2024

Robust interventions in network epidemiology

Erik Weis
University of Vermont

Robust interventions in network epidemiology

A Thesis Presented

by

Erik Weis

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Master of Science
Specializing in Complex Systems and Data Science

January, 2024

Defense Date: October 2, 2023
Defense Examination Committee:

Jean-Gabriel Young, Ph.D., Advisor
Laurent Hébert-Dufresne Ph.D., Advisor
Sarah Nowak, Ph.D., Chairperson
Jeffrey S. Buzas, Ph.D.
Holger Hoock, DPhil, Dean of the Graduate College

Abstract

Which individual should we vaccinate to minimize the spread of a disease? Designing optimal interventions of this kind can be formalized as an optimization problem on networks, in which we have to select a budgeted number of *dynamically important* nodes to receive treatment that optimizes a dynamical outcome. Describing this optimization problem requires specifying the network, a model of the dynamics, and an objective for the outcome of the dynamics. In real-world contexts, these inputs are vulnerable to misspecification—the network and dynamics must be inferred from data, and the decision-maker must operationalize some (potentially abstract) goal into a mathematical objective function. Moreover, the tools to make reliable inferences—on the dynamical parameters, in particular—remain limited due to computational problems and issues of identifiability. Given these challenges, models thus remain more useful for building intuition than for designing actual interventions. This thesis seeks to elevate complex dynamical models from intuition-building tools to methods for the practical design of interventions.

First, we circumvent the inference problem by searching for robust decisions that are insensitive to model misspecification. If these robust solutions work well across a broad range of structural and dynamic contexts, the issues associated with accurately specifying the problem inputs are largely moot. We explore the existence of these solutions across three facets of dynamic importance common in network epidemiology.

Second, we introduce a method for analytically calculating the expected outcome of a spreading process under various interventions. Our method is based on message passing, a technique from statistical physics that has received attention in a variety of contexts, from epidemiology to statistical inference. We combine several facets of the message-passing literature for network epidemiology. Our method allows us to test general probabilistic, temporal intervention strategies (such as seeding or vaccination). Furthermore, the method works on arbitrary networks without requiring the network to be “locally tree-like”. This method has the potential to improve our ability to discriminate between possible intervention outcomes.

Overall, our work builds intuition about the decision landscape of designing interventions in spreading dynamics. This work also suggests a way forward for probing the decision-making landscape of other intervention contexts. More broadly, we provide a framework for exploring the boundaries of designing robust interventions with complex systems modeling tools.

The question isn't “what are we going to do”, the question is “what aren't we going to do?”

-Ferris Bueller

Acknowledgements

First and foremost, thank you to my advisors, Jean-Gabriel Young and Laurent Hébert-Dufresne, for their strong support of my work throughout my time at UVM. In particular, thank you to Jean-Gabriel for young for your patience and dedication in teaching me the ropes of science. Thank you also to the fantastic community at the Vermont Complex Systems Center for the amazing and ever-stimulating environment. Finally, thank you to my committee members for your patience in reviewing my work and for your helpful feedback. I'm extraordinarily grateful for your support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Mathematical models help us to better understand the behavior of the systems we study. That understanding comes from two features of models. The first is models' ability to abstract away detail and thereby expose the fundamental mechanisms that drive the system's behavior. The second is that models are sandboxes through which we can explore the many possible outcomes that may arise due to randomness in the system. Such exploration is especially useful when accidents of history drive the system towards vastly different outcomes.

As an example, consider the spread of a disease through a population. When a single individual becomes infected with a disease, we might get lucky and find that none of the individuals that come into contact with this individual become infected, resulting in the outbreak dying out quickly. By contrast, we might get particularly unlucky if another did attract the disease and was then due to attend a large gathering the next day, which could lead to a large and uncontrollable outbreak. It has been shown that models of the spread of disease exhibit extremely heterogeneous outcome distributions [1].

We can contrast this rich picture of possible outcomes produced by our model with the knowledge gained from observational data, where data is usually available for only a single outbreak. To truly understand the underlying behavior of the system, we would need to experience outbreaks many times to understand how a disease spreads. Doing so would, of course, be highly unethical, as our goal should be to minimize the spread of disease, not allow it for the purposes of data collection. Beyond ethics, real-world systems present many reasons that limit our ability to gather representative experimental data. In these situations, turning to models helps us explore the entire range of possible outcomes and their probabilities.

Over and above providing a richer understanding of some system of interest, models can serve as testing grounds for policy interventions. Mathematical models are specified by a potentially huge number of quantitative variables. Some of these parameters will be out of our control and are best set to values that accurately represent the real world. Other parameters, however, could, in principle, be changed. Changing the values of these *decision parameters* amounts to simulating what would happen if we enacted a particular intervention. In the case of epidemics, non-pharmaceutical interventions such as vaccination and social distancing reduce the number of people who get the disease. By specifying who is immunized or social-distanced, we can simulate many outbreaks under these conditions. Calculating the size of these outbreaks allows us to evaluate various vaccine rollout strategies. The results of these *in silica* experiments can then inform our choice about what intervention to implement in reality.

The potential to use models in decision-making raises the stakes for our belief about how well their behavior mimics reality. From a statistical perspective, specifying the dynamics of a model constitutes making a causal assumption about the behavior of the system. If we want to use models to make decisions about policy interventions, we must remain confident that models meaningfully represent the world, as ,isspecification of our model could lead to sub-optimal decisions. Moreover, bias in our model specification could result in biased outcomes that disproportionately favor some parts of the system over others. Hence, we should approach model-based decisions with caution and be able to account for our inevitable inability to represent the true dynamics of our system perfectly.

This thesis focuses on the potential for using complex systems models in decision-making. How can we feel confident that our models do not lead us astray or introduce bias in our decision-making? What are the stakes for model misspecification? To explore these questions, we focus on models of spreading processes on networks, as they provide a prototypical example of the complex network-dynamic behavior that we experience in many real-world policy systems.

In the remainder of this chapter, we first introduce models of network spreading dynamics. We then introduce a series of intervention design problems that apply when considering interventions on spreading dynamics and close by highlighting the challenges this thesis seeks to address.

1.1 Dynamical processes on networks

Many complex systems are comprised of many entities, which we refer to as agents. These agents are often people but could equivalently be a huge range of other elements, such as firms, countries, and governments in economics; proteins, cells, organisms, or species in biology; servers, and robots in autonomous systems. Each of these agents maintains an internal state X and interacts with the world according to its state and that of other agents in the system. In some systems, the state of any individual agent depends on the state of every other in the system. In others, however, the behavior of the system is distributed, meaning agents depend only on a small subset of others.

Consider the example of a group of people deciding whether to attend a social gathering. Each individual will decide to attend based on whether they think others will also be in attendance. However, we could reasonably assume that not every individual's decisions matter. Any given individual primarily cares whether their friends will attend the party. Their behavior depends less directly (if at all) on those they do not know. We refer to the specific ways in which agents' behaviors depend on each other as a *network*. In social contexts, the meaning of a social network is clear; each individual has a friendship connection to others in the network. The study of social networks—which reaches back to the 1930s [2] but proliferated in the 1970s—arose to study the strong patterns in the way we form friendships.

Beyond friends, sociological research on networks extends to a wide range of social and cultural processes, including the adoption of beliefs [3], the diffusion of innovations [4], information spreading [5], consensus and group decision-making [6], market behavior [7], and science [8] [9]. More recently, in the late 1990s and early 2000s, researchers from applied mathematics and statistical physics have expanded the scope of network science to an even larger range of systems such as the internet [10] [11], the brain [12], food webs [13], power grids [14] [15], and many more. The diversity of systems exhibiting such similar structures has led network scientists to explore the potentially universal role that network structure plays in the behavior of systems, such as scale-free properties [16] and critical behavior [17].

Networks can be represented mathematically by a graph $G = (V, E)$. Each node i in the set V refers to a specific agent in the system. Similarly, we use a pairwise edge (i, j) to signify that the state X_i of agent i influences that of agent j. The set E of all such pairs, along with the node set V, fully defines the graph.

The set of nodes that share an edge with node i is referred to as the *neighbors* of i, which we denote ∂i. Dynamically, this means that i's state X_i is a function of the states of its neighbors $X_{\partial i}$, which we write

$$X_i = f_i(X_{\partial i}). \tag{1.1}$$

The notion that states influence each other has many interpretations. In some cases, the states are correlated because some physical entity is literally spreading across the network. This is true for the spread of a disease, where an interaction between two agents leads to one infecting another. Another example of this type is power grids, where power is distributed across terminals, and energy available to one node is related to how much energy is distributed to it through its neighbors. A second interpretation is that of influence, where an individual agent updates its own state in response to others. Opinion dynamics is a common example, where an agent adjusts their opinion because of exposure to the opinions of their neighbors. Regardless of the interpretation, Eq. 1.1 abstractly defines a network dynamical process.

The discipline of network science seeks to understand the way dynamics *on* a network are affected by the particular structure *of* the network. When all agents respond to their neighbors in the same way, i.e., $f_i = f \ \forall i \in V$, network structure is the defining feature of how the states X_i evolve. This assumption is true for models of the spread of disease or information, where we assume the disease spreads according to some biological mechanism that is (roughly) the same for all individuals. We will discuss this point further in the next section, where we describe specific models of spreading processes. The important point is that network structure indeed plays a critical role in the outcomes of these models.

1.1.1 Contagion models on networks

One important driver of network science research has been its application to the spread of disease [18], and the last few years of the COVID-19 pandemic have painfully illustrated the importance of such work. Researchers have used sophisticated network science models to evaluate various intervention strategies [19] [20]. Ironically, the same pandemic has also highlighted some other important spreading processes. Throughout the pandemic, misinformation about the nature of the disease spread rampantly through online social networks. Furthermore, people seemed to adopt patterns of behavior, such as mask-wearing and social distancing, based on the behavior of their peers. Even policy seemed to spread throughout the network as governments searched for the right intervention strategies [21]. COVID-19 also impacted the economic sphere, where the fragility of supply chains became apparent [22] [23].

Mathematical models of spreading processes trace their origins to a model of epidemics put forth by Kermack and McKendrick in 1927 [24]. A common simplification of their model, which nonetheless contains all the relevant details, is known as the SIR model, so-called because individuals could occupy one of three distinct states: susceptible (S), infected (I), or recovered (R). Susceptible agents can be infected, infectious agents can recover from the disease, and recovered agents are resistant to further infection. The SIR model assumes that all individuals in a population interact randomly, meaning the chance that a susceptible individual interacts with an infected individual is proportional to the fraction of infected individuals in the population. This assumption is known as *random mixing* and implies that each individual in the population is identical. Because this infection probability depends only on the average state of the system, the SIR model is known as a *mean-field model*.

To understand the SIR model formally, let $\mathfrak{S}(t)$ and $I(t)$ represent the fraction of susceptible and infected individuals in the population, respectively, at time t. During each interaction, susceptible individuals come into contact with an infected individual with probability $I(t)$. Those susceptible individuals that are exposed to an infectious individual become infected with probability β, a parameter that specifies the contagiousness of the disease. Hence, the total probability that a randomly chosen individual is infected at time t is the probability they are susceptible, $\mathfrak{S}(t)$, multiplied by the probability they interact with an infected individual $I(t)$ multiplied by the probability that they are infected β. Additionally, a fixed proportion α of infected individuals recover from the disease at each time step. Using the above logic and benefiting from random-mixing, we can write the straightforward set of differential equations that completely describe the average state of the system:

$$\begin{aligned} \frac{d\mathfrak{S}}{dt} &= -\beta\mathfrak{S}(t)I(t) \\ \frac{dI}{dt} &= \quad \beta\mathfrak{S}(t)I(t) - \alpha I(t) \end{aligned} \tag{1.2}$$

The problem with the standard SIR model is that random mixing is an unrealistic assumption. Network science expanded on this model by letting the same SIR dynamics take place on a network. Now, at each time step, newly infected individuals

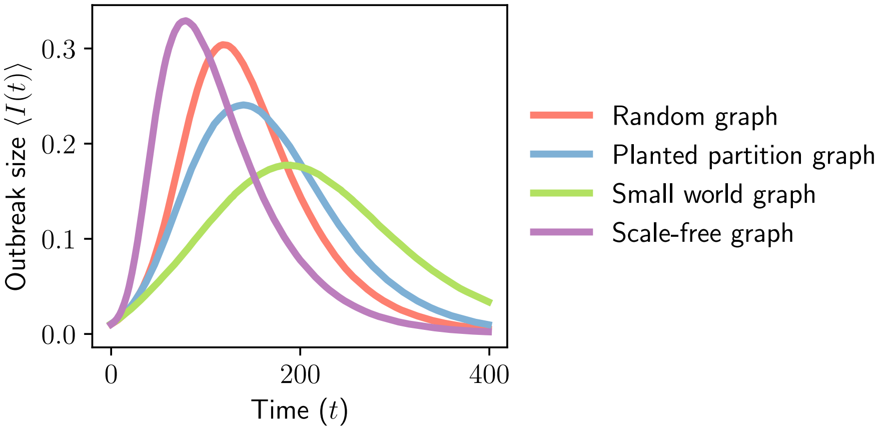

Figure 1.1: An SIR process on three graphs with the same density of contacts: 1000 nodes and 5000 edges. The same process shows extremely different results depending on the contact patterns of individuals in the network.

infect their *neighbors* with a fixed and independent probability β. If we design our network such that connections between individuals are completely random, we recover the behavior of the randomly mixed SIR model. However, if we introduce structural features observed in real-world networks, we observe vastly different behavior to the randomly mixed case, as shown in Figure 1.1.

The network SIR model is the foundation for a great many model variations that capture additional detail. For example, not all individuals are equally susceptible to a disease nor likely to infect another. Most generally, the probability of spread when two individuals interact is some function of the properties of both individuals and their interaction. We can represent this by defining the probability of transmission between two nodes i and j as β_{ij}. Another implicit assumption in the mean-field SIR model is that the number of recoveries in the system for a given time interval is proportional to the number of infected individuals in the system. This assumption is necessary to make the mean-field SIR model Markovian and therefore amenable to description by differential equations. By implication, the recovery time for a randomly chosen individual follows an exponential distribution—which has this Markovian property. Moreover, the most likely common recovery time is zero, an unlikely assumption in practice. We can relax this assumption by assuming a general distribution of recovery times. For our purposes, we will assume that the recovery time is constant (i.e. described by a constant random variable). While this assumption is also not perfectly realistic, it does represent an improvement over the exponential recovery times. Furthermore, it provides a great deal of mathematical convenience. In the next section, assuming constant recovery times will allow us to make an exact equivalence between the SIR model and another mathematical process known as bond percolation [25, 26]. Studying the spreading processes as a percolation will thus provide helpful insights into the system's behavior.

1.1.2 The independent cascade model

Percolation was originally studied in the context of liquid trickling through a porous solid, such as rocks. If the solid was porous enough, a path through the solid would allow liquid to *percolate* through it. One example is found in brewing coffee, where water fails to percolate through coffee if it is ground too finely or tamped down too hard. Mathematically, such problems were studied on regular lattices, where sites were randomly removed from the lattice to signify the presence of a solid that prevents any flow. With no sites on the lattice, the liquid will flow perfectly; with every site occupied, no liquid will percolate. Such processes demonstrate a phase transition [27]. That is, the lattice would percolate only when the fraction of randomly removed sites was below a certain value.

A similar process can be defined on networks. While *site* percolation considers the removal of nodes, we consider *bond percolation*, where edges are removed with probability $1 - p$. If p is low, most edges will be removed, and the network will be broken apart into many disconnected components. With a high p, the network will stay largely connected to each other. On networks, we observe a similar phase transition, where there exists a critical value p_c that characterizes the transition.

It turns out that the process of percolation also maps exactly to the spreading processes of the following model. We choose a node at random to act as the seed of the contagion. This seed node infects each neighbor with probability p. In the following time step, each of these neighbors infects their remaining neighbors (besides the seed node) with probability p. The process proceeds until all nodes have been infected or no infections occur in a time step.

Such a spreading process is equivalent to the model of percolation in the following way. Create a random instance of the percolation process by removing each edge in the network with probability $1 - p$. The remaining configuration will contain only edges that have (or could have) spread the disease. All that remains is to randomly choose a seed node and observe which nodes are reachable (via an edge) in the given instance of the percolation process. This mapping allows several convenient mathematical advantages, which we'll see in Chapter 2.

The model can be further expanded by removing an edge connecting nodes i and j with probability $1 - p_{ij}$ instead of with the uniform probability $1 - p$, thus leading to much more variable and diverse dynamical outcomes. This more general model is often referred to as the independent cascade (IC) model [28, 29] because the probability of spread through each of the network is independent of all others, though heterogeneous.

For the purposes of this thesis, we consider the only case of a fixed p for all edges. We note again that this restricted model is quite general, being equivalent to SIR-processes when the recovery times are constant [25]. When this assumption fails, the mapping is still quite good, if not exact, in most regimes [26]. Hence, our results should have broad applicability to disease models on networks.

Beyond the spread of disease, many other types of contagions exist. In particular, nodes might modify their state based on the state of more than one of their neighbors. Rather than a node updating its state to I with probability p each time a neighbor becomes infected, we say that a node updates its state I based on the states of all its neighbors. One example of such a model is the k-threshold model, which requires that at least k of an individual's neighbors become infected for an agent to become infected [30, 31, 32]. Such models are often used to describe the spread of behavioral and cultural norms, where individuals seek to adopt behaviors more tentatively. For the purposes of this thesis, we consider only the independent cascade model and leave the exploration of these other dynamics to future work.

1.1.3 The outcomes of contagion models

Now that we've described the dynamics of contagion models on networks, we can now turn to describing the outcomes of these processes mathematically. Unlike the mean-field SIR model, the precise outcome of a spreading process is not determined simply by the fraction of individuals in each state. We first define the end of a contagion process as the moment when the number of new infections reaches zero, at which point no further nodes can become infected. Let $\mathbf{X}$ be the final state of the nodes as a result of a stochastic spreading process. For each node, let X_i be the infection status of node i, which can be defined as a boolean random variable

$$X_i = \begin{cases} 1 & \text{i is infected,} \\ \\ 0 & \text{i is susceptible.} \end{cases} \tag{1.3}$$

We are most often be interested in the average state of a node over many instances of the process, which we describe as the marginal probability of infection

$$\pi_i = \sum_{\mathbf{X}} X_i P(\mathbf{X}), \tag{1.4}$$

where $P(\mathbf{X})$ is the probability that particular outcome occurred. The expected proportion of infected individuals at the end of the process is

$$\langle I \rangle = \sum_{\mathbf{X}} \sum_i X_i P(\mathbf{X}) = \sum_i \pi_i. \tag{1.5}$$

These expected quantities are easy to write down but difficult to compute in practice. The support for $\mathbf{X}$ is a large space, with 2^n possible states, where n is the number of nodes, making exact calculation of these quantities seemingly impossible. To overcome this difficulty, we can rely on Monte Carlo sampling, meaning we run a large number of simulations and make the approximation

$$\langle I \rangle \approx \frac{1}{N} \sum_{k=1}^{N} \sum_i X_i^{(k)}, \tag{1.6}$$

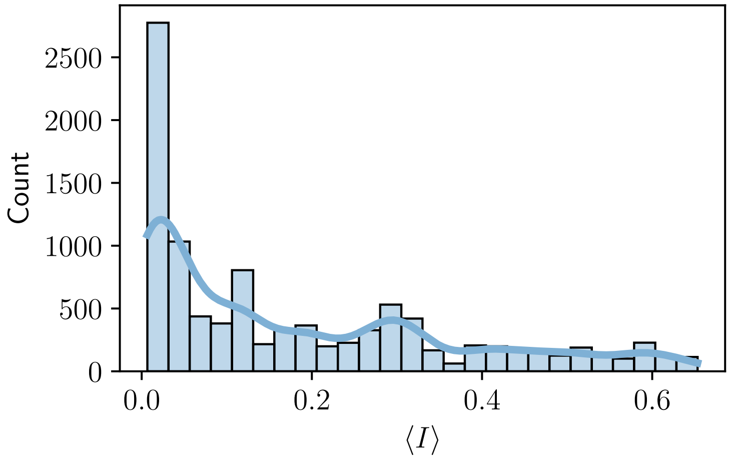

Figure 1.2: The distribution of outbreak sizes for a SIR process on an Erdős-Réyni network of $N = 150$ nodes and edge density $p = 0.05$ over 10^4 simulations.

where $X_i^{(k)}$ is the outcome on sample k. In Chapter 2, we'll develop a fast algorithm that allows us to estimate these quantities more efficiently.

