networks

Figure 6.1: Atoms and reconstruction accuracy using synthetic networks. In panels (a)–(c), we show the $r = 25$ atoms that we learn from three synthetic networks. In panels (d)–(f), we show self- and cross-reconstruction accuracies of the same three network models. Each curve corresponds to reconstruction using a dictionary that we learn from some network; see the legend in panel (f). The horizontal axis indicates the number r of atoms in the dictionary that we use. The vertical axis indicates the accuracy based on the Jaccard distance between the original and reconstructed networks' edge sets. For each combination of (1) network to reconstruct, (2) network from which we learn a dictionary, and (3) size of dictionary, we conduct 5 trials, each of which uses a different realization of the network models. Each marker indicates the mean accuracy over these 5 trials, and the error bars indicate the standard deviations.

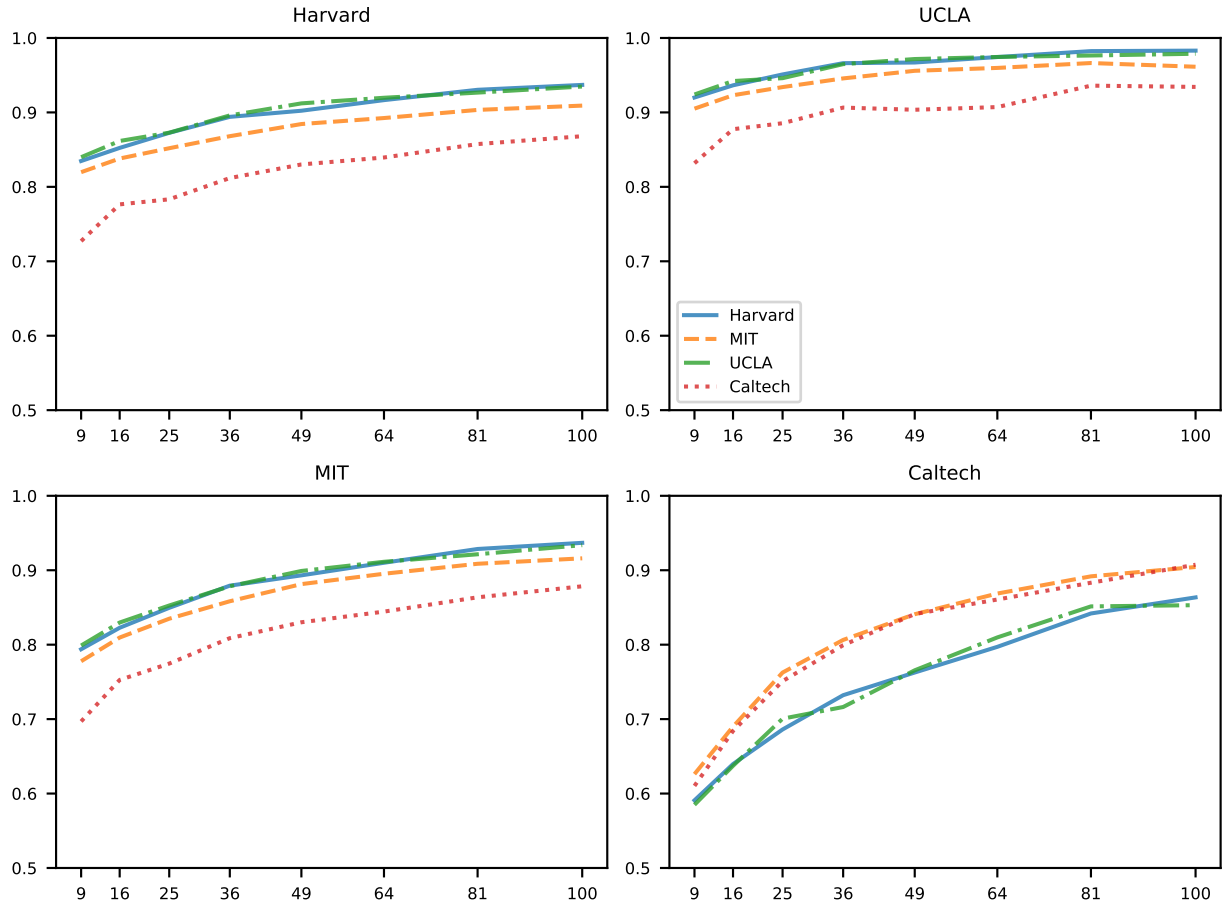


Figure 6.2: Self- and cross-reconstruction accuracy of Facebook networks using NDL. Each plot's title indicates the university whose network we reconstruct. Each curve corresponds to reconstruction using a dictionary that we learn from some university's network; see the top-right plot for the legend. The horizontal axis indicates the number r of atoms that we use, and the vertical axis indicates the accuracy based on the Jaccard distance between the original and reconstructed networks' edge sets.

CHAPTER 7

Conclusions

In this thesis, we explored tools to model and analyze complex systems. In the first half, we discussed two classes of opinion models on networks, voter models and bounded confidence models. In the second half, we explored applications of topological data analysis and network dictionary learning.

We introduced a nonlinear version of coevolving voter models (CVM) in which the probability that a node rewires or adopts is a function of how well it "fits in" with the nodes in its neighborhood. We compared the effects of various rewiring mechanisms and various network structures in linear CVMs and our nonlinear version. We observed that our nonlinear CVM exhibits qualitatively similar characteristics as the linear CVM of Ref. [DGL12] with respect to terminal state densities when both models are initialized on ER networks with equal state densities. For example, both types of models possess a regime with rapid fragmentation into communities of different opinion states and a regime in which the system reaches a consensus. However, when we seed the opinion models with more complicated network architectures, such as ones with community structure or core–periphery structure, we observed striking differences between our nonlinear CVM and the aforementioned linear CVM. In these scenarios, when the nodes have distorted views of local densities — such that they believe that they are in the majority or minority when the opposite is true — the value of our CVM's nonlinearity parameter has a major effect on terminal state densities. For certain values of the parameter and certain initial network topologies, the initially minority state consistently became the consensus in our simulations. For other values of it, the initially majority state consistently became the consensus.