In Chapter 3, we take another approach to avoid sampling altogether using the technique of message passing, which will allow us to calculate the expected value analytically. Such a technique will be useful when we test various interventions for the spreading process, where we'll need to calculate the values of equation 1.5 for many possible intervention candidates. We now turn to the problems that make such testing necessary.

1.2 Node importance in networks

Given a spreading process on a social network, it is often the case that some nodes have considerably more influence on the outcome of the system than others. We can imagine that a meme shared on social media by a celebrity is far more likely to spread than the same meme shared by an average person. Determining which nodes have an outsized influence on the system is referred to as *node importance.* Broadly, notions of node importance fall into two categories: structural importance and dynamical importance.

Structural importance refers to nodes that are important with respect to the way nodes are connected to each other–their structure. For example, we might assume that nodes with many connections are much more important than those with few connections. This definition makes sense in the context of an online social media platform, where we might deem users with millions of followers more important than those with a few dozen. The number of a node's connections is called its degree, and the ranking of nodes by this measure is known as *degree centrality.* Another common example is to define a node's importance in terms of the importance of its neighbors. In the social network context, an individual is important if their personal connections are also important. This recursive but self-consistent definition of importance is known as *eigenvector centrality* [18]. In both these examples, we measure importance purely based on the structure of relationships between individuals. In doing so, we ignore the fact that information is continually spread through the network and focus instead on the structure of the connections through which that information spreads.

By contrast, *dynamical importance* is defined with respect to a specific dynamical process happening on the network. Furthermore, dynamical importance can be defined with respect to some *objective* about the outcome of the dynamical process that takes place in the system. Though structural features should give an indication of a node's importance in a dynamical process, they do not have all the information about the underlying system. As such, structural importance has been shown to be an imperfect predictor of dynamical importance for general notions of network dynamics [33].

1.2.1 Notions of dynamical importance in network epidemiology

A definition of importance depends again on what one might hope to achieve with an intervention. Returning to the spread of disease on a network, consider three distinct but related notions of dynamical importance, as described in [34].

Suppose we have only k doses of a vaccine available to protect against an outbreak. Which nodes should receive the vaccine so that when an outbreak occurs, its total size is minimized? This problem is known as *targetted immunization.*

Our second notion of dynamical importance concerns the case where we have resources to surveil a subpopulation of k nodes, such that when one tests positive for the disease, we detect the presence of an outbreak. Naturally, we should choose a set of nodes that are likely to be infected early on in the spreading process. This problem is known as *sentinel surveillance.*

Our third notion of dynamical importance concerns the search for nodes that are best at spreading the disease widely through the network. Most often, this problem is considered not in the context of epidemics but for viral marketing, where the goal is to maximize the spreading through the network. More concretely, which set S of k nodes will lead to the largest overall outbreak on average? This problem is known as *influence maximization.* While this notion of importance differs from targeted immunization, we might expect that influential spreaders are also the nodes we should immunize. We'll discuss the relationship between different notions of importance in Chapter 2.

1.2.2 Optimal interventions

These three definitions of dynamical importance imply the existence of a quality function that allows us to compare the quality of different intervention sets. This quality is defined with respect to the outcomes of the system. In the context of resource-constrained interventions, an intervention set S of k nodes, chosen from the n possible choices, that receive some treatment. For example, if we are doing vaccination, the outcome on which we should evaluate the quality of S is the total outbreak size $\langle I \rangle$, which we want to be as small as possible.

In the context of resource-constrained interventions, our decision constitutes choosing a set of k nodes from the n possible choices. The number of possible sets is $\binom{n}{k}$, a huge space that is computationally infeasible to search exhaustively. (It grows exponentially with system size when measured in bits.) The problem of choosing k dynamically important nodes is a challenging and nonlinear *combinatorial optimization problem.*

A number of *global optimization* strategies can be used to address problems, such as simulated annealing and evolutionary documents [35]. In addition, some approximation algorithms exist in special cases, such as a greedy algorithm that provides $1-1/e$–approximation for the problem of influence maximization. For our purposes, we adopt this greedy algorithm for efficiency and consistency across problems. While the focus of our work is not on the optimization itself, it is worth noting that optimization poses a challenge for problems of this type.

1.3 Model misspecification

Our ability to find optimal intervention problems is founded on a perfect knowledge of the problem's key parameters, such as the network structure and dynamics. We now turn to the issues that could potentially arise if we misspecify the network dynamic model and ways to tackle them.

One possible way of handling modeling specification is through statistical inference. If we can quantify uncertainty about the parameters of our model, decision-makers have an adequate tool to use the model confidently and with knowledge of the quality of its predictions. Ideally, we could gather a wealth of data on the system to learn the parameters of the model. In the case of epidemiology, such a procedure would involve inferring two things: (1) the structure of the physical contact network that provides pathways for the spread of the disease and (2) the parameters of a model that shows how people get infected when contact between a susceptible and infections individual occur. Particularly with the latter, making reliable inferences of the infection rate (or equivalently, the effective reproduction number R_0) is extremely challenging to do in practice. The first problem is that we usually only have summary statistics, such as number of counts per day, and do not have the underlying specifics of how the disease spreads. The second problem concerns the computational infeasibility of writing down a closed-form likelihood for the system. In practice, fitting these models to data requires some kind of likelihood-free inference, which requires an extreme number of numerical simulations [36].

To make matters worse, there often exists an even more fundamental problem. These models are not usually identifiable with the data available to us. Often, with policy issues, we only have access to summary statistics and aggregated data, which is usually insufficient to infer the exact underlying mechanisms of the system [37]. For the spread of disease, the data collected on who is infected is often far too noisy. Unlike the computational issues, the limitations of the model and data cannot be surmounted by brute force. Supposing we can improve on these methods, it remains a challenge to use the parameter uncertainty to make decisions. We return to such questions in Chapter 4.

1.4 Contributions of thesis

The goal of this thesis, broadly speaking, is to study node importance under problem uncertainty. In doing so, however, our work sidesteps the inference problem altogether. Instead, we attempt to provide insights to decision-makers in a different way, by exploring the structure of the decision space. Suppose there exists an intervention that is optimal regardless of the parameter space in question. Such a scenario is good news for decision-makers since failing to infer the correct model parameters would not lead to any loss of quality for the intervention. More generally, *robust interventions* are those that work well in a variety of dynamical contexts. Finding robust solutions is somewhat related to other mathematical tools, such as sensitivity analysis, where we focus on finding how small changes in the model's specification affect outcomes, both qualitatively and quantitatively. However, our work focuses not only on how the model outcomes themselves change but how optimal decisions change.

In 2, we consider the robustness of interventions in the context of network epidemiology. Doing so requires finding the optimal decision for the entire parameter space. Such optimization is challenging to do efficiently when testing and evaluating the quality of different interventions holds the same computational challenges as using them for statistical inference. In Chapter 3, we explore the possibility of testing the quality of interventions more efficiently with an applied probability theory technique known as message passing. We expand the existing literature on message-passing techniques for disease dynamics, making them suitable for probing the kinds of questions explored in Chapter 2. We conclude by examining how a full uncertainty-aware decision-making pipeline would work for network epidemiology and discuss the possibilities of implementing such a thing in practice.

Chapter 2

Evaluating node importance in the face of model errors

An information-rich and networked society offers countless opportunities to make targeted yet impactful interventions, from micro-targeting campaigns designed to affect small behavior changes to large-scale immunization campaigns aiming to contain emerging contagions. Ample theoretical and applied work has taught us how to select these interventions effectively, with results collected under the broad umbrella of "nodal importance problems" outlined in Chapter 1. In importance problems, we model a system of interest as a graph whose edges support the transmission of a contagion—of information, of behavior, or of a pathogen. We then imagine spending a limited budget to select dynamically important nodes which, when targeted with an intervention, will modify the system's outcome. Perhaps most famous among these abstract problems is influence maximization [38], where the goal is to seed a subset of nodes with information so as to maximize the extent of a contagion [39]. But influence maximization is just one example of dynamical importance [34]; see Figure 2.1. In the context of epidemiology, we may seek to limit the spread of a disease by means of targeted immunization [40]. Yet another possible objective may be sentinel surveillance [41], where we choose a subset of nodes to monitor such that we may detect an outbreak as soon as possible when it occurs. Thus, a robust understanding of dynamical importance can guide interventions in distributed systems of all kinds, including financial systems [42, 43], supply chains [44], power grids [45], the spread of information on social media [46], and, of course, marketing [39].

As we have alluded to in the introductory chapter, nodal importance is largely a solved problem. On the modeling side, we know of numerous models that incorporate dynamics believed to emulate real-world spread, whether it be the simple independence cascade model [28, 29] possibly with assortative propensity for spread [48], or complex behavior requiring, for example, adoptions by a certain fraction of an individual social circle to move forward [49, 30, 32, 50]. On the optimization side, we now know of several algorithms of varying degrees of complexity and accuracy, thus allowing intervention designers to trade off compute and results' quality [51]. For instance, a greedy approach to influence maximization is known to cheaply [52] provides a $1 - 1/e$ approximation to the optimal solution of the NP-Hard problem of finding a set of k maximally influential nodes [38]. At the other end of the spectrum, an exhaustive search finds truly optimal solutions though its run-time scales exponentially with problem size. And various meta-heuristics—like neighbor-hop-based genetic algorithms [35], divide and conquer strategies [53], or simulated annealing searches [54]—strike a balance. In a pinch, one can even use structural methods that exploit a correlation between the structural and dynamical importance of nodes to construct cheap solutions (though this correlation can be weak [33]).

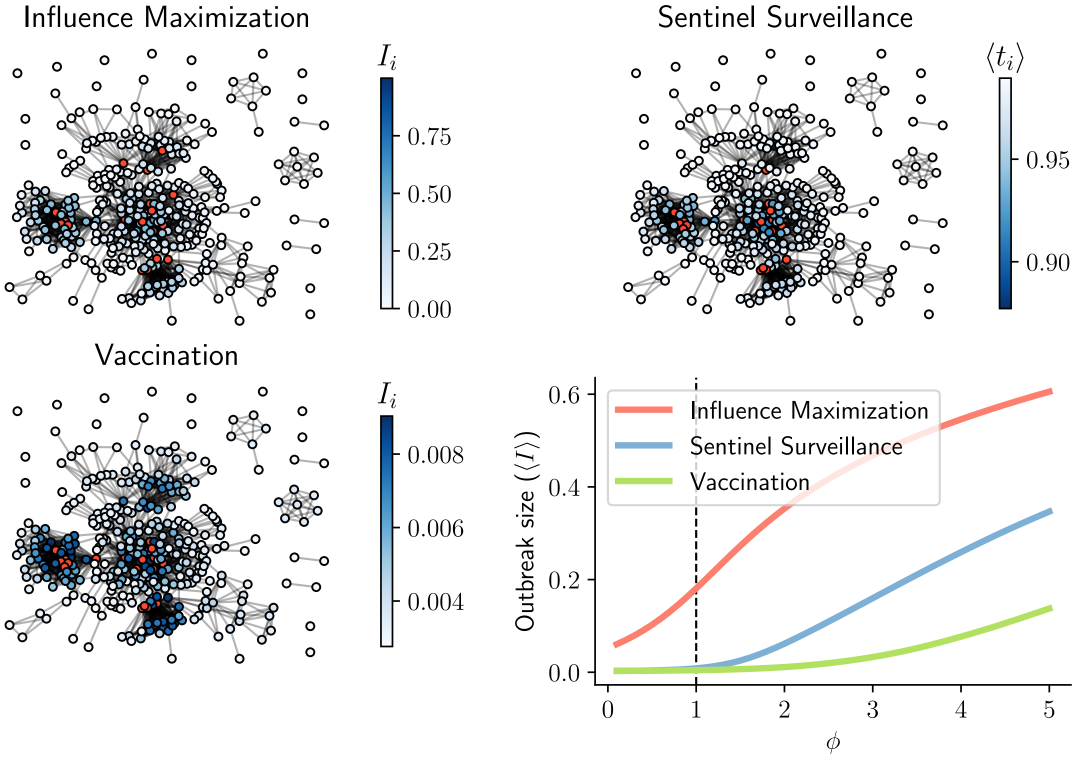

Figure 2.1: Dynamical importance problems on networks. Each panel shows the same social network of 362 Facebook users [47], and highlights different sets of dynamically important nodes in red. For influence maximization and vaccination (left), the optimized outcome is the expected outbreak size $\langle I \rangle = \sum_i I_i$ where I_i is the marginal probability that nodes i become contagious at any given point. These marginal probabilities are shown as shades of blue. For sentinel surveillance, the minimized outcome is the expected infection of the important nodes, $\langle t_i \rangle$. The bottom-right panel shows the expected outbreak size over a range of infection rates $\phi = p/p_c$, where p_c is the critical threshold of the dynamics in the absence of an intervention.

A rich landscape of methodological work on nodal importance thus begs the question of why these methods have not been deployed more visibly and widely. One reason is possibly the wide gap between models and the real world: Models can be misspecified, and correct specifications matter for optimal interventions [48]. Misspecification can lurk in the structure [55] or dynamical model parameters [56, 48, 50], or both [57, 55]. So-called "robust nodal importance" problems have thus attracted growing attention, particularly the influence maximization; see Ref. [57] and references therein for an overview of algorithms designed to circumvent adversarial and random noise. Another reason for the lack of widespread deployment of nodal importance techniques could be uncertainty around what one should even optimize for. Superspreaders are not necessarily good immunization targets [58, 59], and superspreaders may not be good targets for networked surveillance. And thus, optimizing for the wrong facet of a node's dynamical importance may lead to incorrect interventions [34].

This chapter addresses the problem of dynamical node importance under misspecification of the model and optimization objective. Our approach is empirical rather than theoretical. Hence, instead of designing robust objectives with worst-case guarantees in mind, we use a simulation on empirical networks to check how various decisions fare, on average, when they are determined using incorrect assumptions. In doing so, we are following in the steps of Holme [34], who used a similar methodology to highlight the various faces of node importance in small networks ($n \leq 7$ nodes). Unlike Holme, however, we scale that analysis to realistic networks whose structure is determined by human behavior. This comparison study is made possible by new fast Monte Carlo simulation techniques developed by adapting the Newman-Ziff percolation algorithm described in Ref. [60].

2.1 Methods

2.1.1 Dynamical importance

The goal of a dynamical importance problem is to find nodes that have an outsized influence on the outcome of some spreading process on a network. Here, we will consider that contagion is governed by the independent cascade (IC) model [61, 29, 38]. Recall from Chapter 1 that IC is a simple spreading process in which a cascade spreads along an edge connecting active node i and inactive node j with probability p_{ij} (this outcome is determined independently for all edges). Each edge is only given a single chance of spreading the contagion, and this stochastic process stops once no new nodes are infected. It has been shown that these dynamics map to the static problem of edge percolation for both continuous [62] and discrete time. Hence, this model applies to a wide range of applications. For simplicity, we consider only the case when all edges have an equal probability of spreading the contagion, i.e., $p_{ij} = p$ for all edges (thus yielding a bond-percolation process [25]), but all the methods presented here can be easily extended to the general case.

With the spreading dynamics fixed, we focus on quantifying how a set of nodes is deemed important [34]. We do this by considering sets of nodes as intervention targets and associating types of interventions with different notions of importance.

For influence maximization, we activate the nodes in S at the beginning of the cascade. A set's quality is then given by the expected average outbreak size

$$\langle I(S)\rangle = \sum_{i=1}^{n} I_i(S), \tag{2.1}$$

where $I_i(S)$ is the marginal probability that node i becomes infected when the contagion is seeded with S. A set is deemed more important when it leads to larger outbreaks, and influence maximization is thus the problem of finding

$$\hat{S} = \underset{S\in\mathcal{S}_k}{\operatorname{argmax}}\, \langle I(S)\rangle, \tag{2.2}$$

where $\mathcal{S}_k$ is the space of all possible sets of k nodes in a network.

For targeted vaccination, we immunize the nodes in S, meaning that a cascade can never become activated. We also use the average outbreak of Eq. (2.1) to quantify a set's quality, but now compute the marginal infection probabilities $\{I_i\}_{i=1,\ldots,n}$ by averaging over random initialization of the cascade at a single node. The goal of targeted vaccination is then to find

$$\hat{S} = \underset{S\in\mathcal{S}_k}{\operatorname{argmin}}\, \langle I(S)\rangle, \tag{2.3}$$

i.e., the set of nodes that most limit spread when immunized.

Finally, for sentinel surveillance, we simply observe the nodes in S and our goal is to learn of a cascade as rapidly as possible. We define the expected time to infection $\langle t_i\rangle$ of node i as the average number of time steps before node i becomes infected when a cascade is initialized at a node chosen uniformly at random from all the set of all nodes. (When computing this average, we attribute a time to infection of $t_{\max}^{\ell}$, the length of a particular stochastic realization ℓ of the cascade, to nodes that never get activated [34].) To evaluate the quality of a set, we then define the minimum expected infection time of S as

$$t(S) = \min_{i \in S} \langle t_i \rangle \tag{2.4}$$

and define sentinel surveillance as the problem of finding

$$\hat{S} = \operatorname*{argmin}_{S \in \mathcal{S}_k} t(S), \tag{2.5}$$

i.e., the set of nodes that will learn about a cascade the earliest, on average.

2.1.2 Choosing optimal sets

Finding the best solution among the space of all possible sets of a fixed size k is a challenging optimization problem. The design space grows as $\binom{n}{k} \sim n^k$ with the number of nodes n, which is exponential in network size when calculated in bits, and the problem is known to be NP-hard in the case of influence maximization [38] with IC. We thus employ a greedy method for all three problems to approximate the optimal sets. For the problem of influence maximization, this method provides a $1 - 1/e$–approximation to the optimal solution since the objective function (2.1) is submodular [63, 38]. While no such guarantee exists for vaccination or sentinel surveillance, we use the greedy solution for consistency of the comparison.

The basic greedy algorithm works by iteratively building the intervention set. We begin by evaluating the quality of each node individually, yielding the best set of size $k = 1$, which we label S_1. Formally, we proceed by adding the node that most improves the quality of the existing set as

$$S_k = \underset{i \notin S_{k-1}}{\text{argmax}} f(S_{k-1} \cup \{i\}). \tag{2.6}$$

where $f : \mathcal{S} \to \mathbb{R}$ is the relevant objective (e.g., outbreak size for influence maximization, and -1 times the outbreak size for targeted vaccination). Each stage of the greedy algorithm requires testing the quality of each of the $n - (k - 1)$ sets, which we do by averaging over T realizations of the spread process, as described above. If we require T realizations per set, each stage of the greedy algorithm requires $\approx \mathcal{O}(nT)$ simulations. (We assume a set of constant size $k = O(1)$ with respect to problem size n such that we need to inspect nearly all n nodes as candidates at every round of the greedy algorithm.) The entire algorithm thus requires $\mathcal{O}(nTk)$ simulations. Here, we use $T = 10^5$ for influence maximization and $T = 10^3$ for targeted vaccination and sentinel surveillance. (But for the latter two, we also average over seeds, e.g., over $n = 362$ starting location for the network of Facebook users shown in Fig. 2.1, thus leading to a similar number of simulations)

2.1.3 Efficient Monte Carlo

Computing the averages in Eqs. (2.1) and (2.4) is the major bottleneck for practical importance problems at scale, and some optimization steps are worth taking. Hence, we modify a fast algorithm originally developed to simulate bond percolation [60] to all three importance problems. Our method takes advantage of a well-known mapping between (1) the final outcome of the *dynamical* process described by the independent cascade model and (2) the static process described by bond percolation, in which a fraction p of a network's edges are chosen to be active [64]. Indeed, we can think of the edges used to transmit a cascade as the active edges of a bond percolation. As such, if we list the nodes reachable from at least one seed in a realization of bond percolation on a network, we have effectively run a Monte Carlo simulation of the IC model.

The mapping between IC and bond percolation does not provide us with a faster way to simulate IC in and of itself, but it allows us to leverage Newman and Ziff's fast percolation algorithm for IC [60]. This algorithm builds on the insight that a particular percolation instance can be augmented by a single extra edge to provide an additional data point—as in: another Monte Carlo simulation—*nearly* for free. This thus shifts the Monte Carlo paradigm from simulating IC many times, with each simulation costing $O(m)$ operations where m is the number of edges, to create a series of correlated percolation instances by adding one edge at a time (yielding m instances for roughly the same costs of $O(m)$ operations). All the bookkeeping necessary to make this approach work can be made efficient with a union-find (also known as disjoint-set) data structure, which is designed to maintain a collection of growing and merging sets while allowing for efficient retrieval [65].

The algorithm goes as follows [60]. First, we start with all edges removed (inactive) from the network. Each node is initially assigned a unique label to track its cluster, signifying that each node is in its own cluster. We then add each edge to the network, one at a time and in random order, using the union-find data structure to keep track of which node is in which cluster. A *union* operation is performed each time a newly added edge joins to nodes that were previously part of different clusters. We continue the algorithm until all m edges have been added to the network. We repeat the whole process a number T times, yielding a total of mT percolation instances, m for each sweep of the algorithm.

When it comes time to compute averages over the instances (e.g., of the outbreak size), we need to correct for the fact not all of these samples are equally likely under bond percolation or IC. Indeed, for any particular instance of percolation, the probability it has exactly $x \leq m$ edges follows a binomial distribution of mass

$$\Pr(m = x|p) = \binom{m}{x} p^x (1-p)^{m-x} \tag{2.7}$$

We can thus compute averages of functions $f(\cdot)$ of percolation realizations by taking the expectation

$$\langle f(S) \rangle \approx \sum_{t=1}^{T} \sum_{x=1}^{m} f(\Omega_{xt}) \Pr(m = x|p), \tag{2.8}$$

where Ω_{xt} is a percolation instance with x edges at the t^{th} sweep of the algorithm. We note that our samples are correlated since edges are added sequentially. However, each sweep is drawn independently and identically from this process because we add each edge in a randomly chosen order. Hence, when we run many sweeps, correlations wash out, and the overall effect is an improved efficiency for calculating the average $\langle f(s) \rangle$.

This simulation strategy works to evaluate Eqs. (2.1) and (2.4) directly in the inner loop of a greedy maximization algorithm, but it turns out we can solve importance problems with a greedy search more efficiently if we mesh Monte Carlo simulations and union-find more tightly.

For influence maximization, we note that the cluster membership of each node and the size of these clusters can both be queried cheaply from the union-find data structure and that computing the total outbreak size for a seed set merely requires summing the size of all clusters that include at least one of the seed. Hence, we can compute the change in average quality $f(S_{k'-1} \cup \{i\})$ of all $n - (k' - 1)$ possible updates to the greedy solution simultaneously as we run a sweep, with the quality of given seed set changing only when a seed in involved in a merge.

We can use a similar technique for the targeted vaccination problem, though we now need to account for the fact that each instance of the spreading process requires choosing a node as the seed. As we run a sweep, we now track the average outbreak size starting from all $n - (k' - 1)$ seeds (vaccinated nodes cannot be seeds) and simply forbid the addition of vaccinated nodes to clusters. Aside from the minimal overhead required to save and retrieve the outbreak size for each seed, this is a nearly $\mathcal{O}(n)$ speed-up.

The application of these ideas to sentinel surveillance is less straightforward because the union-find algorithm does not explicitly account for the time in which a node is infected, but rather *whether* a node will be infected when the infection is seeded at a particular seed. We solve this problem by maintaining an additional data structure corresponding to the actual infection tree that occurred on the network, assuming the initial seed was node i. Lengths of shortest paths in this tree correspond to infection times in a particular realization of IC. Once we have the tree, we can test all surveillance sets easily, taking the minimum infection time of each node in the set. To maintain the tree, we initialize it with an infection time of ∞ for all nodes except the seed i, which has infection time $t = 0$. As we add edges to the network, we check whether the shortest path to the seed node i has improved through the addition of this edge. (This can be done by checking the infection time of the node on either end of the new edge, which should be equal or differ by at most one.) If so, we propagate this information through the infection tree until the infection times are consistent, which we do recursively by going through a stack of pairs of connected nodes with inconsistent infection times.[1]

2.2 Results

Given a network, our general strategy for testing the impact of misspecification on intervention efficacy will be to (1) construct an optimal intervention $\hat{S}$ with a greedy approximation to Eqs (2.2),(2.3) or (2.5) with an assumed transmission probability p, and (2) check whether the intervention $\hat{S}$ can be transferred to a different problem, i.e., a nodal importance problem defined by a mismatched optimization target or the wrong transmission probability p'.

For the purpose of this Chapter, we will focus our analysis on the small anonymized ego network of 362 Facebook users shown in Figure 2.1, whose nodes are profiles and edges are Facebook friendships [47]. Preliminary results show that our results generalize to networks of an entirely different kind, such as infrastructure networks (e.g., the Internet at the autonomous system level) and transportation networks (e.g., highly connected airports).

[1] This update is typically cheap, though it is possible to construct adversarial updates that need to be propagated to as many as $O(n)$ nodes, say when a new edge transforms a path of n nodes whose end node is the seed into a cycle, in which case a full half of all nodes will receive a new infection time.

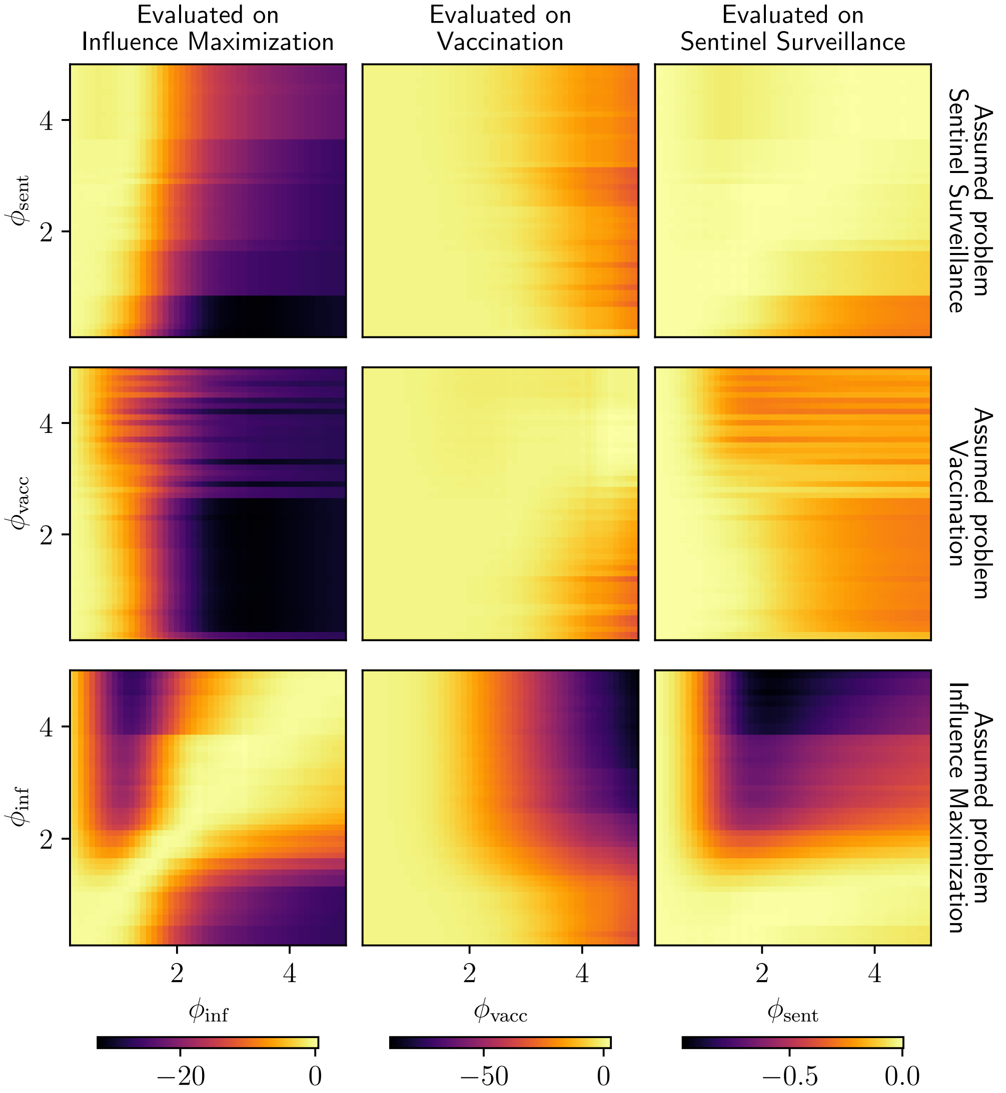

Figure 2.2: Evaluating quality loss on all pairs of problem instances. The quality loss (Eq. 2.9) measures the degree to which assuming the wrong problem during the optimization results in a loss of quality during evaluation; this quantity is asymmetric by construction. On the horizontal axis, we show the true importance problem instance (maximization objective and transmission rate), while the horizontal axis represents the importance problem we incorrectly assumed during evaluation. The plots on the diagonal (bottom-left to top-right) show comparisons between problem instances within the same importance problem but with varying incorrect transmission rates assumed (except on the very center of that diagonal, where problems are correctly matched). The off-diagonal plots make comparisons between problem instances with varying rates when the importance problem has been misspecified. For example, consider the top-left heatmap, where the true importance problem is influence maximization. If S^*_{sent} is the optimal set for the problem of sentinel surveillance at relative infection rate ϕ_{sent} and S^*_{inf} is the optimal set at ϕ_{inf}, the heatmap shows the value $f_{\text{inf}}(S^*_{\text{sent}}) - f_{\text{inf}}(S^*_{\text{inf}})$. These results are computed using the network of Facebook users shown in Figure 2.1.

2.2.1 Robust outcome at the edge of criticality

Our first set of results aims to illuminate the *quality* of interventions constructed with incorrect assumptions. More precisely, we want to quantify the extent to which the optimal solution for original problem instance A (defined as an objective function and transmission rate) is better than the optimal set for incorrect problem instance B, with respect to the objective associated with problem instance A. If we let S_A^* and S_B^* be the optimal sets for each problem instance, the quality loss associated with misspecifying the importance problem is the asymmetric quantity

$$\Delta f_{AB} = f_A(S_A^*) - f_B(S_B^*), \tag{2.9}$$

where $f(\cdot)$ is: the outbreak size (influence maximization), minus one times the outbreak size (targeted vaccination), and the minimum time to infection counted (sentinel surveillance). This metric is not only useful for quantifying performance loss across importance problems but also for evaluating the cost associated with misspecifying the correct dynamical parameter for a single importance problem.

Figure 2.2 shows our results, the quality loss Δf_{AB} for all possible combinations of problems and transmission rate. We use a budget of $k = \lfloor\sqrt{n}\rfloor = 19$ nodes for this analysis. We measure this transmission rate as a multiple ϕ of the critical threshold $p_c \approx 0.043$ in the absence of intervention.[2]

Focusing first on the diagonal, where problems are compared to themselves, we see that we consistently find $\Delta f_{AB} \approx 0$, indicating unsurprisingly that optimizing

[2] There is arguably no critical threshold in finite systems, so we use the susceptibility of the percolation process to detect a finite-size analog [66].

under correct assumptions is the best thing to do. In fact, there is a relatively generous region around this diagonal where the quality loss is nearly 0, indicating that getting the transmission rate wrong is not dramatic if we're not off by much. That said, we also observe a roughly block-diagonal structure in the case of influence maximization, where assuming a sub- (super-) critical process leads to a dramatic reduction in performance.

This sub- versus super-critical distinction becomes key once we move off the diagonal and compare entirely different problems. For example, when we evaluate important sets on the influence maximization task, any sets of nodes found with a low assumed transmission rate perform well *if* the true transmission rate is also low—even if they are found for different purposes, such as sentinel surveillance. This is due to the fact that outbreak size is very small in this regime, such that nearly any intervention is likely to result in a similar outbreak. As such, the solution is not transferable to influence maximization with a much higher transmission rate.

Importantly, we find that the most influential nodes found with $\phi \approx 1$—i.e., when the contagion is at the proverbial edge of criticality—seem to work across all problems. This finding thus suggests a possible empirical strategy for practical nodal importance: simply find influential nodes with $\phi \approx 1$ and transfer the solution to vaccination or sentinel surveillance—a rare trifecta of objective satisfied by the same decision.

2.2.2 The composition of optimal choice sets

Having compared the quality of solutions, we now turn to the solutions themselves. Is there anything particular about the selected nodes? Are the sets of optimal nodes

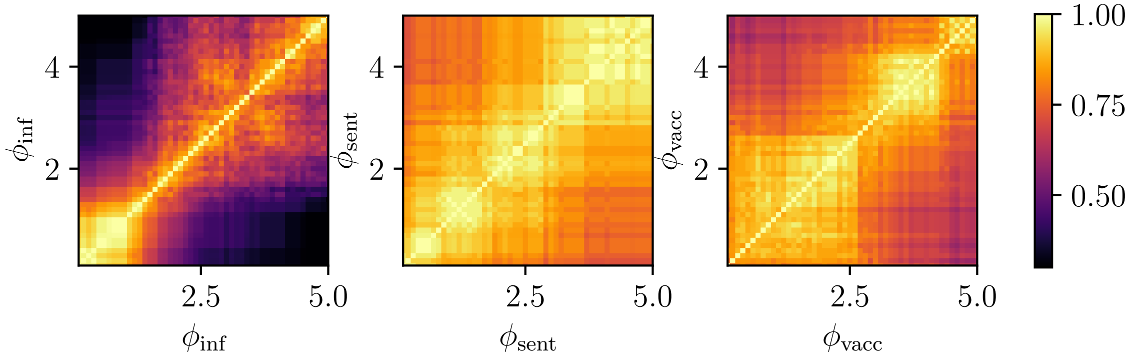

Figure 2.3: Similarity of optimal sets within an importance problem. Each plot shows the similarity (Eq. 2.10) of the optimal decision set for all three importance problems when this set is found for a relative transmission rate ϕ_x versus when found with a different transmission rate ϕ_y. The importance problems are: (a) influence maximization (b) sentinel surveillance, and (c) vaccination. These results are again computed on the network of Facebook users, shown in Figure 2.1.

similar at all?

As a means of capturing nuanced notions of similarity between sets of nodes (S, S'), we use a network-based metric based on minimum distances, defined as

$$D(S, S') = \frac{1}{k} \max_{\pi \in \mathcal{P}} \left[\sum_{v \in S} \frac{1}{1 + d(v, \pi(v; S'))} \right], \tag{2.10}$$

where $d(u, v)$ is the length of the shortest path between nodes u and v (defined as 0 if $u = v$ and as ∞ when there is no such path), and where $\pi(v; S')$ is a bijective mapping between every node $v \in S$ to a node in S' and $\mathcal{P}$ is the space of all such mapping. Conceptually, we're searching for a mapping π that minimizes the distance between all nodes and their image (and thus, that maximizes their similarity.) This metric is a special case of the assignment problem [65], which can be solved using the Hungarian algorithm in $\mathcal{O}(k^3)$ [67], and hence poses no computational challenge in

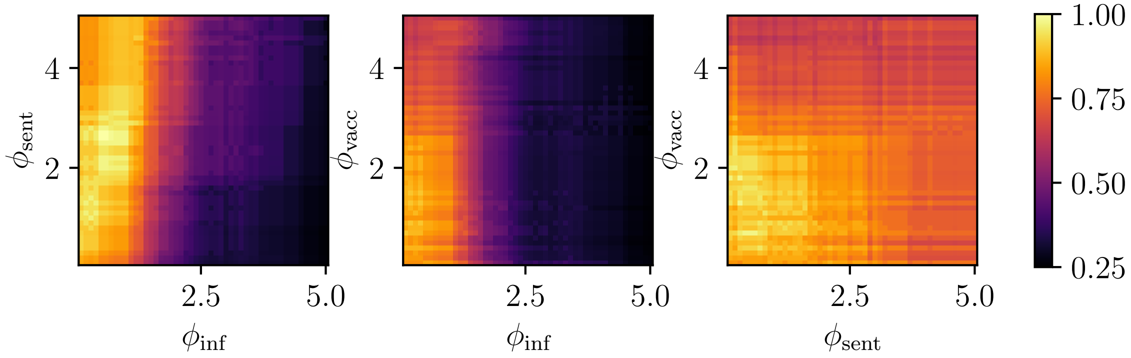

Figure 2.4: Similarity of optimal sets across importance problems. This figure shows essentially the same results as Fig. 2.3—the similarity of solutions to various importance problems—but compares solutions to different importance problems instead of comparing problems with itself. (a) Comparison of sentinel surveillance and influence maximization. (b) Comparison of targeted vaccination and influence maximization. (c) Comparison of vaccination and sentinel surveillance.

practice since k is typically small.