We also discussed ongoing work that extends the Deffuant–Weisbach model for opinion

dynamics to hypergraphs. We presented some preliminary analysis and numerical results on the convergence dynamics of this model for complete hypergraphs.

We then explored applications of topological data analysis to integrated circuits (IC). Current methods to analyze IC layouts for manufacturability are limited to low-dimensional tolerance checks. However, using tools from topological data analysis — specifically, persistent homology — we present ways to examine high-dimensional coverage data at multiple scales. By computing Betti plots, we obtained helpful summaries of coverage data sets. We also demonstrated that Betti plots are robust to a certain type of subsampling that is similar to clipping in chip testing. We then discussed how Betti plots enable fast and quantitative comparisons between coverage data sets, and we used these comparisons to determine pattern symmetries and inter-layer relations that arise in the photolithography process. We were also able to find similar and dissimilar patterns using only coverage data.

Finally, we discussed ongoing work that investigates applications of a recently developed data analysis algorithm called network dictionary learning (NDL). We presented some preliminary results on network reconstruction using NDL on both synthetic and real networks.

Networked systems are ubiquitous, from human interaction to physical devices. We studied a few of the many ways in which one can use networks to model dynamics and analyze data. Looking forward, there are many opportunities to unite these modeling and statistical approaches. For example, there are interesting questions regarding model inference in the space of opinion dynamics. Conversely, combining a model-based approach with methods such as TDA may yield insights into many pertinent questions about the IC design-manufacturing process.

REFERENCES

[ABB17] C. J. Aine, H. J. Bockholt, J. R. Bustillo, J. M. Cañive, A. Caprihan, C. Gasparovic, F. M. Hanlon, J. M. Houck, R. E. Jung, J. Lauriello, J. Liu, A. R. Mayer, N. I. Perrone-Bizzozero, S. Posse, J. M. Stephen, J. A. Turner, V. P. Clark, and Vince D. Calhoun. "Multimodal neuroimaging in schizophrenia: description and dissemination." *Neuroinformatics*, **15**(4):343–364, 2017.

[ALM92] Enrique D. Andjel, Thomas M. Liggett, and Thomas Mountford. "Clustering in one-dimensional threshold voter models." *Stochastic Processes and their Applications*, **42**(1):73–90, 1992.

[Axe97] Robert Axelrod. "The dissemination of culture: A model with local convergence and global polarization." *Journal of Conflict Resolution*, **41**(2):203–226, 1997.

[BA99] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks." *Science*, **286**(5439):509–512, 1999.

[Ban71] Albert Bandura. *Social Learning Theory*. General Learning Press, New York, New York, USA, 1971.

[BCP11] Andrea Baronchelli, Claudio Castellano, and Romualdo Pastor-Satorras. "Voter models on weighted networks." *Physical Review E*, **83**(6):066117, 2011.

[BDZ15] Anirban Basak, Rick Durrett, and Yuan Zhang. "The evolving voter model on thick graphs." *arXiv:1512.07871*, 2015.

[BG11] Gesa A. Böhme and Thilo Gross. "Analytical calculation of fragmentation transitions in adaptive networks." *Physical Review E*, **83**(3):035101, 2011.

[BG12] Gesa A. Böhme and Thilo Gross. "Fragmentation transitions in multistate voter models." *Physical Review E*, **85**(6):066117, 2012.

[BKR03] Eli Ben-Naim, Paul L. Krapivsky, and Sidney Redner. "Bifurcations and patterns in compromise processes." *Physica D: Nonlinear Phenomena*, **183**(3):190–204, 2003.

[Bod16] Leticia Bode. "Pruning the news feed: Unfriending and unfollowing political content on social media." *Research & Politics*, **3**:2053168016661873, 2016.

[Boh13] Gesa Angelika Böhme. "Emergence and persistence of diversity in complex networks." *The European Physical Journal Special Topics*, **222**(12):3089–3169, 2013.

[BR00] Stefan Bornholdt and Thimo Rohlf. "Topological evolution of dynamical networks: Global criticality from local dynamics." *Physical Review Letters*, **84**(26):6114–6117, 2000.

[Bre19] M. Brede. "How does active participation effect consensus: Adaptive network model of opinion dynamics and influence maximizing rewiring." *Complexity*, **2019**:1486909, 2019.

[BS17] Riddhipratim Basu and Allan Sly. "Evolving voter model on dense random graphs." *The Annals of Applied Probability*, **27**(2):1235–1288, 2017.

[CD91] J. T. Cox and R. Durrett. *Random Walks, Brownian Motion, and Interacting Particle Systems: A Festschrift in Honor of Frank Spitzer*, chapter Nonlinear Voter Models, pp. 189–201. Birkhäuser Boston, Boston, Massachusetts, USA, 1991.

[CDF18] Colin Cooper, Martin Dyer, Alan Frieze, and Nicolás Rivera. "Discordant voting processes on finite graphs." *SIAM Journal on Discrete Mathematics*, **32**(4):2398–2420, 2018.

[CEH07] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. "Stability of persistence diagrams." *Discrete & Computational Geometry*, **37**(1):103–120, 2007.

[Cer03] Paul E. Ceruzzi. *A History of Modern Computing*. MIT press, Cambridge, Massachusetts, USA, second edition, 2003.

[CFL09] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. "Statistical physics of social dynamics." *Reviews of Modern Physics*, **81**(2):591–646, 2009.

[CFP17] Jason P. Cain, Moutaz Fakhry, Piyush Pathak, Jason Sweis, Frank E. Gennari, and Ya-Chieh Lai. "Pattern-based analytics to estimate and track yield risk of designs down to 7nm." In *Design-Process-Technology Co-optimization for Manufacturability XI*, volume 10148. International Society for Optics and Photonics, 2017.