This metric has the nice property of being defined even if some terms involve nodes (u, v) that are not connected or that are paired with themselves. It also varies slowly enough to offer a more granular view of similarity (instead of acting as a $0-1$ loss like a Jaccard similarity, which does not distinguish between sets with no common nodes, regardless of their distance) Finally, it varies from $|S|$ when a set is compared to itself to 0 if all optimally matched nodes are in different components. [3]

Figures 2.3-2.4 show the similarity of the decision sets for all pairs of problems, as quantified Eq. (2.10). Unlike the performance loss discussed in Sec. 2.2.1, this quantity is symmetric, and we can thus use six heatmaps to summarize our findings. Figure 2.3 compares problems with themselves. Focusing on influence maximization

[3] We note that the metric imposes a somewhat arbitrary scale of what distance matters for the purposes of set comparison. One could imagine a version of the metric with a tunable exponent, i.e., $1/(1 + d(i, j)^{\alpha})$. Here, we use $\alpha = 1$ for simplicity.

first, we see that no optimal set persists for very long when we vary *phi*—the diagonal displays a narrow band of similar solutions. Echoing the results of Sec. 2.2.1, we also notice that the optimal set remains largely the same below the critical threshold and further that it bears no resemblance to the optimal solution above the critical threshold.

Interestingly, we see much less variation for the sentinel surveillance and vaccination problems: All optimal solutions are largely similar irrespective of ϕ. This result is consistent with the performance loss analysis of the previous section, in Fig. 2.2, which shows that these two problems are much less sensitive to transmission rate misspecification.

When comparing different problems, in Fig. 2.4, we see that sentinel surveillance and vaccination are largely the same (rightmost panel). In line with our previous results, we also see that the solution to influence maximization is similar to that of these two problems—when a subcritical regime is assumed for influence maximization.

Taken together, all our observations explain why optimizing for influence maximization at the edge of criticality might be a good strategy: Doing so gets us a solution to influence maximization that can work robustly across transmission rates for that problem. When we port this solution to the other problems, this solution remains good regardless of the assumed rate because they are not very sensitive to rates in the first place.

2.3 Conclusion

In this chapter, we have studied the impact of model misspecification on the efficacy of interventions on networks. Using a small Facebook social network as an example, we evaluated the performance loss when the problem was incorrectly specified during the design of targeted interventions. We found that influence maximization solutions optimized near criticality were robust across all objectives, providing an empirical strategy for real-world applications where model inputs may be uncertain. This computationally intensive analysis was enabled by adapting efficient Monte Carlo simulation techniques to important problems, a contribution in its own right.

Our results demonstrate that, despite inevitable uncertainty, interventions designed with theoretical tools like influence maximization can still prove useful when applied in practice. We organize avenues for future work in two categories: Empirical analyses and technical innovations.

On the empirical front, it would be worth expanding this analysis to a larger set of networks: Do the findings hold up across the spectrum of human behavior-driven networks? Do significant differences in budgets lead to different findings at all? And are there structural correlates that can predict nodal importance when it matters—at the edge of criticality?

On the technical side, it would be interesting to devise new estimators that formally take the type of robustness highlighted here into consideration [50, 68], perhaps using ideas around Bayesian decision theory. Developing Monte Carlo algorithms for generic IC models would also be a significant advance, unlocking the possibility of running more nuanced analyses of robustness (with rich transmission dynamics characterized by heterogeneous transmission rates.)

Finally, to facilitate the practical deployment of importance problems in scenarios where the bottleneck is network size rather than the large number of simulations needed to study robustness, it would be important to develop scalable methods to evaluate the objective itself. The next chapter addresses this open challenge.

Chapter 3

Message passing equations for probabilistic, temporal interventions in spreading dynamics on networks with loops

Designing good interventions in network dynamics requires evaluating how a large number of alternatives impact the outcomes of a system. For disease models, this outcome is the expected outbreak size, which we have thus far calculated using the sampling approximation of Eq. 1.6. This chapter introduces the method of message passing, which can be used to calculate the expected outbreak size directly, thus avoiding the sampling approximation altogether. Our algorithm can be used to test interventions on networks under a variety of scenarios, including those discussed in Chapter 2.

For example, our method enables us to test probabilistic scenarios such as partial immunity and temporal interventions such as vaccine rollouts or determining the average quality of a sentinel set. Furthermore, our method can be applied to a broader class of networks than standard message passing techniques.

We begin this chapter by reviewing the existing literature on message passing for network epidemiology. We then introduce a series of progressively more sophisticated passing methods that can be used for tree or tree-like networks, networks with many short loops, temporal dynamics, and finally network interventions of several types.

3.1 Message passing for network epidemiology

Message passing, as a technique, was first introduced by Bethe in the context of statistical physics in the 1930s [69] and more recently by Pearl in 1982 as a broadly applicable technique in computer science [70]. It has since been used for a variety of purposes in computer science, Bayesian inference, and statistical physics [71]. The method centers around causal relations between subsets of variables. In many cases, any single variable is directly caused by only a small number of other variables. For example, the probability node i is infected *directly* depends only on the infection probability of its neighbors, in the sense that the only mechanism for i to be infected is by one of its neighbors. Since these variables depend on each other in specific ways, the structure of these interactions forms a network. In this case, the physical interaction network defined on the set of nodes also represents the causal diagram for a set of random variables: the marginal probabilities of those nodes. When this network is a tree, message passing algorithms give exact answers, allowing us to calculate the average outcome of the system without resorting to Monte Carlo sampling.

Message passing has been applied to a variety of applications within network science such as percolation [72], community detection [73], spin glasses [74, 71], and others [75]. In the context of epidemiology, message passing has been extended to a dynamical context for general SIR models [76] and recurrent-state (SIS) models [77]. These dynamic message passing (DMP) algorithms accurately represent not only the final state of the system but also the state of the system over time. Message passing has also been applied to the problem of optimal vaccination [78, 46].

The traditional message passing framework, as defined in the examples above, is only exact on trees. In practice, networks that are tree-like—meaning the region around any given node is a tree—also produce very accurate message passing solutions. Moreover, even when the tree-like assumption is not true, as is the case for many empirical networks, the tree-like approximation still works surprisingly well in many cases [79]. However, the networks for which message passing fails most strikingly are those that contain many short loops. These loops are common in social systems, which tend to have a high degree of clustering. Recent methods have corrected this failure by accounting for short loops explicitly [80, 81]. The message passing on networks with loops (MPL) framework is the basis of our method, which additionally combines features from the vaccination framework of [**?**] and dynamic message passing [76, 77].

While we use message passing to test interventions, it can also be used for statistical inference. In the context of network epidemiology, message passing has been used to infer the origin of an epidemic [82] from data on the final infection status of each node. While we do not explore such applications here, the methods we develop here could be used to expand the applications of such an inference procedure to networks with more loops.

Our approach for this chapter will be as follows. We first introduce the message passing framework for the independent cascade model. We then modify the model to explicitly account for networks with loops. Finally, we show how the framework can be used to test interventions and validate those applications empirically.

3.2 An introduction to message passing equations

Designing a message passing algorithm begins with establishing the set of probabilities whose value we are interested in learning. In the context of network epidemiology, we want to calculate the expected fraction of infected individuals from a contagion that has spread via the independent cascade (IC) model, which we discussed in Sections 1.1.2 and 1.1.3. Recall that under this model, each edge has an independent probability p_{ij} of spreading the contagion. Let X_i be a boolean random variable that describes the infection status of node i at the end of a contagion process. All possible outcomes of the system can be represented as a joint outcome distribution $P(\mathbf{X}) = P(X_1, X_2, \ldots, X_n)$ over these node-level variables. From this distribution, we can calculate the fraction of infected individuals, which can be written as $I = \frac{1}{n}\sum_i X_i$, and its expected value as

$$\begin{aligned}\langle I\rangle &= \sum_{\mathbf{X}} \left(\frac{1}{n}\sum_i X_i\right) P(\mathbf{X}) \\ &= \frac{1}{n}\sum_i \sum_{\mathbf{X}} X_i P(\mathbf{X}) \qquad , \\ &= \frac{1}{n}\sum_i \pi_i\end{aligned} \tag{3.1}$$

where π_i is the marginal probability of infection for node i. The fact that our quantity of interest can be represented as a sum of marginal expectations is quite useful since it will be these marginals that we will calculate directly through message passing.

As a starting point, we observe that the marginal π_i for node i is simply a function of the same quantity for i's neighbors, which we denote ∂i. In other words, suppose we knew the probability that one of i's neighbors k would be infected with probability π_k. The probability that i becomes infected due to this node is $p_{ki}\pi_k$. Naively aggregating over all neighbors, the probability that i is infected is

$$\pi_i = 1 - \prod_{k\in\partial i}\left(1 - p_{ki}\pi_k\right), \tag{3.2}$$

which can be interpreted as the probability that at least one neighbor has infected i. This equation is a great start except for one problem: the values π_k are unknown. However, a similar equation can be written for any neighboring node π_k, assuming we knew the values of the marginals for each of its neighbors. The intuition behind message passing, then, is to write down a set of equations for each of the marginals π_i. These equations should be self-consistent, such that if we found the true marginal probabilities for each node, all the equations would satisfy each other. In practice, once we have this set of self-consistent equations, we can choose any convenient starting value and iterate the equations to convergence.

If we try this approach for Eq. 3.2, we run into several problems, which we will demonstrate with a simple example. Consider a graph with just two nodes that are connected by an edge. Let each of the nodes have an initial infection probability of $\frac{1}{2}$, the probability node 1 infects node 2 is $p_{12} = \frac{1}{4}$, and the probability node 2 infects node 1 is $p_{21} = \frac{1}{2}$.

Such a simple system can be solved analytically quite easily by enumerating all possibilities in the system. We begin by considering all possible outcomes for node 2 and the probability of these outcomes. Node 2 could become infected in one of two ways: it is either initially infected, or it is infected by node 1. The former happens, of course, with probability $\frac{1}{2}$. The latter happens with probability $\frac{1}{2} \times \frac{1}{2} \times \frac{1}{4}$, which is the probability that 2 was not infected initially times the probability node 1 is initially infected times the probability the transmission is successful. All together, the probability 2 is infected is $\frac{1}{2} + \left(\frac{1}{2}\right)^2 \frac{1}{4} = \frac{9}{16}$. By the same logic, the probability 1 is infected as $\frac{1}{2} + \left(\frac{1}{2}\right)^3 = \frac{5}{8}$.

Now, we can calculate these same probabilities using message passing. According to Eq.3.2, the message passing equations for the system are

$$\begin{aligned} \pi_1 &= p_{21}\pi_2 \\ \pi_2 &= p_{12}\pi_1. \end{aligned} \tag{3.3}$$

Setting $\pi_1 = \pi_2 = \frac{1}{2}$ initially, we run message passing to convergence and find that the self-consistent solution is $\pi_1 = \pi_2 = 0$, as shown in Figure 3.1b. Clearly, we have an error.

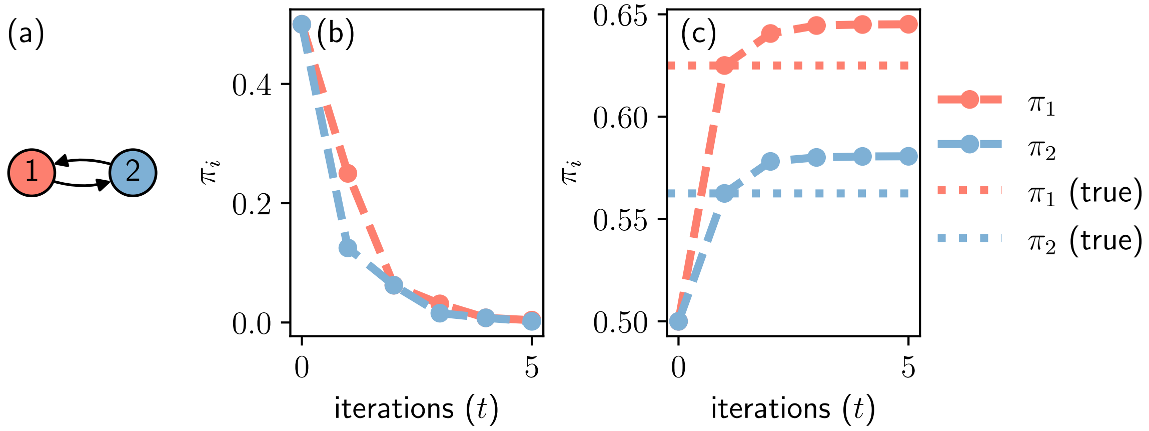

Figure 3.1: (a) A minimal two-node network on which message passing algorithms have been run. (b) Evolution of the message passing solution according to Eq. 3.2. (c) Evolution of the message passing solution according to Eq. 3.4.

One problem is that Eq. 3.2 assumes node i could only be infected in one way: through infection by one of its neighbors. In actuality, i could also be infected initially as a seed of the contagion. These two possibilities are mutually exclusive, which implies that Eq. 3.2 does not apply when i is initially infected (and hence could not be infected again). To fix this problem, let the quantity s_i be the probability that node i is initially infected and is the seed of a contagion. The updated message passing equation then becomes

$$\pi_i = s_i + (1 - s_i)\left[1 - \prod_{k \in \partial i} (1 - p_{ki}\pi_k)\right]. \tag{3.4}$$

Returning to our simple two-node graph, we set $s_1 = s_2 = \frac{1}{2}$. The message passing equations become

$$\pi_1 = \frac{1}{2}(1 + p_{21}\pi_2) \tag{3.5}$$

$$\pi_2 = \frac{1}{2}(1 + p_{12}\pi_1). \tag{3.6}$$

With $s_i > 0$, we now choose to set $\pi_1 = \pi_2 = 0$, initially. Testing the effect of these updated equations, we observe something closer to expected. After the first iteration of the algorithm (Figure 3.1c), both nodes gain additional probability due to the possibility of being infected by the other. The marginals of each node then continue to increase and converge to a self-consistent equilibrium, just as expected.

Why did we choose $\pi_i = 0$? We could have chosen any other value (aside from $\pi_1 = \pi_2 = 1$), and the equations would still have converged to the same value. The reason is that $\pi_i = 0$ results in marginals that represent the initial state of the system, before any infections across edges have taken place. This similarity is more than just a convenience. It turns out that, by setting the initial value of the system in such a way, the progression of the marginals as they converge to their equilibrium is exactly equivalent to temporal marginals of the discrete-time system. In other words, the value of π_i after t steps of message passing is exactly $\pi_i(t)$ in the real system. Taking advantage of this exact correspondence is known as dynamic message passing, which we discuss more explicitly in Section 3.5.

Consider the example of our two-node system, where $\pi_i(0) = \frac{1}{2}$ for both nodes, which exactly matches the message passing solution. After a single iteration of the algorithm, the marginals increase directly to their true final values, just as we expect in the real system. Very often, nodes are infected initially, but sometimes, they may also be infected by their neighbor, and the marginals $\pi_i(1)$ reflect this fact. In this simple two-node system, the contagion process lasts a maximum of one time step, since if one node infects the other, both are infected and there is nothing else that could happen.

In light of this dynamic interpretation, Eq. 3.4 has another problem. The marginals continue to equilibrate after 1 time step of the message passing algorithm, whereas the dynamic interpretation of message passing suggests that this should not happen. Once π_1 increases due to potential infection from node 2, node 2 now increases its own probability π_2 in response to this change. With respect to the true disease model, this additional update violates common sense. How could π_2 increase due to an increase in π_1, when node 2 was the cause of that increase in π_1 in the first place?

In terms of probability theory, the marginal π_i represents an aggregate of two mutually exclusive events, the probability that i was initially infected and the probability that i was infected by another. When we update the marginal for node 2, it should not cause a further increase in π_1 because the increase was due to a scenario for which π_2 has already accounted (node 2 was a seed). Because of this effect, we observe a feedback loop where both nodes overestimate their probabilities π_i. The extent of the overestimation is the sum of all the possible paths that are admitted by Eq. 3.4 but excluded in the actual system. For example, Eq. 3.4 allows for node 1 to be infected in the following ways:

1. initial infection ($\rightarrow 1$), with probability s

2. path $2 \rightarrow 1$, with probability p_{21}

3. path $1 \to 2 \to 1$, with probability $p_{12}p_{21}$

4. path $2 \to 1 \to 2 \to 1$, with probability $p_{21}p_{12}p_{12}$, and so on.

Any of these recurrent paths are incompatible with respect to the disease model and thus should be excluded. Describing these pathways through which π_1 provides some intuition to the name "message passing". The messages being passed around the network are probabilities. Each node sends a message to its neighbors, describing the probability it has been infected. The feedback loop described above is known as *backtracking*, where messages that have traversed an edge of the network then have an effect that traverses that same edge in the opposite direction. Because backtracking effectively creates marginal probability where it should not, we will try to mitigate its effects.

To adjust for backtracking, we would like the "message" that j receives from its neighbor i to contain only the probability that node i has been infected by some other source than j itself. We denote this quantity $\pi_{i\backslash j}$, and it will form the foundation of a successful message passing approach. Modifying the message passing equations of our two-node system one more time, we get

$$\pi_1 = \frac{1}{2}(1 + p_{21}\pi_{2\backslash 1}) \tag{3.7}$$

$$\pi_2 = \frac{1}{2}(1 + p_{12}\pi_{1\backslash 2}). \tag{3.8}$$

In the case of our two-node graph, $\pi_{2\backslash 1} = \pi_{1\backslash 2} = s$ because the only other way for either node to be infected is through being initially infected. In general, this will not be the case, and $\pi_{i\backslash j}$ is a quantity that needs to be updated dynamically. Nevertheless, we were able to calculate $\pi_{i\backslash j}$ easily for this small example, and Eq. 3.8 does produce the correct answer.

This simple example has demonstrated, in very rudimentary terms, all the elements of message passing that will be relevant. We first defined initial infection as an independent event that should be treated separately from infection by peers. Next, we briefly discussed the dynamic interpretation of the message passing equations, a subject to which we will return to later. Finally, we introduced a way of preventing messages from backtracking, helping us avoid overestimation of the marginal probabilities. Now, we turn to describing the standard message passing for tree (and tree-like) networks.

3.3 Message passing on trees

When nodes have more than one neighbor, we need a more general way of avoiding the over-estimation of message passing due to inadmissible infection pathways. The key to avoiding these pathways lies in the quantity $\pi_{i\setminus j}$ that we introduced in the previous section. This quantity represents the probability i has been infected by a node other than j and can be written as

$$\pi_{i\setminus j} = s_i + (1 - s_i)\left[1 - \prod_{k\in\partial i\setminus j} 1 - p_{ki}\pi_{k\setminus i}\right]. \tag{3.9}$$

We observe this equation is quite similar in form to Eq. 3.4, with the only difference being that we prevent backtracking by excluding the previous influences of j from the message i sends to j.

To show that this equation works, consider the tree in Figure 3.2a. We start by analyzing the messages passed between nodes 2 and 4. From the perspective of leaf

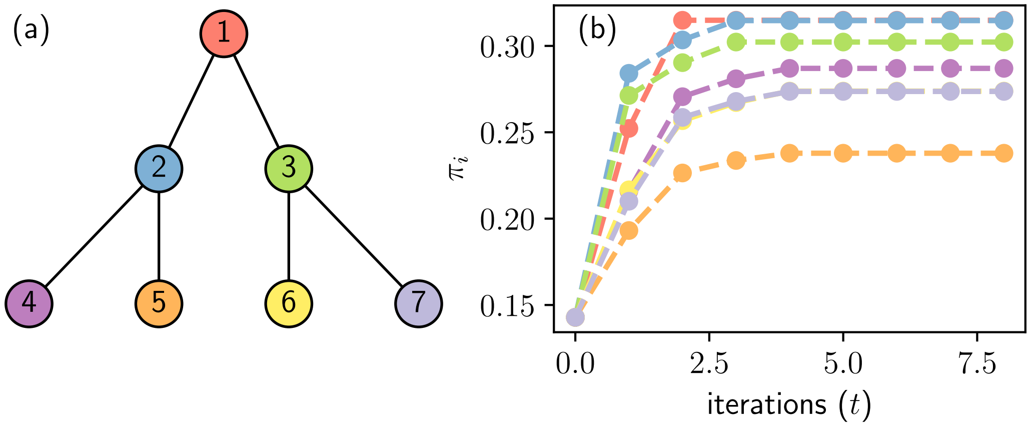

Figure 3.2: (a) A small tree for demonstrating message passing equations. (b) The marginal probabilities π_i for each node in the network, assuming $s_i = 1/n$ and the infection rates are slightly randomized, with $p_{ij} \sim \text{Beta}(10, 10)$.

node 4, the situation is nearly identical to the two-node network discussed earlier. Node 4 can be infected in two ways: either it was initially infected or was infected by node 2. The message that 4 sends to 2 is

$$\pi_{4\backslash 2} = s_4 + (1 - s_4)\left[1 - 1\right] = s_4 \tag{3.10}$$

which only contains the first possibility. We exclude the possible infection of 4, since we want to avoid counting the case where 2 infects 4, only to be infected again by 2. The scenario from the perspective of node 2 is only slightly different. It can be infected in four ways: it could either be initially infected or infected by one of nodes 2, 5, or 1. Three of these events have nothing to do with node 4, so the message node 2 sends to 4 will be the combined effect of each of them,

$$\pi_{2\backslash 4} = s_2 + (1 - s_2)\Big[1 - (1 - p_{52}\pi_{5\backslash 2})(1 - p_{12}\pi_{1\backslash 2})\Big]\,. \tag{3.11}$$

The sources that make up the message $\pi_{4\backslash 2}$ are shown in Figure 3.3d. The only excluded possibility is the probability that 4 has infected 2, since 2 would be returning the message that 4 already sent it.

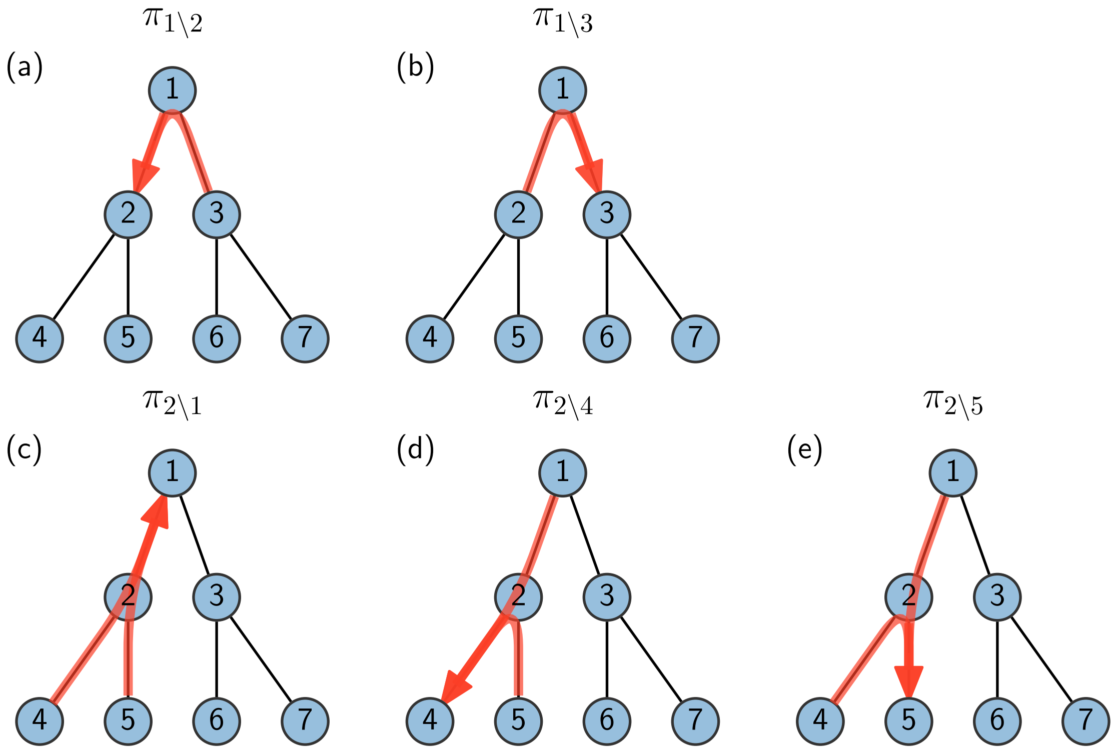

Figure 3.3: A visual explanation of the messages $\pi_{i\backslash j}$ passed on the tree in Figure 3.2. Each path represents a source of infection that i aggregates and passes to j. Each row shows all the messages passed by nodes 1 and 2, respectively.

Hence, node 2 will never over-estimate its probability of infection due to node 4 because 4 has no influence on any of the sources of infection it receives in the message $\pi_{2\backslash 4}$. By symmetry, the same arguments hold for nodes 5, 6, and 7.

We can tell a similar story for more central edges. Consider the messages passed between nodes 1 and 2. The message 2 passes to 1 is visualized in Figure 3.3c. It includes only infections due to nodes 4 and 5. For these messages to be part of an inadmissible path, they would have to be infected via node 1 by some other pathway than through 2.

Now that we have shown that Eq. 3.9 can generate self-consistent solutions for the quantities $\pi_{i\setminus j}$, it becomes straightforward to use them to calculate the true marginals π_i by aggregating all possible sources of infection at once:

$$\pi_i = s_i + (1 - s_i) \left[1 - \prod_{k \in \partial i} 1 - p_{ki} \pi_{k \setminus i} \right] . \tag{3.12}$$

This equation generalizes Eq. 3.8, for our simple two-node system. In that simple case, we could calculate the probabilities $\pi_{1\setminus 2}$ and $\pi_{2\setminus 1}$ directly since they depended only on the initial infection probability. A general system requires Eq. 3.9 to find a solution to the interdependent values of $\pi_{i\setminus j}$ first.

Importantly, Eqs. 3.9 and 3.12 both assume that the probabilities $\pi_{k\setminus i}$ are statistically independent. Looking at the tree network in Figure 3.2a, this condition is satisfied because for any node i's neighbors have no way of being infected by each other except through i. This feature is true of all tree networks. If an alternative path did exist for a neighbor j to be infected by i itself, then backtracking would not be able to account for this issue. Such an alternative path would form a loop in the network. In other words, this independence condition is satisfied if and only if the network has no loops, which is true of trees by definition.

In practice, most empirical networks are not trees, which may seem to doom the practical use of message passing. However, the message passing equations (Eq. 3.9 and Eq. 3.12) turn out to work quite well when networks are “locally tree-like” [79]. This property applies when the immediate neighborhoods around nodes are trees, a feature that implies the only loops in the network are usually quite long. Such long

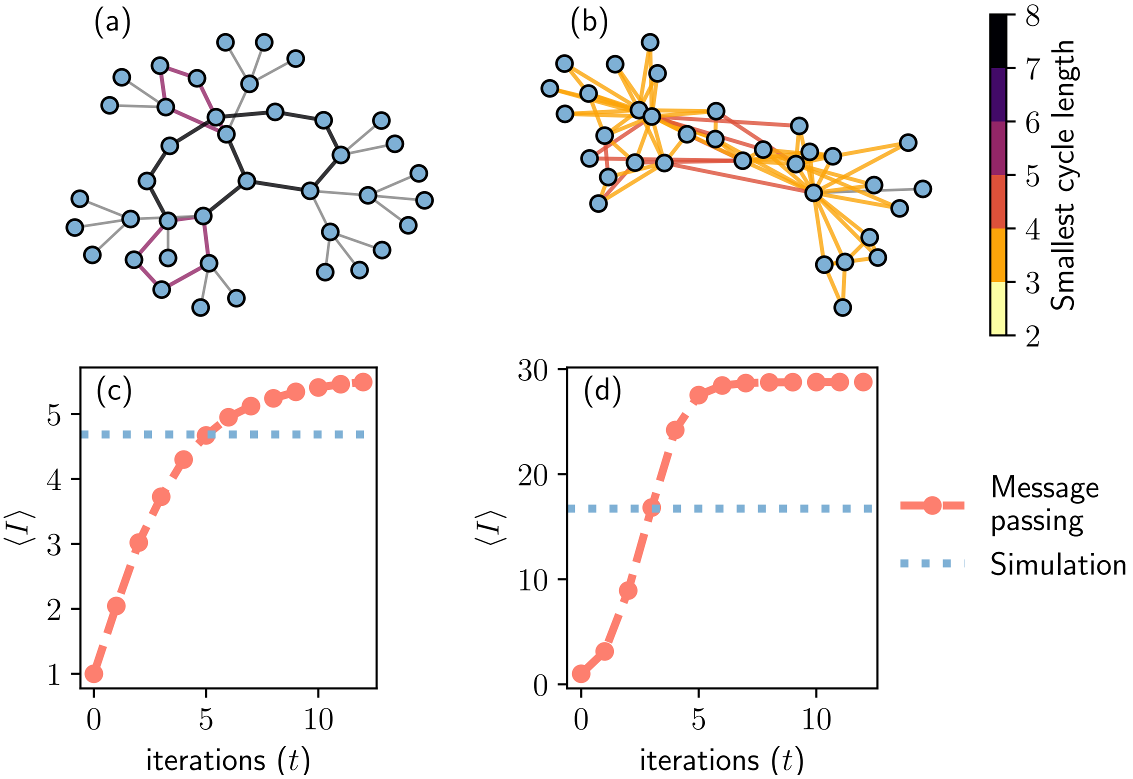

Figure 3.4: (a) A locally tree-like network and (b) Zachary's karate club [83], a network with many small loops. Each edge is colored according to the shortest path on which that edge appears.(c)-(d) A comparison between the message passing approach and the true solution, calculated via simulation. Each network uses $p = 0.5$ and seeds are chosen at random.

loops are a minimal issue for the message passing equations because the probability of a node being infected by a long path is small. If the probability of infection is p for each edge, the probability of a node being infected by a loop of length ℓ is p^ℓ, a quantity which decays exponentially. Figure 3.4 demonstrates the varying effect based on loop lengths by comparing two networks, one with many short loops and one with a few long ones.

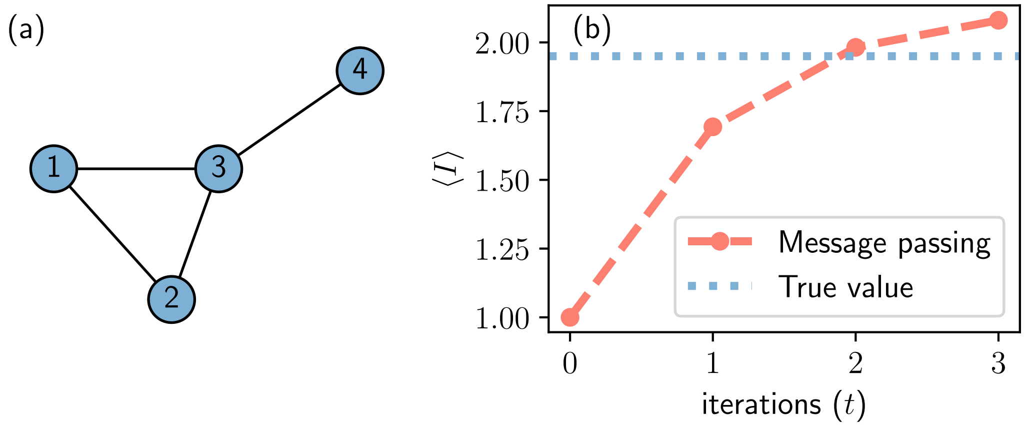

Figure 3.5: (a) A simple network with short loops. (b) The over-estimation of traditional message passing techniques.

3.4 Message passing on networks with loops

When short loops do appear in the network, the standard message passing equations are not sufficient. The problem is that Eq. 3.12 assumes that the neighbors of any node i are infected with independent probability. To demonstrate this independence, it will be helpful to instead consider the conditional probability that node i is *not* infected, $P(X_i = 0|\mathbf{X}_{\partial i})$, where $\mathbf{X}_{\partial i} = (X_{k_1}, X_{k_2}, \ldots)$ represents the states of the neighbors of i. Eq. 3.12 implicitly assumes that this probability can be factorized as

$$P(X_i = 0|\mathbf{X}_{\partial i}) = \prod_{k \in \partial i} P(X_i = 0|X_k). \tag{3.13}$$

When networks have loops, this assumption is not true, as we expect the random variables to have some conditional dependence.

As an example, consider the network in Figure 3.5a. Consider the marginal probability of infection for node 1

$$\pi_1 = s_1 + (1 - s_1)\left[1 - (1 - p\pi_{2\backslash 1})(1 - p\pi_{3\backslash 1})\right] \tag{3.14}$$

There are two important inputs to the marginal: $\pi_{2\backslash 1}$ and $\pi_{3\backslash 1}$. The message that 2 passes to 1 is the probability that 2 has been infected by a node other than 1, which we write

$$\pi_{2\backslash 1} = 1 - (1 - p\pi_{3\backslash 2}). \tag{3.15}$$

which implicitly depends on the marginal of node 3. However, the marginal π_1 already depends on node 3 via the message 3 sends directly to 1. We've now implicitly accounted for two infection pathways that occur through node 3. One that depends on the probability $\pi_{3\backslash 2}$ and another that depends on the probability $\pi_{3\backslash 1}$. If these probabilities were somehow independent, then we would have no problem with accounting for both pathways. Unfortunately, this is not true since both are conditionally dependent on $\pi_{4\backslash 3}$. These correlations cause Eq. 3.12 to overestimate the infection status of node 1.

We can also view this overestimation problem from a different lens. Working backwards from our expression for $\pi_{2\backslash 1}$, consider the message that 3 sends 2 is

$$\pi_{3\backslash 2} = 1 - (1 - p\pi_{1\backslash 3})(1 - p\pi_{4\backslash 3}). \tag{3.16}$$

Now we have a problem. Because $\pi_{2\backslash 1}$ implicitly depends on the message $\pi_{3\backslash 2}$, it also implicitly depends on itself through the probability $\pi_{1\backslash 3}$. In other words, the message passing equations account for an inadmissible infection pathway. More generally, the loop between nodes 1, 2, and 3 causes an infinite number of inadmissible infection pathways due to the endless feedback between the marginals: $\pi_{2\backslash 1}$ depends on $\pi_{3\backslash 2}$, which depends on $\pi_{1\backslash 3}$, which depends on the $\pi_{2\backslash 1}$, and so on and so forth. We can visualize the correlations in Figure 3.6. Like in our simple two-node system, failing to account for inadmissible pathways will cause message passing to overestimate the marginals π_i.

To resolve the problem, it would be preferable for 3 to instead send 2 the probability it has not been infected by either 2 itself or 1, which we write as $\pi_{3\backslash 1,2}$. This quantity suggests a way forward that might help us avoid the inadmissible infection pathways that the current message passing techniques fail to take into account.

We start by defining a neighborhood $\mathcal{N}_i$ as the edges in some region around node i. In this case, we choose the neighborhood around node 1 to be

$$\mathcal{N}_1 = \{(1,2),(1,3),(2,3)\} \tag{3.17}$$

Though a neighborhood is a set of edges, we will also occasionally refer to the set of nodes in the neighborhood as all those part of at least one edge. We will explain later why we choose these specific edges in $\mathcal{N}_i$, but first it is helpful to consider an example.

Our goal is to calculate the probability that node 1 is infected. We should return to the quantity $\pi_{3\backslash 1,2}$ and realize that it is equivalent to the probability that node 3 was infected in the absence of any edge in the neighborhood $\mathcal{N}_1$. As a step towards the general method, it will be more helpful to consider the quantity $\pi_{3\backslash \mathcal{N}_1}$. A similar quantity can be written for node 2 representing the probability it was infected except through nodes 1 or 3 as $\pi_{2\backslash \mathcal{N}_1}$. We note that the quantities $\pi_{2\backslash \mathcal{N}_1}$ and $\pi_{3\backslash \mathcal{N}_1}$ are

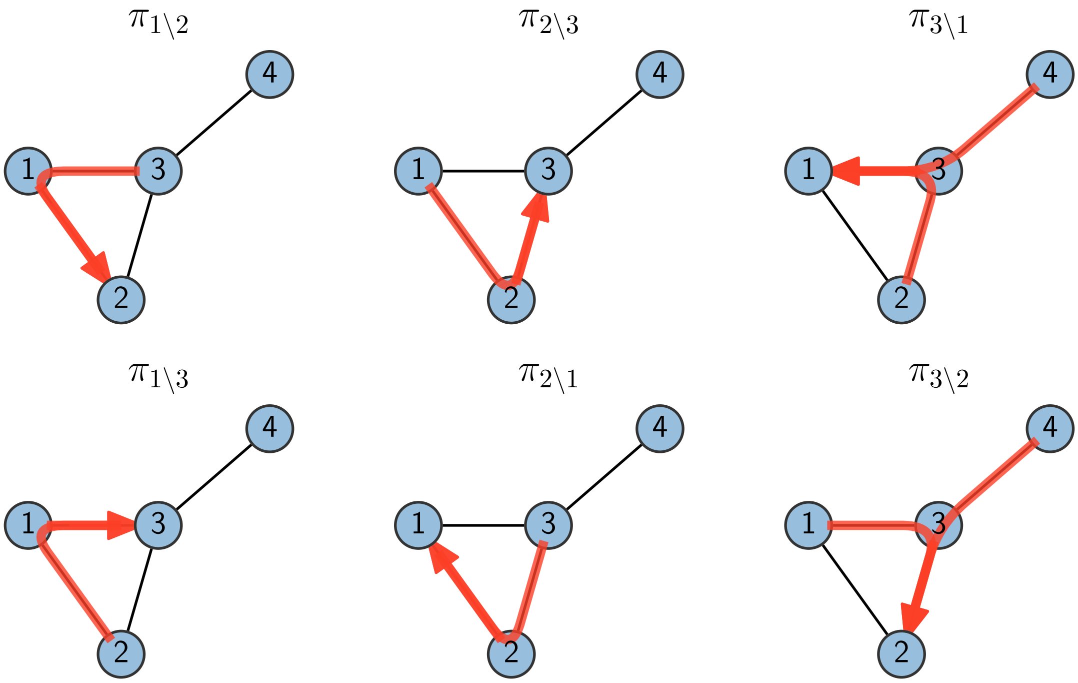

Figure 3.6: A visualization of the information for a sample of messages on a loopy network. The arrow shows the source of the message each node sends. For example, the message π_1 implicitly relies on the marginal $\pi_{3\backslash 1}$, so there is an arrow pointing from node 3 to node 2, where the message is being sent. Collectively, each row of messages produces a cycle through which nodes can effectively infect themselves.