[CGC94] John M. Cohn, David J. Garrod, Rick Carley, and Rob A. Rutenbar. *Analog Device-Level Layout Automation*. Springer Science & Business Media, New York, New York, USA, 1994.

[Cho19] Philip S. Chodrow. "Configuration models of random hypergraphs and their applications." *arXiv:1902.09302*, 2019.

[CHY15] Moo K. Chung, Jamie L. Hanson, Jieping Ye, Richard J. Davidson, and Seth D. Pollak. "Persistent homology in sparse regression and its application to brain morphometry." *IEEE Transactions on Medical Imaging*, **34**(9):1928–1939, 2015.

[CID11] Iain D. Couzin, Christos C. Ioannou, Güven Demirel, Thilo Gross, Colin J. Torney, Andrew Hartnett, Larissa Conradt, Simon A. Levin, and Naomi E. Leonard. "Uninformed individuals promote democratic consensus in animal groups." *Science*, **334**(6062):1578–1580, 2011.

[CLG16] Jason P. Cain, Ya-Chieh Lai, Frank Gennari, and Jason Sweis. "Methodology for analyzing and quantifying design style changes and complexity using topological patterns." In *Design-Process-Technology Co-optimization for Manufacturability X*, volume 9781, p. 978108. International Society for Optics and Photonics, 2016.

[CLW13] Peter Csermely, András London, Ling-Yun Wu, and Brian Uzzi. "Structure and dynamics of core/periphery networks." *Journal of Complex Networks*, **1**(2):93–123, 2013.

[CM20] Philip S. Chodrow and Peter J. Mucha. "Local symmetry and global structure in adaptive voter models." *SIAM Journal on Applied Mathematics*, **80**(1):620–638, 2020.

[CMP09] Claudio Castellano, Miguel A Muñoz, and Romualdo Pastor-Satorras. "Nonlinear q-voter model." *Physical Review E*, **80**(4):041129, 2009.

[Coc34] W. G. Cochran. "The distribution of quadratic forms in a normal system, with applications to the analysis of covariance." *Mathematical Proceedings of the Cambridge Philosophical Society*, **30**(2):178–191, 1934.

[Cox89] J. Theodore Cox. "Coalescing random walks and voter model consensus times on the torus in $\mathbb{Z}^d$." *The Annals of Probability*, **17**(4):1333–1366, 1989.

[CS73] Peter Clifford and Aidan Sudbury. "A model for spatial conflict." *Biometrika*, **60**(3):581–588, 1973.

[CT12] Fan Chung and Alexander Tsiatas. *International Workshop on Algorithms and Models for the Web-Graph*, chapter Hypergraph coloring games and voter models, pp. 1–16. Springer, Heidelberg, Germany, 2012.

[CTS16] Adrián Carro, Raúl Toral, and Maxi San Miguel. "The noisy voter model on complex networks." *Scientific Reports*, **6**:24775, 2016.

[DBZ13] Su Do Yi, Seung Ki Baek, Chen-Ping Zhu, and Beom Jun Kim. "Phase transition in a coevolving network of conformist and contrarian voters." *Physical Review E*, **87**(1):012806, 2013.

[DGL12] Richard Durrett, James P. Gleeson, Alun L. Lloyd, Peter J. Mucha, Feng Shi, David Sivakoff, Joshua E. S. Socolar, and Chris Varghese. "Graph fission in an evolving voter model." *Proceedings of the National Academy of Sciences of the United States of America*, **109**(10):3682–3687, 2012.

[DNA00] Guillaume Deffuant, David Neau, Frederic Amblard, and Gérard Weisbuch. "Mixing beliefs among interacting agents." *Advances in Complex Systems*, **3**(01n04):87–98, 2000.

[DTX17] Vito Dai, Edward Kah Ching Teoh, Ji Xu, and Bharath Rangarajan. "Optimization of complex high-dimensional layout configurations for IC physical designs

using graph search, data analytics, and machine learning." In *Design-Process-Technology Co-optimization for Manufacturability XI*, volume 10148, p. 1014808. International Society for Optics and Photonics, 2017.

[DVB14] Guven Demirel, Federico Vazquez, G. A. Böhme, and Thilo Gross. "Moment-closure approximations for discrete adaptive networks." *Physica D: Nonlinear Phenomena*, **267**:68–80, 2014.

[DVB16] Michela Del Vicario, Gianna Vivaldo, Alessandro Bessi, Fabiana Zollo, Antonio Scala, Guido Caldarelli, and Walter Quattrociocchi. "Echo chambers: Emotional contagion and group polarization on Facebook." *Scientific Reports*, **6**:37825, 2016.

[EB02] Holger Ebel and Stefan Bornholdt. "Coevolutionary games on networks." *Physical Review E*, **66**(5):056118, 2002.

[EH08] Herbert Edelsbrunner and John Harer. "Persistent homology— survey." *Contemporary Mathematics*, **453**:257–282, 2008.

[ELZ02] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. "Topological persistence and simplification." *Discrete & Computational Geometry*, **28**:511–533, 2002.

[Fel68] William Feller. *An Introduction to Probability Theory and Its Applications, Vol. 1.* John Wiley & Sons, Inc., New York, New York, USA, third edition, 1968.

[Fel91] Scott L. Feld. "Why your friends have more friends than you do." *American Journal of Sociology*, **96**(6):1464–1477, 1991.

[Fer90] Irene Ferreira. "The probability of survival for the biased voter model in a random environment." *Stochastic Processes and their Applications*, **34**(1):25–38, 1990.

[FES11] Juan Fernández-Gracia, Víctor M. Eguíluz, and Maxi San Miguel. "Update rules and interevent time distributions: Slow ordering versus no ordering in the voter model." *Physical Review E*, **84**(1):015103, 2011.

[FH16] Santo Fortunato and Darko Hric. "Community detection in networks: A user guide." *Physics Reports*, **659**:1–44, 2016.

[FMG16] Peter G. Fennell, Sergey Melnik, and James P. Gleeson. "Limitations of discrete-time approaches to continuous-time contagion dynamics." *Physical Review E*, **94**(5):052125, 2016.