Γ_1	Active edges (Γ_1)	$P(\Gamma_1)$	Reachable nodes from 1	$P(X_1 = 1 \vert \Gamma_1)$
1	$\{\emptyset\}$	$(1-p)^3$	None	0
2	$\{(1,2)\}$	$p(1-p)^2$	2	$\pi_{2\backslash\mathcal{N}_1}$
3	$\{(1,3)\}$	$p(1-p)^2$	3	$\pi_{3\backslash\mathcal{N}_1}$
4	$\{(2,3)\}$	$p(1-p)^2$	None	0
5	$\{(1,2),(2,3)\}$	$p^2(1-p)$	$2,3$	$1-(1-\pi_{2\backslash\mathcal{N}_1})(1-\pi_{3\backslash\mathcal{N}_1})$
6	$\{(1,3),(2,3)\}$	$p^2(1-p)$	$2,3$	$1-(1-\pi_{2\backslash\mathcal{N}_1})(1-\pi_{3\backslash\mathcal{N}_1})$
7	$\{(1,2),(1,3)\}$	$p^2(1-p)$	$2,3$	$1-(1-\pi_{2\backslash\mathcal{N}_1})(1-\pi_{3\backslash\mathcal{N}_1})$
8	$\{(1,2),(1,3),(2,3)\}$	p^3	$2,3$	$1-(1-\pi_{2\backslash\mathcal{N}_1})(1-\pi_{3\backslash\mathcal{N}_1})$

Table 3.1: All possible realizations of Γ for calculating the probability that 1 is infected via nodes 2 or 3.

independent, since nodes 2 and 3 are not connected by any path once the edges in $\mathcal{N}_1$ are removed. Together these probabilities are all we need to calculate the marginal π_1. Doing so, however, is more involved than applying a simple product, as in Eq. 3.9.

Instead, calculating π_1 will require considering all the possible ways in which 1 could be infected via nodes 2 and 3. Recall from Section 1.1.2 that the independent cascade model can me mapped to the static problem of bond percolation, in which edges in the network are randomly removed from the network with probability $1-p$. Let Γ_1 be a random variable whose support is the outcomes of a percolation process on the edges in the neighborhood $\mathcal{N}_1$. We list all possible outcomes of Γ_1 in the first two columns of Table 3.1. The table also lists probabilities of each outcome, which are simply $p^m(1-p)^{3-m}$, where m is the number of active edges in the instance of Γ_i.

Once we've enumerated the possible outcomes of Γ, we need to understand whether 1 would be infected in each scenario. Given a particular instance of Γ_1, let $N(\Gamma_1)$ be the reachable nodes from 1. If any of these reachable nodes are infected, node 1 will be infected with certainty. Therefore, the conditional probability that node 1 is infected is

$$P(X_1 = 1|\Gamma_1) = 1 - \prod_{j \in N(\Gamma_1)} (1 - \pi_{j\backslash\mathcal{N}_1}) \tag{3.18}$$

Since the quantities $\pi_{j\backslash\mathcal{N}_1}$ are independent, this equation makes no approximation. Finally, we can calculate the expected probability that node i becomes infected as the expected outcome over all outcomes of Γ_1.

Consider the case when edges $(1, 2)$ and $(2, 3)$ are both active, which we label $\Gamma_1^{(5)}$ (line 5 of Table 3.1). If either 2 or 3 becomes infected, we know—with certainty, because we know which edges are active—that 1 will become infected. The probability that at least one of these nodes are infected is

$$P(X_1 = 1|\Gamma_1 = \Gamma_1^{(5)}) = 1 - (1 - \pi_{2\backslash 1,3})(1 - \pi_{3\backslash 1,2}). \tag{3.19}$$

Finally, to calculate the total probability that 1 is infected due to the neighborhood $\mathcal{N}$, we can marginalize over all the various outcomes Γ_1, yielding

$$\begin{aligned} \pi_1 &= \sum_{\Gamma_1} P(X_1 = 1|\Gamma_1) P(\Gamma_1) \\ &= p(1-p)^2 \left[\pi_{2\backslash\mathcal{N}_1} + \pi_{3\backslash\mathcal{N}_1}\right] + p^2(3 - 2p)\left[1 - (1 - \pi_{2\backslash\mathcal{N}_1})(1 - \pi_{3\backslash\mathcal{N}_1})\right] \end{aligned} \tag{3.20}$$

Naturally, this result is quite different from what we would obtain by the independence assumption of Eq. 3.9. Now that we have considered an example, we will describe the approach formally in the next section.

3.4.1 General case

The general approach for message passing on networks with loops will be to define a neighborhood around node i, which we label $\mathcal{N}_i$. Within this neighborhood, we will take into account correlations in the marginals by explicitly calculating the paths through which i could become infected.

The assumption remains, however, that any message passed into the neighborhood $\mathcal{N}_i$, i.e. from some external source, will be completely independent. In general, this assumption will not be true, as it would be impractical to account for every loop in a network since some loops can be quite long. To make the neighborhood approach exact, we would need to establish extremely large neighborhoods around each node, and at that point, we might as well just do simulations on the entire network. Hence, the neighborhood approach will still be an approximation, in the same way that Eq. 3.9 is an approximation on tree-like networks. However, as we previously discussed (Figure 3.4), short loops have a greater impact on over-estimation than long ones. It will turn out that if we take into account short loops, we can improve quite significantly on the standard approach.

We now turn to defining what edges should be included in the neighborhood $\mathcal{N}_i$. Let a primitive cycle of length ℓ be a cycle that starts and ends at node i and that contains at least one edge not in any shorter primitive cycle. We define the neighborhood of node i to be all edges on any primitive cycle of length $\ell \leq r+2$. The parameter r represents the size of the neighborhood we construct around i. Setting $r = 0$ corresponds to the tree-like approximation since the only primitive cycles of length $\ell = 2$ are cycles that move to a neighboring node and immediately return.

Increasing r will include more edges (and possibly more nodes) in the neighborhood of i.

In the standard version of message passing, we wrote down a set of self-consistent equations (Eq. 3.9) for the quantities $\pi_{i\backslash j}$ and then modified the equation slightly as a means of calculating the marginals (Eq. 3.12). This time, we will proceed in reverse work in reverse, as it will be easiest to write down the equation for the marginals first and adjust it accordingly to obtain the quantities $\pi_{i\backslash \mathcal{N}_j}$.

As in the example above, we calculate the marginals in a two-step process. First, we calculate the probability that i becomes infected given a specific outcome Γ_i of the bond percolation process on $\mathcal{N}_i$. Then, we marginalize this conditional probability $P(X_i = 1|\Gamma_i)$ over all outcomes.

The conditional probability that i is infected given a particular outcome of Γ_i

$$P(X_i = 1|\Gamma_i) = s_i + (1 - s_i) \left[1 - \prod_{j \in N(\Gamma_i)} (1 - \pi_{i\backslash \mathcal{N}_i}) \right], \tag{3.21}$$

where $N(\Gamma_i)$ is the set of nodes reachable from node i under the configuration Γ_i. Eq. 3.21 effectively calculates the probability that at least one of the potential sources of infection $N(\Gamma_i)$ has infected i. Now, we can get the total infection probability for node i by marginalizing over these conditional probabilities, as

$$\pi_i = \sum_{\Gamma_i} P(X_i = 1|\Gamma_i) P(\Gamma_i). \tag{3.22}$$

To compute the quantities $\pi_{i\backslash \mathcal{N}_j}$, we don't have to change much in Eq. 3.22. We simply have to exclude from our counting any infection pathways that use edges appearing in $\mathcal{N}_j$. We represent this constrained neighborhood as $\mathcal{N}_{i\backslash j}$ and the constrained space of percolation outcomes on this neighborhood as $\Gamma_{i\backslash j}$. The conditional probability given a specific outcome is

$$P(X_i = 1|\Gamma_{i\backslash j}) = s_i + (1 - s_i)\left[1 - \prod_{k\in N(\Gamma_{i\backslash j})} (1 - \pi_{k\backslash \mathcal{N}_i})\right], \tag{3.23}$$

and marginalizing over the outcomes of $\Gamma_{i\backslash j}$, the final marginal becomes

$$\pi_{i\backslash \mathcal{N}_j} = \sum_{\Gamma_{i\backslash j}} P(X_i = 1|\Gamma_{i\backslash \mathcal{N}_j})P(\Gamma_{i\backslash j}). \tag{3.24}$$

Together, Eqs. 3.21 - 3.24 completely define the MPL framework. We now turn to some important details for implementing the method in practice.

3.4.2 Computation and algorithmic performance

Computing Eq. 3.24 requires marginalizing over all possible outcomes of the random variable $\Gamma_{i\backslash j}$. For a node i of degree k_i, even the $r = 0$ (tree-like) approximation, the number of possible outcomes of Γ_i is 2^k. As k increases to even a modest size, enumerating these possibilities becomes computationally infeasible, in the same way that calculating the probabilities for the system as a whole is computationally infeasible. Fortunately, we can rely on Monte Carlo sampling and a few computational tricks to overcome this obstacle. To make this sampling efficient, we leverage the same bond percolation algorithm used in Chapter 2 for this purpose [60].

Suppose we have T samples from the random variable $\Gamma_{i\backslash j}$, which we denote γ_z.

Then, we can approximate Eq. 3.24 in the following way:

$$\pi_{i\setminus\mathcal{N}_j} = \frac{1}{T}\sum_{x=1}^{T} P(X_i = 1|\gamma_x). \tag{3.25}$$

The above expression is the computational bottleneck of message passing with loops. At each step of the algorithm, we need to sample T times from $\Gamma_{i\setminus j}$ and then compute Eq. 3.23 for each sample γ_x. Fortunately, we can re-use the samples generated for each neighborhood, as the probability $P(\Gamma_{i\setminus j} = \gamma)$ depends only on the infection rate p. Using the same set of samples throughout the algorithm has other beneficial properties besides computational efficiency, one being that the marginals will increase monotonically throughout the algorithm. Otherwise, sampling would introduce fluctuations to their values as they reach convergence.

There are a few ways we can make the sampling over the neighborhood $\Gamma_{i\setminus j}$ more efficient. One source of performance improvement is to take advantage of structure in the neighborhood $\mathcal{N}_i$. Let $\mathcal{N}_i^-$ be the set of edges in $\mathcal{N}_i$ which are *not* connected to node i. For $r = 0$, $\mathcal{N}_i^-$ is the null set, since only edges connected i to its neighbors are present. The neighborhood $\mathcal{N}_i^-$ represents a subgraph of the network, which is comprised of one or more independent components. These components each infect i with independent probability, and can therefore be computed separately. Let $\mathcal{C}$ be an independent component of $\mathcal{N}_i^-$, and let $\Gamma_{i\setminus k}^{(\mathcal{C})}$ be a random variable that represents the outcome of a percolation process on the subset of edges present in $\mathcal{C}$. We can thus factor the outcome distribution of the percolation process on $\mathcal{N}_i$ as

$$P(\Gamma_{i\setminus j}) = P(\Gamma_{i\setminus k}^{(\mathcal{C}_1)})P(\Gamma_{i\setminus k}^{(\mathcal{C}_2)})\dots P(\Gamma_{i\setminus k}^{(\mathcal{C}_c)}), \tag{3.26}$$

where c is the number of independent components in $\mathcal{N}_i$.

A second computational advantage also concerns the neighborhood subset $\mathcal{N}_i^-$. For simplicity, we will assume that the neighborhood $\mathcal{N}_i$ is one single component. Consider a percolation process on $\mathcal{N}_i^-$, which we represent with the variable $\Gamma_{i\setminus j}^-$. We can compute the probability that i is infected given an outcome of this percolation process on a more constrained set of edges. For a particular outcome of this random variable, we will observe a series of independent components, which we will label $C^{(\Gamma)}$ to distinguish that these are components created via a specific instance of a percolation process on the edges $\mathcal{N}_i$.

By the same logic as before, we can calculate the probability that i is infected due to each of these independent components. The calculation involves two terms. The first is the probability that at least one node in the component $C^{(\Gamma)}$ becomes infected. If this occurs, all nodes will become infected with certainty, conditioned on the particular outcome of $\Gamma_{i\setminus j}^-$. The second term is the probability that i is connected to the cluster of active edges $C^{(\Gamma)}$ via an active edge. If there are $q(C^{(\Gamma)})$ of such edges, each of which has a probability p of being active, the second can be written $1-(1-p)^{q(C^{(\Gamma)})}$. Combining these two terms, we can calculate the probability that i is infected for a particular outcome of $\Gamma_{i\setminus j}^-$ and a particular component of active edges in that cluster as

$$P(X_i = 1|\Gamma_{i\setminus j}^-, C^{(\Gamma)}) = \left[1-(1-p)^{q(C^{(\Gamma)})}\right]\left[1-\prod_{k\in N(C^{(\Gamma)})}(1-\pi_{k\setminus\mathcal{N}_i})\right], \qquad (3.27)$$

where, in a slight abuse of notation, we let $N(C^{(\Gamma)})$ be the nodes present in component

$C^{(\Gamma)}$. Aggregating over all clusters, we get

$$P(X_i = 1|\Gamma_{i\backslash j}) = 1 - \prod_{C^{(\Gamma)}} \left(1 - P(X_i = 1|\Gamma^{-}_{i\backslash j}, C^{(\Gamma)})\right), \tag{3.28}$$

which represents the probability that X_i was infected due to at least one cluster in $\Gamma_{i\backslash j}$. By defining the neighborhood in this way, we reduced the size of the Monte Carlo sampling space to the size of $\mathcal{N}_i^-$.

Another computational concern is how to set the values of r and T. These hyperparameters govern how much complexity we add to the message passing algorithm. For $r = 0$, the message passing equations are the most straightforward, as they are the same as the standard approach. For large values of r, we may be performing Monte Carlo simulations for very large neighborhoods that may even span the entire network. Setting r this high would be wasteful since we would be sampling over sets of neighborhoods that overlap considerably.

In practice, the sampling accuracy is quite good after surprisingly few number of samples T, regardless of the value of r, as shown in Figure 3.7. By contrast, the value of r is more significant to performance and therefore to the computation time in practice. As we will see later, the performance of MPL improves each successive value of r.

3.5 Dynamic message passing

Up to this point, we have used message passing equations to calculate the node marginals π_i, where π_i represents the probability i is infected at the end of the process. As we briefly claimed in Section 3.2, by choosing initial values for the marginals π_i

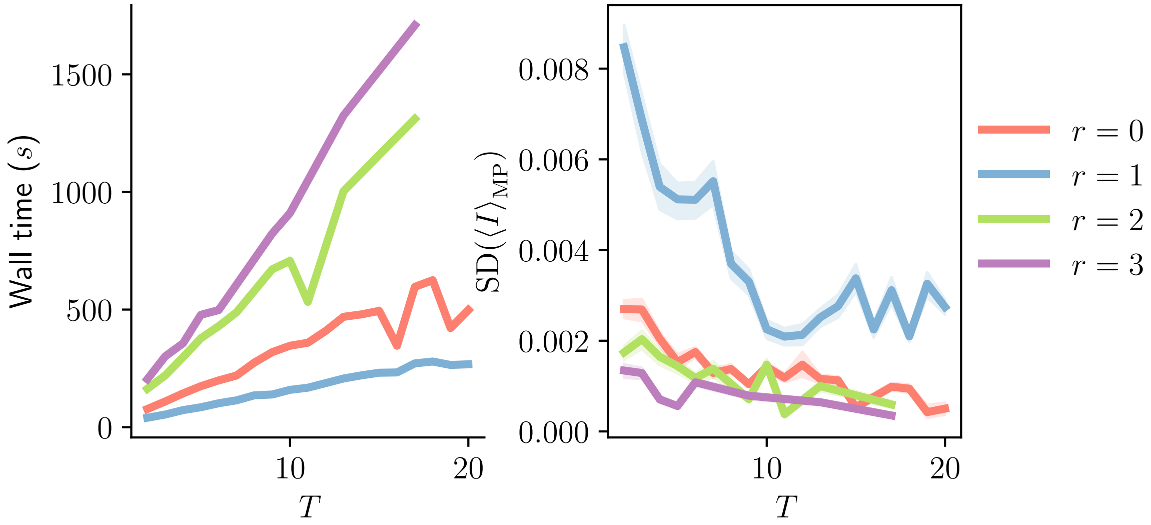

Figure 3.7: Exploring the space of neighborhood depth r and sample complexity T, computed where $\phi = 1.7$. (a) Computation time for MPL over for a range of r and T values. (b) The standard deviation (SD) of the expected outbreak sizes for $N = 10$ independent runs of MPL at each r and T configuration. In practice, a very small number of samples is sufficient to achieve accurate results.

that correspond to the initial state of the system, the value of π_i after t steps of the message passing algorithm is exactly the temporal marginal $\pi_i(t)$, the probability that node i has been infected after t discrete-time steps of the independent cascade model. The purpose of this section is to justify that claim.

To demonstrate that dynamic message passing works, we consider the example of a simple line network, as shown in Figure 3.8. Consider the case when node 1 is the only node initially infected. The only way for node 4 to be infected is through the path $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$. This infection, if it does occur, must happen at exactly three time steps, one for each of the three edges in the path. There is no other path that allows 5 to be infected. A similar argument can be made for any other starting node.

If, instead, we allow each node to be a seed with probability $s_i = 1/5$ to be the seed, then more options are possible. Node 4 can now be infected by a number of sources. If it is infected, it's most likely to be infected by an immediate neighbor. An infection from further away, e.g. node 1, carries less probability because each edge in the chain must be active for the node to be infected. We see this effect in Figure 3.8(f), where the marginal probability increases at each time step due to the possibility that more potential infection pathways are possible. Furthermore, we observe that at $t = 3$, the marginal probability increases less than at time step 1, indicating that the marginal is less impacted by pathways that are further away.

Hence, to appropriately interpret message passing in a dynamic context, we should describe $\pi_i(t)$, the value of π_i at the t^{th} iteration of message passing, as the probability that node i has been infected by a path of length $\leq t$. As a consequence of this interpretation, we should observe that all marginals converge in exactly $t_{\max} = d$ time steps, where d is the longest path in the network. This rule, however, is not true when loops are involved. Recall that loops introduce infection pathways that may pass through node i more than once. For a network with loops, the number of pathways in the network is infinite, so technically speaking, the marginals $\pi_i(t)$ will continue increasing forever, though they will asymptotically approach some value due to the fact that long loops are extremely unlikely. The important point is that as the number of time steps increases, message passing includes more inadmissible are accounted for in the system. At $t = 1$, no inadmissible pathways are possible, so the tree-like message passing is exact on any network. With further time steps, errors are possible in loopy networks (Figure 3.4). We now turn to the temporal interpretation of message passing with loops, which requires a bit of careful consideration.

3.5.1 Dynamic message passing on networks with loops

For the loopy version of message passing, we sample many possible infection pathways that could occur in the neighborhood of i. For $r > 0$, these neighborhoods include pathways that of lengths greater than 1. Hence, we actually require an amendment to the original equations.

For a particular outcome of Γ_i, we need to account for all the possible pathways through which i could be infected. For each node $k \in N(\Gamma_i)$, i.e. the nodes reachable from i, let ℓ_k be the shortest path from node i to k using only edges that are active in Γ_i. Because we know exactly which edges are activated for a given Γ_i, we know with certainty that if k is infected at time $t - \ell_k$, then i will be infected at time t. The probability that i is infected under a particular Γ_i at time t is that at least one

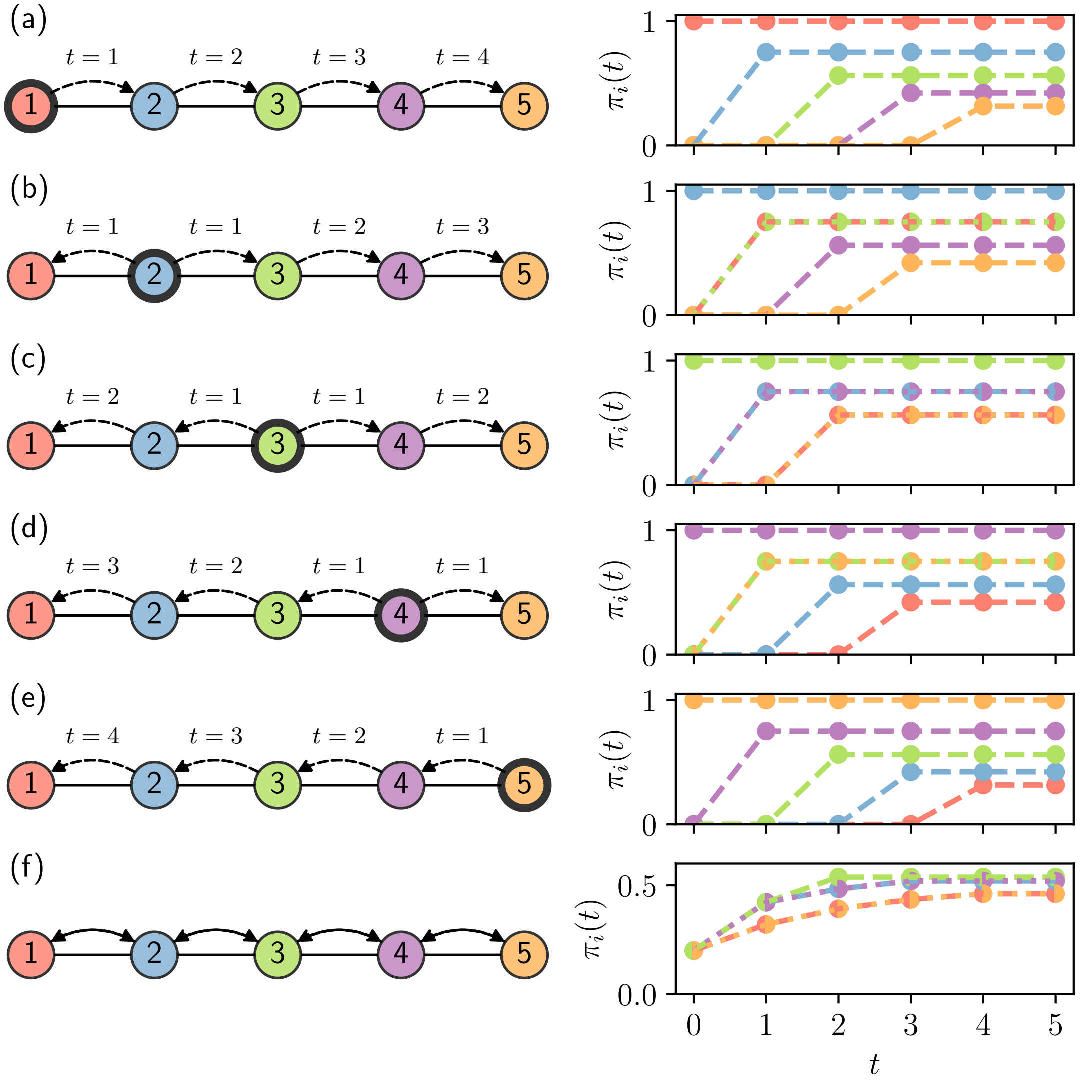

Figure 3.8: A dynamic interpretation of message passing on a simple line graph. (a-e) The evolution of node marginals $\pi_i(t)$ when the initial infection probability $s_i = 1$ for a single node (shown in bold) and zero otherwise. (f) The marginals for the entire system where $s_i = 1/n$ for all nodes.

of these events happens, i.e.,

$$P(X_i = 1|\Gamma_i)(t) = 1 - \prod_{k \in N(\Gamma_i)} \left(1 - \pi_{i \setminus \mathcal{N}_i}(t - \ell_k)\right). \tag{3.29}$$

With this, we now have a fully temporal framework message passing loops. What remains is to see how we can use these equations to test interventions such as the ones discussed in Chapter 2.

3.6 Interventions in the message passing framework

We can use the message passing framework to test any possible intervention that can be defined by fixing a random variable to a particular state. Interventions constitute altering the causal structure of a model. Under the discrete-time independent cascade model, there are a few parameters we can adjust. The first is the edge-independent infection probabilities p_{ij}. The second is the probability that nodes are initially infected s_i.

Any intervention in the system can be evaluated by fixing one or more of these variables. For example, if we want to test the outcome of a spreading process where a specific set of nodes is initially infected, we simply set $s_i = 1$ for all seed nodes and $s_i = 0$ otherwise. This is useful for testing interventions related to *influence maximization*.

For *vaccination*, our goal is to fix the system such that a vaccinated node should never be infected, meaning that we want to achieve $\pi_i(t) = 0$. To accomplish this, we set $s_i = 0$ for all vaccinated nodes, ensuring that vaccinated nodes are not initially infected. (For the remaining nodes should set $s_i = \frac{1}{n-k}$, where k is the number of vaccinated nodes.) To ensure vaccinated nodes remain uninfected for the course of the dynamics, we set $p_{ij} = 0$ for all edges connected to vaccinated nodes i, since i will never infect any of its neighbors. Adjusting both of these variables will effectively eliminate vaccinated nodes from the network.

We can also test partial immunity within this framework, and we have two ways of doing so. For example, we know that the COVID-19 vaccine is not completely effective, meaning vaccinated individuals may still be infected but with decreased probability. However, how should we model this reduction in probability? The question is still under investigation, and it does have consequences. It has been proposed that partial immunity could cause a random-targeting vaccination strategy to outperform one that targets high-risk individuals [84].

Modeling partial immunity requires us to know whether a vaccine works perfectly on a fraction of individuals or imperfectly for every individual. In other words, does a vaccinated node become infected via 5% of interactions that ordinarily would spread the disease, or is the vaccine completely ineffective for 5% of nodes that get immunized? Our framework allows both, but they must be specified differently. The former is specified by scaling the pairwise infection rates p_{ij} with $\sigma_i \sigma_j p_{ij}$, where σ_i is the fraction of breakthrough interactions for node i. Naturally, $\sigma_i = 1$ for non-vaccinated nodes, and the value can be variously adjusted to consider partial immunity. The second way to intervene is to suggest that node i has a probability v_i of being completely immune to the disease. To write this formally, let $V_i \sim \text{Bernoulli}(v_i)$ be a random variable that indicates whether the vaccination was successful. The marginal can be re-written as

$$\begin{aligned} \pi_i &= P(X_i = 1) \\ &= P(X_i = 1 | V_i = 0) P(V_i = 0) + P(X_i = 1 | V_i = 1) P(V_i = 1) \\ &= P(X_i = 1 | V_i = 0) P(V_i = 0) \\ &= P(X_i = 1 | V_i = 0)(1 - v_i) \end{aligned}, \tag{3.30}$$

where we used the fact that $P(X_i = 1|V_i = 1) = 0$ remove the second term. The core message passing equations (allowing for vaccination of both types) become

$$\pi_{i \setminus j} = (1 - v_i) \left\{ s_i + (1 - s_i) \left[1 - \prod_{k \in \partial i \setminus j} 1 - p_{ki} \sigma_i \sigma_j \pi_{k \setminus i} \right] \right\}. \tag{3.31}$$

The distinction between edge- and node-based immunity does make a difference, for example, when nodes have many connections. Even with substantial vaccine efficacy, only one of these connections must be successful for this node to be infected, mitigating the effect of vaccination. In such cases, it may be more worthwhile to vaccinate lower-degree nodes that may, for example, lie at critical junctures between communities. On the contrary, perfect vaccination for some individuals does not have the same drawback, and vaccinating high-degree nodes may be worth the risk.

Finally, we could also test temporal vaccination strategies. Let $v_i(t)$ be the probability that the vaccine was effective at time t. Suppose that a vaccine becomes available at $t = 10$, at which time we vaccinate a subset of nodes. This case amounts to setting $v_i(t > 10) = 1$ for vaccinated nodes. A similar approach can be done for intervening on interaction immunity by setting $\sigma_i(t)$.

Testing the quality of *sentinel surveillance* sets seems straightforward since sentinels do not actually impact the dynamics of the system. Therefore, it may seem we can simply run message passing once and then evaluate each potential sentinel set using the marginals $\pi_i(t)$. Recall that the quality of a sentinel set is the time at which the first node in the sentinel set becomes infected. We can calculate the probability that at least one sentinel has been infected by time t as

$$P(X_S = 1|t) = 1 - \prod_{s \in S}(1 - \pi_s(t)), \tag{3.32}$$

where X_S is a random variable that takes the value of 1 if at least one node in the sentinel set S is infected. Unfortunately, this equation (once again) improperly assumes that the temporal states of each sentinel are uncorrelated random variables. By contrast, we expect that two sentinels that are very close to each other in the network should have correlated infection probabilities. Failing to take into account such correlations could lead one to choose sentinels in the same area of the network when such an approach is less than optimal.

Fortunately, we can reinterpret our message passing equations to suit the problem of sentinel surveillance. Let $\pi_i^{(S)}$ be the probability that i has been infected by a path that does not contain a node in the sentinel set S. In other words, this quantity calculates the sources of infection that have not already passed through a sentinel node. Such a quantity is almost identical to the original marginal π_i in that it depends on its neighbors in the same way. The only modification is that sentinels should not spread the infection further, as this would violate the definition of $\pi_i^{(S)}$. The message passing equation for testing sentinel sets (on tree-like networks) is

$$\pi_{i\backslash j}^{(S)}(t) = 1 - \prod_{k \in \partial i \backslash j} (1 - p_{ki} \mathbf{1}[k \notin S] \pi_{k\backslash i}^{(S)}(t)), \tag{3.33}$$

where $\mathbf{1}[z] = 1$ if expression z is true and zero otherwise. Effectively, this intervention is the same as our approach to vaccination, where instead of preventing a vaccinated node from infecting others, we apply this property to sentinels.

This adjustment removes the possibility of errors in Eq. 3.32. It doesn't, however, deal with correlations due to network structure. Fortunately, we can leverage the MPL framework to modify the above equations in the same way as for the standard case. Now that we've defined how to implement interventions in these systems, we turn to how well they work in practice.

3.7 Empirical results

As a means of testing how interventions work, we test our message passing algorithm on a small friendship network ($n = 362$) from Facebook, the same one we used in Chapter 2. This network has a very high clustering coefficient, a metric that indicates the presence of many short loops. As such, it is a difficult challenge for any message passing algorithm. Despite taking into account short loops, the network has many medium and longer-length loops, which MPL does not take into account. Hence, we expect MPL to not correspond exactly to the simulation.

We test a selection of $N = 25$ intervention sets of size $k = \sqrt{n}$, $N = 20$ of which are chosen randomly. The remaining $N = 5$ sets are composed by selecting the top k nodes according to five structural centrality rankings: degree centrality, betweenness centrality, k-core centrality, closeness centrality, and eigenvector centrality [18]. We compute the expected outbreak size $\langle I \rangle$ after each node in the vaccination set has been vaccinated. As in Chapter 2, we calculate the simulated solution with the modified percolation algorithm we used in Chapter 2 [60].

The correspondence between these sets is shown in Figure 3.9. We observe that for all infection rates, the performance increases for higher values of r. The extent to which this happens depends on the infection rate $\phi = p/p_c$. As we discussed in 3.3, short loops pose a bigger problem for message passing than long ones because the probability of a node being infected by a loop of length ℓ is p^ℓ. This term also exposes a dependence on p. In particular, this term decays less quickly for larger infection rates, which explains the degrading performance as ϕ increases.

Furthermore, there appears to be a somewhat abrupt transition in the onset of this error. This transition occurs at the crossover from low infection rates, where the correspondence is extremely accurate, to higher infection rates, where message passing overestimates the expected outbreak size considerably. We show this abrupt transition in Figure 3.10. Furthermore, higher values of r can stave off this transition for higher infection rates than low values of r. The explanation for this abrupt change is that, around the critical threshold, message passing will fail to see correlations that exist, leading it to inflate the extent of infection in the network. to a supercritical expected outbreak size when the true value is still subcritical.

Despite this error, whether we might still be able to use message passing for testing interventions? It appears from Figure 3.9 that the rank-ordering of the nodes is well-preserved by message passing. If the ranking is preserved, we could still use it to choose the optimal intervention set from a range of choices.

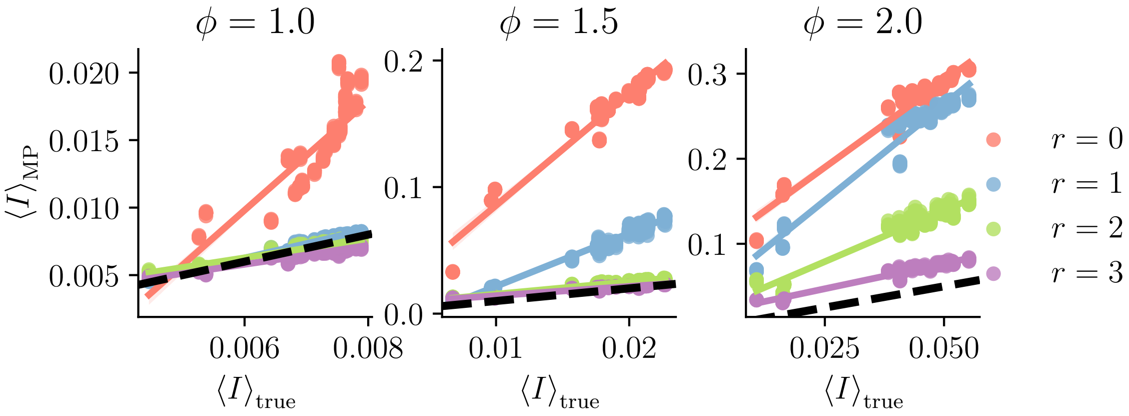

Figure 3.9: Comparing message passing with simulation for $N = 25$ vaccination sets. As the relative infection rate $\phi = p/p_c$ increases, message passing becomes more difficult. A larger neighborhood size staves off the error produced by message passing.

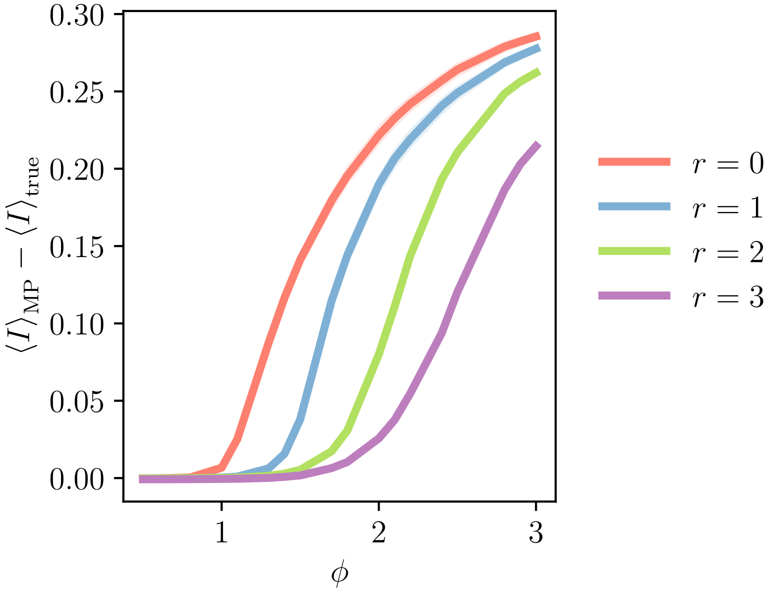

Figure 3.10: The overestimation of message passing due to loops in the network. We plot the as a function of $\phi = p/p_c$, the relative infection rate. Higher values of r decrease the overestimation, but increasing ϕ sufficiently will always impact the stones eventually.

To measure the rank correspondence between the two methods, we apply Spearman's rank correlation coefficient to the selection of intervention sets described above [85]. We treat the qualities calculated via message passing and simulation as two independent rankings. This metric is useful because it does not assume the relationship between two rankings is linear, which is the case for some of the fits shown in Figure 3.9.

The correlation is shown in Figure 3.11. While the rank preservation is reasonably good in all cases, we observe a noticeable dip in the correlation for some values of ϕ. This dip is largely due to a non-uniformity in which different intervention sets differ from the true solution. In other words, message passing exhibits greater error for some intervention sets than others. More work must be done to determine more precisely the source of this nonuniform error, but the abrupt transition we discussed in Figure 3.10. Essentially, the rank ordering gets jumbled around this threshold, where some sets are suddenly supercritical much sooner than others. Once message passing establishes all of these sets as supercritical, the rank-preservation increases once again. Understanding exactly what structural features lead to these nonuniform errors is a subject for future work.

3.8 Conclusion and future work

We proposed a message passing method for computing the temporal marginals for the independent cascade model on networks with many short loops. The method has the theoretical potential to be used to evaluate the quality of interventions, though it does not work perfectly for in all cases.

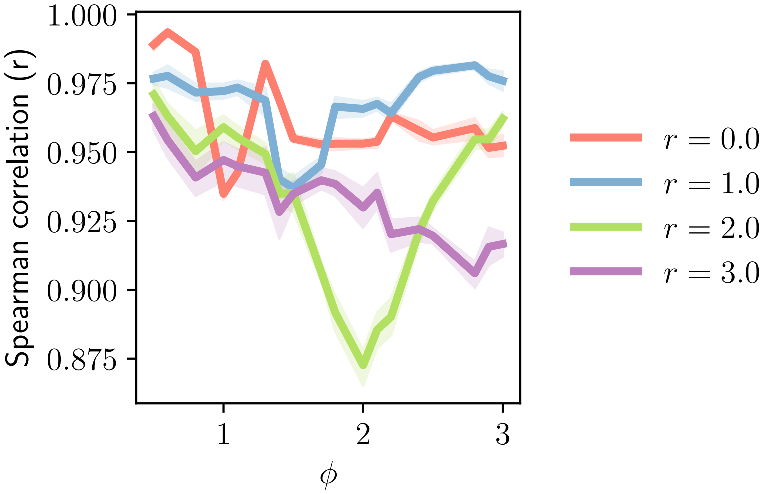

Figure 3.11: The rank correlation between message passing and simulation, compared using Spearman's rank correlation coefficient for different values of r

We found that, despite accounting for loops, message passing struggles with high infection rates, which seems inevitable for any approximate message passing algorithm. Furthermore, message passing seems to show a particular bias towards certain interventions in a way that does affect the node rankings. Previous work has suggested that this bias is due to the structural property of k-coreness [86], though more work is needed to understand this bias thoroughly.

Existing literature also suggests a few possible ways to further improve the performance of MPL. For example, previous work on message passing for non-treelike networks has focused on small motifs in the network, such as triangles [87] or fully connected cliques [74], that may appear in the network. These motifs can be treated easily analytically, offering a potential performance improvement. By definition, any node that is attached to a clique will contain all the edges of that clique in its neighborhood, assuming $r > 0$. Thus, accounting for cliques would not necessarily improve the estimation of MPL, but it may allow us to compute the probability $P(X_i = 1|\Gamma_i)$ more efficiently.