[FSR14] Juan Fernández-Gracia, Krzysztof Suchecki, José J. Ramasco, Maxi San Miguel, and Víctor M. Eguíluz. "Is the voter model a model for voters?" *Physical Review Letters*, **112**(15):158701, 2014.

[FWG17] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez-Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. "Coevolve: A joint point process model for information diffusion and network evolution." *The Journal of Machine Learning Research*, **18**(1):1305–1353, 2017.

[Gas15] Michael T. Gastner. "The Ising chain constrained to an even or odd number of positive spins." *Journal of Statistical Mechanics: Theory and Experiment*, **2015**:P03004, 2015.

[GB07] Thilo Gross and Bernd Blasius. "Adaptive coevolutionary networks: a review." *Journal of the Royal Society Interface*, **5**:259–271, 2007.

[GDB06] Thilo Gross, Carlos J. Dommar D'Lima, and Bernd Blasius. "Epidemic dynamics on an adaptive network." *Physical Review Letters*, **96**(20):208701, 2006.

[Ghr08] Robert Ghrist. "Barcodes: the persistent topology of data." *Bulletin of the American Mathematical Society*, **45**(1):61–75, 2008.

[Ghr14] Robert Ghrist. *Elementary Applied Topology*. Createspace, first edition, 2014.

[GJ10] Benjamin Golub and Matthew O. Jackson. "Naïve learning in social networks and the wisdom of crowds." *American Economic Journal: Microeconomics*, **2**(1):112–49, 2010.

[Gla63] Roy J. Glauber. "Time-dependent statistics of the Ising model." *Journal of Mathematical Physics*, **4**(2):294–307, 1963.

[Gle11] James P. Gleeson. "High-accuracy approximation of binary-state dynamics on networks." *Physical Review Letters*, **107**(6):068701, 2011.

[Gle13] James P. Gleeson. "Binary-state dynamics on complex networks: Pair approximation and beyond." *Physical Review X*, **3**:021004, 2013.

[GLH19] Daniel Geschke, Jan Lorenz, and Peter Holtz. "The triple-filter bubble: Using agent-based modelling to test a meta-theoretical framework for the emergence of filter bubbles and echo chambers." *British Journal of Social Psychology*, **58**:129–149, 2019.

[GM95] Boris L. Granovsky and Neal Madras. "The noisy voter model." *Stochastic Processes and their Applications*, **55**:23–43, 1995.

[GMW12] James P. Gleeson, Sergey Melnik, Jonathan A. Ward, Mason A. Porter, and Peter J. Mucha. "Accuracy of mean-field theory for dynamics on real-world networks." *Physical Review E*, **85**(2):026106, 2012.

[GOG18] Michael T. Gastner, Beáta Oborny, and Máté Gulyás. "Consensus time in a voter model with concealed and publicly expressed opinions." *Journal of Statistical Mechanics: Theory and Experiment*, **2018**(6):063401, 2018.

[GTG19] Michael T. Gastner, Károly Takács, Máté Gulyás, Zsuzsanna Szvetelszky, and Beáta Oborny. "The impact of hypocrisy on opinion formation: A dynamic model." *PloS ONE*, **14**(6), 2019.

[GZ06] Santiago Gil and Damián H. Zanette. "Coevolution of agents and networks: Opinion spreading and community disconnection." *Physics Letters A*, **356**(2):89–94, 2006.

[Hat05] Allen Hatcher. *Algebraic Topology.* Cambridge University Press, Cambridge, United Kingdom, 2005.

[HD18] Ran Huo and Rick Durrett. "Latent voter model on locally tree-like random graphs." *Stochastic Processes and their Applications*, **128**(5):1590–1614, 2018.

[HD19] Ran Huo and Rick Durrett. "The zealot voter model." *The Annals of Applied Probability*, **29**(5):3128–3154, 2019.

[HG93] Bernardo A. Huberman and Natalie S. Glance. "Evolutionary games and computer simulations." *Proceedings of the National Academy of Sciences of the United States of America*, **90**:7716–7718, 1993.

[HK02] Rainer Hegselmann and Ulrich Krause. "Opinion dynamics and bounded confidence models, analysis, and simulation." *Journal of Artificial Societies and Social Simulation*, **5**(3), 2002.

[HK16] Alexander Holiday and Ioannis G. Kevrekidis. "Equation-free analysis of a dynamically evolving multigraph." *The European Physical Journal Special Topics*, **225**:1281–1292, 2016.

[HK20] Leonhard Horstmeyer and Christian Kuehn. "Adaptive voter model on simplicial complexes." *Physical Review E*, **101**(2):022305, 2020.

[HL75] Richard A. Holley and Thomas M. Liggett. "Ergodic theorems for weakly interacting infinite systems and the voter model." *The Annals of Probability*, **3**:643–663, 1975.

[HM13] Donagh Horgan and Colin C. Murphy. "On the convergence of the chi square and noncentral chi square distributions to the normal distribution." *IEEE Communications Letters*, **17**(12):2233–2236, 2013.

[HN06] Petter Holme and M. E. J. Newman. "Nonequilibrium phase transition in the coevolution of networks and opinions." *Physical Review E*, **74**(5):056108, 2006.

[Hol15] Petter Holme. "Modern temporal network theory: A colloquium." *European Physical Journal B*, **88**:234, 2015.

[HS12] Petter Holme and Jari Saramäki. "Temporal networks." *Physics Reports*, **519**:97–125, 2012.

[HS19] P. Holme and J. Saramäki, editors. *Temporal Network Theory.* Springer International Publishing, Cham, Switzerland, 2019.

[HSV90] Jan-Ming Ho, Majid Sarrafzadeh, Gopalakrishnan Vijayan, and Chak-Kuen Wong. "Layer assignment for multichip modules." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **9**(12):1272–1277, 1990.

[HUK10] Sung-Guk Han, Jaegon Um, and Beom Jun Kim. "Voter model on a directed network: Role of bidirectional opinion exchanges." *Physical Review E*, **81**(5):057103, 2010.

[HZD11] Cristián Huepe, Gerd Zschaler, Anne-Ly Do, and Thilo Gross. "Adaptive-network models of swarm dynamics." *New Journal of Physics*, **13**:073022, 2011.

[Isi25] Ernst Ising. "Beitrag zur theorie des ferromagnetismus." *Zeitschrift für Physik*, **31**(1):253–258, 1925.

[JBP15] Lucas G.S. Jeub, Prakash Balachandran, Mason A. Porter, Peter J. Mucha, and Michael W. Mahoney. "Think locally, act locally: Detection of small, medium-sized, and large communities in large networks." *Physical Review E*, **91**(1):012821, 2015.

[JD15] Nicholas A. John and Shira Dvir-Gvirsman. ""I don't like you any more": Facebook unfriending by Israelis during the Israel–Gaza conflict of 2014." *Journal of Communication*, **65**:953–974, 2015.

[Jed17] Arkadiusz Jedrzejewski. "Pair approximation for the q-voter model with independence on complex networks." *Physical Review E*, **95**(1):012307, 2017.

[JLR11] Svante Janson, Tomasz Luczak, and Andrzej Rucinski. *Random Graphs*, volume 45. John Wiley & Sons, Hoboken, New Jersey, USA, 2011.

[JP19] Jonas S. Juul and Mason A. Porter. "Hipsters on networks: How a minority group of individuals can lead to an antiestablishment majority." *Physical Review E*, **99**(2):022313, 2019.

[JS19] Arkadiusz Jędrzejewski and Katarzyna Sznajd-Weron. "Statistical physics of opinion formation: Is it a spoof?" *Comptes Rendus Physique*, **20**(4):244–261, 2019.

[JXC13] M. Ji, C. Xu, C. W. Choi, and P. M. Hui. "Correlations and analytical approaches to co-evolving voter models." *New Journal of Physics*, **15**:113024, 2013.

[KB08] Balazs Kozma and Alain Barrat. "Consensus formation on adaptive networks." *Physical Review E*, **77**(1):016102, 2008.

[KC95] Kei-Yong Khoo and Jason Cong. "An efficient multilayer MCM router based on four-via routing." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **14**(10):1277–1290, 1995.

[KDC19] Yacoub Kureh, Vito Dai, and Luigi Capodieci. "Persistent homology analysis of complex high-dimensional layout configurations for IC physical designs." In *Design-Process-Technology Co-optimization for Manufacturability XIII*, volume 10962. International Society for Optics and Photonics, 2019.

[KGH04] Ioannis G. Kevrekidis, C. William Gear, and Gerhard Hummer. "Equation-free: The computer-aided analysis of complex multiscale systems." *American Institute of Chemical Engineers (AIChE) Journal*, **50**(7):1346–1355, 2004.

[KH08] Daichi Kimura and Yoshinori Hayakawa. "Coevolutionary networks with homophily and heterophily." *Physical Review E*, **78**(1):016103, 2008.

[KMN17] Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov. "Geometry helps to compare persistence diagrams." *Journal of Experimental Algorithmics (JEA)*, **22**:1–20, 2017.

[KMS17] István Z. Kiss, Joel C. Miller, and Péter L. Simon. *Mathematics of Epidemics on Networks: From Exact to Approximate Models*. Springer International Publishing, Cham, Switzerland, 2017.

[KP20] Yacoub H. Kureh and Mason A. Porter. "Fitting in and breaking up: A nonlinear version of coevolving voter models." *arXiv:1907.11608, Physical Review E, in press*, 2020.

[KRB10] Pavel L. Krapivsky, Sidney Redner, and Eli Ben-Naim. *A Kinetic View of Statistical Physics*. Cambridge University Press, Cambridge, United Kingdom, 2010.

[KRM97] M.J. Keeling, D.A. Rand, and A.J. Morris. "Correlation models for childhood epidemics." *Proceedings of the Royal Society of London. Series B: Biological Sciences*, **264**(1385):1149–1156, 1997.

[KS80] Ross P. Kindermann and J. Laurie Snell. "On the relation between Markov random fields and social networks." *Journal of Mathematical Sociology*, **7**:1, 1980.

[KS11] Sandip Kundu and Aswin Sreedhar. "Modeling manufacturing process variation for design and test." In *2011 Design, Automation & Test in Europe*, pp. 1–6. IEEE, 2011.

[Kue16] Christian Kuehn. "Moment closure – A brief review." In Eckehard Schöll, Sabine H. L. Klapp, and Philipp Hövel, editors, *Control of Self-Organizing Nonlinear Systems*, pp. 253–271. Springer International Publishing, Cham, Switzerland, 2016.

[KWD17] Pascal P. Klamser, Marc Wiedermann, Jonathan F. Donges, and Reik V. Donner. "Zealotry effects on opinion dynamics in the adaptive voter model." *Physical Review E*, **96**(5):052315, 2017.

[LA18] Sune Lehmann and Yong-Yeol Ahn. *Complex Spreading Phenomena in Social Systems: Influence and Contagion in Real-World Social Networks*. Springer International Publishing, Cham, Switzerland, 2018.

[LCS16] Deokjae Lee, S. Choi, M. Stippinger, J. Kertész, and B. Kahng. "Hybrid phase transition into an absorbing state: Percolation and avalanches." *Physical Review E*, **93**(4):042109, 2016.

[Len20] Wilhelm Lenz. "Beitrag zum Verständnis der magnetischen Erscheinungen in festen Körpern." *Z. Phys.*, **21**:613–615, 1920.

[Lev05] Harry J. Levinson. *Principles of Lithography*, volume 146. SPIE press, Bellingham, Washington, USA, third edition, 2005.

[LGS99] Koen Lampaert, Georges Gielen, and Willy Sansen. *Analog Layout Generation for Performance and Manufacturability*, volume 501. Springer Science+Business Media, Boston, Massachusetts, USA, 1999.

[Lig99] Thomas M. Liggett. *Stochastic Interacting Systems: Contact, Voter and Exclusion Processes*, volume 324 of *A Series of Comprehensive Studies in Mathematics*. springer–Verlag, Heidelberg, Germany, 1999.

[Lig12] Thomas Milton Liggett. *Interacting Particle Systems*, volume 276. Springer Science & Business Media, New York, New York, USA, 2012.

[LKW19] Eun Lee, Fariba Karimi, Claudia Wagner, Hang-Hyun Jo, Markus Strohmaier, and Mirta Galesic. "Homophily and minority-group size explain perception biases in social networks." *Nature Human Behaviour*, **3**:1078–1087, 2019.

[LLD09] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters." *Internet Mathematics*, **6**(1):29–123, 2009.

[LMS19] Hanbaek Lyu, Facundo Memoli, and David Sivakoff. "Sampling random graph homomorphisms and applications to network data analysis." *arXiv:1910.09483*, 2019.

[LNB19] Hanbaek Lyu, Deana Needell, and Laura Balzano. "Online matrix factorization for Markovian data and applications to network dictionary learning." *arXiv:1911.01931*, 2019.

[Lor05] Jan Lorenz. "A stabilization theorem for dynamics of continuous opinions." *Physica A: Statistical Mechanics and its Applications*, **355**(1):217–223, 2005.

[Lor07] Jan Lorenz. "Continuous opinion dynamics under bounded confidence: A survey." *International Journal of Modern Physics C*, **18**(12):1819–1838, 2007.

[LR07] Renaud Lambiotte and Sidney Redner. "Dynamics of vacillating voters." *Journal of Statistical Mechanics: Theory and Experiment*, **2007**(10):L10001, 2007.

[LR08] Renaud Lambiotte and Sidney Redner. "Dynamics of non-conservative voters." *Europhysics Letters*, **82**(1):18007, 2008.

[LS99] Daniel D. Lee and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." *Nature*, **401**(6755):788–791, 1999.

[LS20] Jens Lienig and Juergen Scheible. *Fundamentals of Layout Design for Electronic Circuits*. Springer, Cham, Switzerland, 2020.

[LSB09] Renaud Lambiotte, Jari Saramäki, and Vincent D. Blondel. "Dynamics of latent voters." *Physical Review E*, **79**(4):046107, 2009.

[LYW16] Kristina Lerman, Xiaoran Yan, and Xin-Zeng Wu. "The "majority illusion" in social networks." *PloS ONE*, **11**(2):e0147617, 2016.

[Ma09] Ning Ma. *Automatic IC Hotspot Classification and Detection Using Pattern-Based Clustering*. PhD thesis, University of California, Berkeley, 2009.

[Mas13] Naoki Masuda. "Voter models with contrarian agents." *Physical Review E*, **88**(5):052803, 2013.

[Mat77] Norman S. Matloff. "Ergodicity conditions for a dissonant voting model." *The Annals of Probability*, pp. 371–386, 1977.

[MBP10] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. "Online learning for matrix factorization and sparse coding." *Journal of Machine Learning Research*, **11**(2):19–60, 2010.

[MCF96] E. Malavasi, E. Charbon, E. Felt, and A. Sangiovanni-Vincentelli. "Automation of IC layout with analog constraints." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **15**(8):923–942, 1996.

[MG05] Mauro Mobilia and Ivan T. Georgiev. "Voting and catalytic processes with inhomogeneities." *Physical Review E*, **71**(4):046102, 2005.

[MGL18] Chul Moon, Noah Giansiracusa, and Nicole A. Lazar. "Persistence terrace for topological inference of point cloud data." *Journal of Computational and Graphical Statistics*, **27**(3):576–586, 2018.

[MGR10] Naoki Masuda, Nicolas Gibert, and Sidney Redner. "Heterogeneous voter models." *Physical Review E*, **82**(1):010103, 2010.

[MMS11] Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. "Zigzag persistent homology in matrix multiplication time." In *SoCG '11: Proceedings of the Twenty-Seventh Annual Symposium on Computational Geometry*, pp. 216–225, New York, New York, USA, 2011.

[MN13] Konstantin Mischaikow and Vidit Nanda. "Morse theory for filtrations and efficient computation of persistent homology." *Discrete & Computational Geometry*, **50**(2):330–353, 2013.

[MNH10] Vincent Marceau, Pierre-André Noël, Laurent Hébert-Dufresne, Antoine Allard, and Louis J. Dubé. "Adaptive networks: Coevolution of disease and topology." *Physical Review E*, **82**(3):036116, 2010.

[Mob03] Mauro Mobilia. "Does a single zealot affect an infinite group of voters?" *Physical Review Letters*, **91**(2):028701, 2003.

[Mob15] Mauro Mobilia. "Nonlinear q-voter model with inflexible zealots." *Physical Review E*, **92**(1):012803, 2015.

[Moo06] Gordon E. Moore. "Progress in digital integrated electronics [technical literature, copyright 1975 IEEE reprinted, with permission.]." *IEEE Solid-State Circuits Society Newsletter*, **11**(3):36–37, 2006.

[Mor] Dmitriy Morozov. " DIONYSUS 2." https://www.mrzv.org/software/dionysus2/.

[MPL17] Naoki Masuda, Mason A. Porter, and Renaud Lambiotte. "Random walks and diffusion on networks." *Physics Reports*, **716–717**:1–58, 2017.

[MPN05] Lauren Ancel Meyers, Babak Pourbohloul, Mark E. J. Newman, Danuta M. Skowronski, and Robert C. Brunham. "Network theory and SARS: Predicting outbreak diversity." *Journal of Theoretical Biology*, **232**:71–81, 2005.

[MS17] Byungjoon Min and Maxi San Miguel. "Fragmentation transitions in a coevolving nonlinear voter model." *Scientific Reports*, **7**:12864, 2017.

[MSC01] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. "Birds of a feather: Homophily in social networks." *Annual Review of Sociology*, **27**(1):415–444, 2001.

[NDB09] Pierre-André Noël, Bahman Davoudi, Robert C. Brunham, Louis J. Dubé, and Babak Pourbohloul. "Time evolution of epidemic disease on finite and infinite networks." *Physical Review E*, **79**(2):026101, 2009.

[New18] Mark E. J. Newman. *Networks*. Oxford University Press, Oxford, United Kingdom, second edition, 2018.

[NKB08] Cecilia Nardini, Balázs Kozma, and Alain Barrat. "Who's talking first? Consensus or lack thereof in coevolving opinion formation models." *Physical Review Letters*, **100**(15):158701, 2008.

[NSC12] Piotr Nyczka, Katarzyna Sznajd-Weron, and Jerzy Cisło. "Phase transitions in the q-voter model with two types of stochastic driving." *Physical Review E*, **86**(1):011105, 2012.

[NVT19] H. Noorazar, K. R. Vixie, A. Talebanpour, and Y. Hu. "From classical to modern opinion dynamics." *arXiv:1909.12089*, 2019.

[Oba18] Ippei Obayashi. "Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology." *SIAM Journal on Applied Algebra and Geometry*, **2**(4):508–534, 2018.

[OPT17] Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather A. Harrington. "A roadmap for the computation of persistent homology." *European Physical Journal—Data Science*, **6**(17), 2017.

[PC86] Richard E. Petty and John T. Cacioppo. "The elaboration likelihood model of persuasion." In *Communication and Persuasion*, pp. 1–24. Springer, New York, New York, USA, 1986.

[PC09] Emanuele Pugliese and Claudio Castellano. "Heterogeneous pair approximation for voter models on networks." *EPL (Europhysics Letters)*, **88**:58004, 2009.

[PCV15] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani. "Epidemic processes in complex networks." *Reviews of Modern Physics*, **87**(3):925–979, 2015.

[PG16] Mason A. Porter and James P. Gleeson. *Dynamical Systems on Networks: A Tutorial*, volume 4 of *Frontiers in Applied Dynamical Systems: Reviews and Tutorials*. Springer International Publishing, Cham, Switzerland, 2016.

[Ran71] William M. Rand. "Objective criteria for the evaluation of clustering methods." *Journal of the American Statistical Association*, **66**(336):846–850, 1971.

[RCS09] Carlos P. Roca, José A. Cuesta, and Angel Sánchez. "Evolutionary game theory: Temporal and spatial effects beyond replicator dynamics." *Physics of Life Reviews*, **6**:208–249, 2009.

[Red19] Sidney Redner. "Reality-inspired voter models: A mini-review." *Comptes Rendus Physique*, **20**(4):275–292, 2019.

[RG13] Tim Rogers and Thilo Gross. "Consensus time and conformity in the adaptive voter model." *Physical Review E*, **88**(3):030102, 2013.

[RPF17] Puck Rombach, Mason A. Porter, James H. Fowler, and Peter J. Mucha. "Core–periphery structure in networks (revisited)." *SIAM Review*, **59**(3):619–646, 2017.

[RSS85] James Reed, Alberto Sangiovanni-Vincentelli, and Mauro Santomauro. "A new symbolic channel router: YACR2." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **4**(3):208–219, 1985.

[SAR08] Vishal Sood, Tibor Antal, and Sidney Redner. "Voter models on heterogeneous networks." *Physical Review E*, **77**(4):041121, 2008.

[SB09] Frank Schweitzer and Laxmidhar Behera. "Nonlinear voter models: The transition from invasion to coexistence." *The European Physical Journal B*, **67**:301–318, 2009.

[SBH20] Michael T. Schaub, Austin R. Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. "Random walks on simplicial complexes and the normalized Hodge 1-Laplacian." *SIAM Review*, **62**(2):353–391, 2020.

[SC04] Vin de Silva and Gunnar Carlsson. "Topological estimation using witness complexes." In Markus Gross, Hanspeter Pfister, Marc Alexa, and Szymon Rusinkiewicz, editors, *SPBG'04 Symposium on Point - Based Graphics 2004*. The Eurographics Association, 2004.

[SCP19] Kazutoshi Sasahara, Wen Chen, Hao Peng, Giovanni Luca Ciampaglia, Alessandro Flammini, and Filippo Menczer. "On the inevitability of online echo chambers." *arXiv:1905.03919*, 2019.

[SDH14] Holly Silk, Güven Demirel, Martin Homer, and Thilo Gross. "Exploring the adaptive voter model dynamics with a mathematical triple jump." *New Journal of Physics*, **16**:093051, 2014.

[Sha20] John Shalf. "The future of computing beyond Moore's law." *Philosophical Transactions of the Royal Society A*, **378**(2166):20190061, 2020.

[SK86] Michael F. Shlesinger and Joseph Klafter. "Lévy walks versus Lévy flights." In *On Growth and Form*, pp. 279–283. Springer, Dordrecht, United Kingdom, 1986.

[SKV09] M Ángeles Serrano, Konstantin Klemm, Federico Vazquez, Víctor M. Eguíluz, and Maxi San Miguel. "Conservation laws for voter-like models on random directed networks." *Journal of Statistical Mechanics: Theory and Experiment*, **2009**(10):P10024, 2009.

[SMD13] Feng Shi, Peter J. Mucha, and Richard Durrett. "Multiopinion coevolving voter model with infinitely many phase transitions." *Physical Review E*, **88**(6):062818, 2013.

[SMD19] Alexander J. Stewart, Mohsen Mosleh, Marina Diakonova, Antonio A. Arechar, David G. Rand, and Joshua B. Plotkin. "Information gerrymandering and undemocratic decisions." *Nature*, **573**(7772):117–121, 2019.

[Smi97] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, California, USA, 1997.

[SPS13] Hiroki Sayama, Irene Pestov, Jeffrey Schmidt, Benjamin James Bush, Chun Wong, Junichi Yamanoi, and Thilo Gross. "Modeling complex systems with adaptive networks." *Computers & Mathematics with Applications*, **65**:1645–1664, 2013.

[SR05] Vishal Sood and Sidney Redner. "Voter model on heterogeneous graphs." *Physical Review Letters*, **94**(17):178701, 2005.

[SRU08] Serguei Saavedra, Felix Reed-Tsochas, and Brian Uzzi. "Asymmetric disassembly and robustness in declining networks." *Proceedings of the National Academy of Sciences of the United States of America*, **105**:16466–16471, 2008.

[SS08] Anja Sturm and Jan Swart. "Voter models with heterozygosity selection." *The Annals of Applied Probability*, **18**(1):59–99, 2008.

[SST19] Meghdad Saeedian, Maxi San Miguel, and Raul Toral. "Absorbing phase transition in the coupled dynamics of node and link states in random networks." *Scientific Reports*, **9**:9726, 2019.

[Sun18] Cass R. Sunstein. *#Republic: Divided Democracy in the Age of Social Media*. Princeton University Press, Princeton, New Jersey, USA, 2018.

[SV10] Jan M. Swart and Karel Vrbenskỳ. "Numerical analysis of the rebellious voter model." *Journal of Statistical Physics*, **140**(5):873–899, 2010.

[SZL18] Marko M. Skoric, Qinfeng Zhu, and Jih-Hsuan Tammy Lin. "What predicts selective avoidance on social media? A study of political unfriending in Hong Kong and Taiwan." *American Behavioral Scientist*, **62**:1097–1115, 2018.

[Tan11] O. Tange. "GNU parallel: The command-line power tool." *;login: The USENIX Magazine*, **36**(1):42–47, 2011.

[TD12] Zunyi Tang and Shuxue Ding. *Advances in Neural Networks – ISNN 2012*, chapter Nonnegative dictionary learning by nonnegative matrix factorization with a sparsity constraint, pp. 92–101. Springer Berlin Heidelberg, Heidelberg, Germany, 2012.

[TDC14] Edward Teoh, Vito Dai, Luigi Capodieci, Ya-Chieh Lai, and Frank Gennari. "Systematic data mining using a pattern database to accelerate yield ramp." In *Design-Process-Technology Co-optimization for Manufacturability VIII*, volume 9053, p. 905306. International Society for Optics and Photonics, 2014.

[TMP12] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. "Social structure of Facebook networks." *Physica A: Statistical Mechanics and its Applications*, **391**(16):4165–4180, 2012.

[UKB11] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. "The anatomy of the Facebook social graph." *arXiv:1111.4503*, 2011.

[VA18] Allan R. Vieira and Celia Anteneodo. "Threshold q-voter model." *Physical Review E*, **97**(5):052106, 2018.

[VES08] Federico Vazquez, Víctor M. Eguíluz, and Maxi San Miguel. "Generic absorbing transition in coevolution dynamics." *Physical Review Letters*, **100**(10):108702, 2008.

[VR04] F. Vazquez and S. Redner. "Ultimate fate of constrained voters." *Journal of Physics A: Mathematical and General*, **37**(35):8479, 2004.

[WDA02] Gérard Weisbuch, Guillaume Deffuant, Frédéric Amblard, and Jean-Pierre Nadal. "Meet, discuss, and segregate!" *Complexity*, **7**(3):55–63, 2002.

[WDC12] Lynn T-N Wang, Vito Dai, and Luigi Capodieci. "Pattern matching for double patterning technology-compliant physical design flows." In *Design for Manufacturability through Design-Process Integration VI*, volume 8327, p. 832708. International Society for Optics and Photonics, 2012.

[WS98] Duncan J. Watts and Steven H. Strogatz. "Collective dynamics of 'small-world'networks." *Nature*, **393**(6684):440–442, 1998.

[XLC17] Fei Xiong, Yun Liu, and Junjun Cheng. "Modeling and predicting opinion formation with trust propagation in online social networks." *Communications in Nonlinear Science and Numerical Simulation*, **44**:513–524, 2017.

[YBR17] Jung Hwan Yang, Matthew Barnidge, and Hernando Rojas. "The politics of "Unfriending": User filtration in response to political disagreement on social media." *Computers in Human Behavior*, **70**:22–29, 2017.

[YK82] Takeshi Yoshimura and Ernest S. Kuh. "Efficient algorithms for channel routing." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **1**(1):25–35, 1982.

[YOA13] Ercan Yildiz, Asuman Ozdaglar, Daron Acemoglu, Amin Saberi, and Anna Scaglione. "Binary opinion dynamics with stubborn agents." *ACM Transactions on Economics and Computation*, **1**:4, 2013.

[YPM15] Bei Yu, David Z Pan, Tetsuaki Matsunawa, and Xuan Zeng. "Machine learning and pattern matching in physical design." In *The 20th Asia and South Pacific Design Automation Conference*, pp. 286–293. IEEE, 2015.

[ZBS12] Gerd Zschaler, Gesa A. Böhme, Michael Seißinger, Cristián Huepe, and Thilo Gross. "Early fragmentation in the adaptive voter model on directed networks." *Physical Review E*, **85**(4):046107, 2012.

[ZC05] Afra Zomorodian and Gunnar Carlsson. "Computing persistent homology." *Discrete & Computational Geometry*, **33**(2):249–274, 2005.

[ZG06] Damián H. Zanette and Santiago Gil. "Opinion spreading and agent segregation on evolving networks." *Physica D: Nonlinear Phenomena*, **224**(1–2):156–165, 2006.

[ZLL18] Wei Zhang, Jinglei Lv, Xiang Li, Dajiang Zhu, Xi Jiang, Shu Zhang, Yu Zhao, Lei Guo, Jieping Ye, and Dewen Hu. "Experimental comparisons of sparse dictionary learning and independent component analysis for brain network inference from fMRI data." *IEEE Transactions on Biomedical Engineering*, **66**(1):289–299, 2018.

[ZSS17] Qinfeng Zhu, Marko Skoric, and Fei Shen. "I shield myself from thee: Selective avoidance on social media during political protests." *Political Communication*, **34**:112–131, 2017.