Nevertheless, this notion of finding highly connected regions of the network also suggests further work on hybrid message passing techniques. For example, there may be general ways of dividing the network into quasi-independent regions and nesting message passing algorithms within each other. Some work of this flavor has already been explored for exact percolation algorithms on small networks [88].

In spite of the challenges, message passing does preserve ranking reasonably well. It would be interesting to explore whether the performance improvements from message passing could allow us to expand a study like Chapter 2 to larger graphs.

Chapter 4

Conclusion

We began this thesis by recognizing that complex systems models have the ability to capture representative behavior of true policy systems but that the challenges of statistical inference with such models make them difficult to trust as an empirically backed representation of the world. This lack of trust is an essential criterion for a model to be confidently used for decisions with real-world impact. So, what tools would allow models to ascend from their role as intuition-builders to true decision-making tools? As a means of reflecting on future directions for robust decision-making, we discuss how modeling, statistical inference, and decision theory could come together to enable confident model-based decision-making. Our discussion will continue to focus on models of epidemiology, but the concepts apply more broadly.

We begin by discussing a Bayesian framework for handling uncertainty in model parameters. We also briefly discuss model selection and model comparison as tools for considering other kinds of model misspecification. Finally, we review the contributions as a whole and propose steps for future work.

4.1 A Bayesian framework for intervention design under uncertainty

In what follows, we consider the following pipeline for model-based decision-making. We start by defining a model $\mathcal{M}$ that is parameterized by a network A and some dynamical parameters θ. We assume the true world can be described well by some values of A and θ. Next, data must be collected on the system. We then use this data and the tools of Bayesian inference to infer the true parameters of the model, A and θ. Naturally, the inference procedure cannot determine the values of the true parameters exactly; instead, it will produce some uncertainty over their values. Finally, once an objective function is defined over the potential outcomes of the system, we use Bayesian decision theory to determine an optimal intervention in the system.

4.1.1 Handling uncertainty

In the framework of Bayesian statistics, we describe uncertainty over the parameters θ parameters mathematically using probability distributions. More specifically, we assume that each model parameter θ has a true value and that our lack of knowledge about that true value can be summarized as the distribution $P(\theta)$. Describing parameters as probability distributions is quite general. For example, the case of perfect certainty can be described with a probability distribution where all weight is placed on a single value. For discrete distributions, we say that $P(\theta = \theta^*) = 1$ for the true value θ^* and zero probability for all other values of θ. Furthermore, these probability distributions describe our uncertainty regardless of how that uncertainty is produced.

If we do not use any data to describe our belief, our uncertainty is known as a *prior probability distributions*. Prior distributions could describe other sources of knowledge about the parameter or, in cases where we know nothing at all about the parameter, place equal weight on every possible value that it may take. Once we used data to update our beliefs about the parameter, we refer to this distribution as a *posterior probability distribution*, e.g., $P(\theta|\text{data})$. The process of updating our beliefs is known as *Bayesian inference*, which comes from an application of Bayes' rule:

$$P(\theta|\text{data}) = \frac{P(\text{data}|\theta)P(\theta)}{P(\text{data})}. \tag{4.1}$$

The term $P(\text{data}|\theta)$, called the *likelihood*, describes a stochastic generative process that created the data. From this likelihood, we can use Bayesian inference to describe our posterior belief about the value of our parameter. This flexible approach to describing uncertainty is useful in contexts when modelers must collate various pieces of information about the system, whether it be expertise from domain practitioners or data from previous studies.

4.1.2 Uncertainty in network structure

Through the thesis, we have employed networks to describe the patterns of interaction between individuals in a population. We have defined this network by placing an edge between two nodes signifies the presence of an interaction between them. However, what qualifies as an interaction? Should two people co-author an academic paper together, follow each other on social media, or attend several of the same activities throughout the week? Any such notion of interaction might be valid in a particular problem context. This definition is formalized using a model to describe what data we'd expect to see with and without the presence of an edge. It is helpful to view the network not as a directly measurable feature of the system but rather as a mathematical object that must be constructed from data. This data is often noisy, leading to uncertainty with respect to the true network structure.

In the context of the spread of disease, what defines contact between two individuals? Should they be friends? What if they stood in the checkout line in the grocery store together? One promising source of data used for expressly these purposes in the COVID-19 pandemic has been detailed mobile phone location data. Using this data, we determine the presence of an edge between pairs of individuals that have been in the same location with each other. However, this definition is far from complete. We could imagine a variety of reasons why we might find two people who were not in close contact sharing the same location data. Perhaps they attended a coffee shop at the same time but sat on opposite sides of a large room. If our proximity criteria are too broad, we might detect an edge between these people where none exists. This scenario is known as a *false positive.* Alternatively, suppose someone's phone runs out of battery halfway through the day. We might miss the concert they attended that evening with hundreds of people. This is an example of an edge (or many) that should exist but doesn't, or a *false negative.* These sources of noise will produce errors in the data.

To formalize this noisy data-creation process, we'll start by representing the true network as an adjacency matrix A, defined as an $N \times N$ matrix with elements $A_{ij} = 1$ when an edge exists from i to j. We also have data, which we assume takes the form of an N matrix X, with elements X_{ij} equal to the number of observations suggesting an edge should be placed between i and j. In many cases, these elements will be one at maximum because repeated measurements of the system were not made. In other cases, such as with location data, we might have multiple observations by compiling data over time, e.g., if two nodes appeared in a coffee shop together and later arrived at the same restaurant. To link the true network to the data we have, we'll define a generative model $P(X|A)$, which describes the noisy process from which the data were created. This function is known as the *likelihood*.

Using the likelihood, our goal is to find the true network's adjacency matrix A. One possible approach could be to find the A that maximizes this likelihood function. Such an approach would return a single network–a point estimate–from which we can use to calculate one or more notions of dynamical importance described above. However, choosing a point estimate does not capture our *uncertainty* about our inference procedure. Intuitively, if we are not so confident in our specification of the network, we might want to account for the fact that other, different networks might also be the ground truth.

As an alternative to maximum likelihood estimation, we can then infer the network A using Bayesian inference [89], which defines a posterior probability that A is the network from which the data were created. The posterior is defined as

$$P(A|X) = \frac{P(X|A)P(A)}{P(X)}. \tag{4.2}$$

Crucially, the posterior distribution does not force us to choose one single network that we believe is the ground truth. Rather, it encapsulates a general notion of how confident we are about all possible networks A.

This extra information, however, introduces new challenges to our notion of dynamical importance. How should we choose dynamically important nodes when the dynamics are happening on a graph we do not know precisely? The solution is to find important nodes in sets of networks broadly compatible with the data. To produce these representative networks, we can use sampling techniques such as Markov Chain Monte Carlo (MCMC) to produce a set of samples $\{A_i\}$ that we know are drawn from the full posterior distribution. Though this inference procedure is mathematically described in terms of the full posterior distribution, it can be algorithmically convenient to think of uncertainty on a set of sample graphs $\{A_i\}$ that could each represent the true graph.

4.1.3 Uncertainty in contagion model

Consider the independent cascade model described in section 1.1.1. Suppose a viral marketing company expects the spread of information about a new product to behave roughly according to this model. In other words, it expects that each individual sharing the product will be equally likely to share it with all their friends. Furthermore, suppose previous products launched by the same company have spread virally. We might use data from these previous products to perform Bayesian inference and determine a posterior distribution $P(p|\text{data})$ for the contagion model parameter. Assuming it expects the next product will spread according to the same class of dynamics as the previous products, the posterior distribution describes our uncertainty of the true model parameters.

Alternatively, imagine instead we have several possible contagion models that we believe might be taking place on the network. A recent study of information spreading and behavioral adoption in rural India is a relevant example here [90]. A company in

India advertised microfinance loans to a few leaders in the community. Subsequently, community members both chose whether to take advantage of the loan as well as whether to share information about it with peers. The team conducting the study posited two models. One model posits a stronger adoption rate for people who heard about the loan from someone who ultimately did accept the loan. The other treats the probability of information sharing as independent of adoption status. This model uncertainty can be encompassed in our framework as

$$P(\theta) = P(\theta|\mathcal{M}_1)P(\mathcal{M}_1) + P(\theta|\mathcal{M}_2)P(\mathcal{M}_2) \tag{4.3}$$

Defining these models separately is necessary when the dynamics of each model belong to a completely separate class. As long as we can specify the probability of each dynamical instance, we have defined our uncertainty of the underlying dynamics.

Now that we have defined a process for quantifying uncertainty for both structural and dynamical parameters, the question becomes how we should go about using this uncertainty to choose optimal interventions.

4.1.4 Bayesian decision theory

Decision theory refers to a broad branch of science dedicated to understanding how decisions are made. *Normative decision theory* refers more specifically to how decisions should be made, defined in terms of a mathematical definition of rationality. Within the framework of normative decision theory, we define the concept of utility as a way of describing a decision-maker's preferences for outcomes.

Ordinal utility refers to an ordering of outcomes according to one's preferences.

If we prefer outcome A to outcome B, then $A \succeq B$, and the ordinal utility function u obeys

$$u(A) \geq u(B). \tag{4.4}$$

When we introduce randomness into the decision-making context, a simple ordering of outcomes does not contain enough information for us to choose an optimal strategy. We need a notion of *how much* better one option is compared to another. Under a set of well-defined rationality axioms, Von Neumann and Morgenstern showed in 1947 that rational decision-making was equivalent to maximizing a *cardinal utility function* [91].

To place the expected-utility framework in terms of contagion models, let's start by assuming a fixed graph G and fixed contagion model parameters θ. We can then define the outcome distribution $P(\mathbf{X}|G, \theta, S)$, which additionally depends on our decision variables S. Our particular notion of importance defines the utility of each outcome $u(\mathbf{X})$. With this utility function defined for all possible outcomes, our decision problem amounts to maximizing the expected utility

$$q(S; G, \theta) = \underset{\mathbf{X}}{\mathbb{E}}[u(\mathbf{X})] = \sum_{\mathbf{X}} u(\mathbf{X}) P(\mathbf{X}|G, \theta, S). \tag{4.5}$$

Reintroducing uncertainty in our graph $P(G|\text{data})$ and contagion model parameters $P(\theta)$, we can write the expected utility for our decision variable as

$$f(S) = \sum_{G} \sum_{\theta} q(S; G, \theta) P(G|\text{data}) P(\theta|\text{data}). \tag{4.6}$$

The key idea is that we use the posterior probabilities of various input parameters to evaluate the quality of a particular decision. From here, Bayesian decision theory suggests we should choose the decision variables that maximize this expected utility.

4.2 Model selection and model complexity

Beyond accurately inferring the parameters of a model, the usefulness of our models also depends on their level of specificity. Just because we want our models to accurately reflect the true world does not mean they should do so for every aspect of reality [92] [93]. Ideally, we should tailor our models to the questions at hand. Excessively complex models confound our ability to interpret them, limiting our trust in their use for policy interventions. The more mechanisms a model has, the less likely we are to be able to determine the effects of interventions from byproducts of arbitrary modeling decisions. Conversely, our model should be specific enough to represent the entire range of policy interventions we want to consider. [1] Moreover, models with a lack of sufficient detail might also mask the richness of the true system's outcome distribution. The tension between simplicity and complexity is well-recognized in both statistical and mathematical modeling and is often described as Occam's razor.

From a statistical perspective, Occam's razor can be managed with the large suite of tools dedicated to model selection. These tools define quantities, such as information criteria and or minimum description length, to judge the quality of fit of a model. However, in small-data regimes, we're likely to select simple models that

[1] Many have raised concerns that using quantitative models causes us to more readily consider interventions for which measures of success are easily quantified. These concerns highlight the importance of the modeling process and, to some extent, operate beyond the scope of this work [94, 95, 96].

might leave flexibility for decision-making on the table. Such regimes are a challenge because we know simple models are insufficient, but we have to use them anyway.

4.3 Deep uncertainty and robust decision-making

As we mentioned in the introduction, the data availability limits our ability to do effective Bayesian inference on complex models. In such situations, it may be impractical to define reasonable posterior distribution for model parameters. This condition is sometimes referred to as deep uncertainty, and the success of formal decision theory might is no longer appropriate [97]. How can we proceed without any formal definition of uncertainty over model parameters?

One might take a scenario-based approach, in which we define a set of scenarios that demonstrate different possible outcomes of the system. This approach is commonly used, though it can be quite ad hoc. A model with appreciable complexity can not be readily intuited from a few scenario choices. The framework of *robust decision-making* lays out a human-centered process for making decisions in conditions of deep uncertainty [97]. It charts a course between formal decision theory and scenario-based approaches. We will not describe the framework in detail, but the basic idea is robust decisions should be designed iteratively using many scenarios. According to the framework, one should define a guess for a robust decision. Then, one should look for weaknesses or scenarios in which the decision fails. Then, more decisions can be defined to hedge against these weaknesses, and more weaknesses can be found until tradeoffs between various model parameters become clear. This human-in-the-loop process provides a way to balance model-based decision-making and human expertise. Human expertise is not something that we should overlook, particularly in scenarios where data is relatively difficult to collect. Especially when fully model-based approaches aren't sufficient, good decision-making requires using all the available resources on hand.

The robust decision-making framework does well in this respect. However, as we continue to improve our ability to collect data, it may be possible to (appropriately) allow models to do more of the heavy lifting. It would be interesting to explore approaches that fall somewhere between a fully formal decision-theoretic avenue to robust decision-making. For example, could we survey experts systematically and construct some kind of prior distribution? Could we build models collaboratively, or in parallel and collate them together? Along these lines, could we encompass a more diverse set of objectives for the system? Recent work on robust decision-making seems to be moving in this direction [98].

4.4 Future work

Our work has invited several interesting avenues for future work.

The first is the notion of bias, a natural discussion point for any work on algorithmic decision-making. If we did implement a formal Bayesian decision-theoretic formalism, as described above, bias could appear in several places. Consider the scenario that nodes in the network have two types and that one group makes up a majority of the nodes.

In data collection, it may be the case that we are unable to capture the entire interaction network completely. In such scenarios, the method of sampling the network has been shown to introduce bias with respect to these groups [99, 100, 101]. This could be due to differences in the number of connections that are attached to at least one member of the majority group compared with the minority. The second place that bias may appear is in the choosing of dynamically important nodes. Some work has been done exploring how structural rankings, such as degree or eigenvector centrality, can be biased by common global properties often found in networks, such as homophily [102, 103]. Similar biases may be introduced in the dynamical context. It may be that the objective function is more likely to select the majority of nodes due to their position in the network. An analysis of this effect would be an interesting topic for future work.

The second line of future work could involve expanding our work to other kinds of network dynamics. Complex contagions are known to exhibit very different behavior in many circumstances. As such, optimal interventions are expected to be quite different. For example, the social reinforcement mechanisms of complex contagion might encourage that they are much closer together in the network [104].

Beyond contagions, voter models represent another class of dynamics that could be studied in a similar framework. Voter models are often used in the context of opinion dynamics, where individuals change their opinions to match those of their neighbors. Interventions could take the form of introducing zealots or nodes that do not change their state regardless of [105, 106].

4.5 Conclusion and future work

Studying intervention in the context of network dynamics is an extremely challenging problem. Fortunately, there are many cases in which the system's causal structure is not so interconnected. For example, if one is interested in designing interventions to improve educational outcomes, the structure of the problem allows one to assume some degree of independence. For instance, it is unlikely that an intervention in one school will affect the performance of an intervention in another. In these cases, the suite of tools dedicated to causal inference is much more appropriate. If we needed to choose which schools, among many, to receive an intervention, a simple ranking would suffice. By contrast, if we were looking to choose amongst students *within* a single school, the interactions between students might make a great deal of difference. Here, taking into account network structure might be appropriate to maximize the effect of an intervention if not necessary. With such problems, computational, methodological, and data-related challenges remain challenging barriers to designing optimal interventions on these systems in practice.

Bibliography

[1] Andrea J. Allen, Mariah C. Boudreau, Nicholas J. Roberts, Antoine Allard, and Laurent Hébert-Dufresne. Predicting the diversity of early epidemic spread on networks. *Physical Review Research*, 4(1):013123, February 2022.

[2] Linton C. Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press, 2004.

[3] Bonnie H. Erickson. The relational basis of attitudes. In *Social Structures: A Network Approach*, Structural Analysis in the Social Sciences, Vol. 2., pages 99–121. Cambridge University Press, New York, NY, US, 1988.

[4] James Coleman, Elihu Katz, and Herbert Menzel. The Diffusion of an Innovation Among Physicians. *Sociometry*, 20(4):253–270, 1957.

[5] Mark S. Granovetter. The Strength of Weak Ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[6] James H. Davis. Group decision and social interaction: A theory of social decision schemes. *Psychological Review*, 80(2):97–125, March 1973.

[7] Harrison C. White. Where Do Markets Come From? *American Journal of Sociology*, 87(3):517–547, 1981.

[8] Nicholas C. Mullins. The Development of Specialties in Social Science: The Case of Ethnomethodology. *Science Studies*, 3(3):245–273, July 1973.

[9] Derek J. De Solla Price. Networks of Scientific Papers. *Science*, 149(3683):510–515, 1965.

[10] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. *ACM SIGCOMM Computer Communication Review*, 29(4):251–262, August 1999.

[11] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the Web. *Computer Networks*, 33(1):309–320, June 2000.

[12] Edward T. Bullmore and Danielle S. Bassett. Brain graphs: Graphical models of the human brain connectome. *Annual Review of Clinical Psychology*, 7:113–140, 2011.

[13] Joel Cohen. ECOWeB 1.1: Ecologists' Cooperative Web Bank. *Cohen Laboratory*, December 2010.

[14] Paolo Crucitti, Vito Latora, and Massimo Marchiori. A topological analysis of the Italian electric power grid. *Physica A: Statistical Mechanics and its Applications*, 338(1):92–97, July 2004.

[15] Giuliano Andrea Pagani and Marco Aiello. The Power Grid as a complex network: A survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700, June 2013.

[16] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, October 1999.

[17] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality: An explanation of the 1/f noise. *Physical Review Letters*, 59(4):381–384, July 1987.

[18] Mark Newman. *Networks*. Oxford University Press, Oxford, New York, second edition, new to this edition:, second edition, new to this edition: edition, September 2018.

[19] Quan-Hui Liu, Juanjuan Zhang, Cheng Peng, Maria Litvinova, Shudong Huang, Piero Poletti, Filippo Trentini, Giorgio Guzzetta, Valentina Marziano, Tao Zhou, Cecile Viboud, Ana I. Bento, Jiancheng Lv, Alessandro Vespignani, Stefano Merler, Hongjie Yu, and Marco Ajelli. Model-based evaluation of alternative reactive class closure strategies against COVID-19. *Nature Communications*, 13(1):322, January 2022.

[20] Dhaval Adjodah, Karthik Dinakar, Matteo Chinazzi, Samuel P. Fraiberger, Alex Pentland, Samantha Bates, Kyle Staller, Alessandro Vespignani, and Deepak L. Bhatt. Association between COVID-19 outcomes and mask mandates, adherence, and attitudes. *PLOS ONE*, 16(6):e0252315, June 2021.

[21] Evan M. Mistur, John Wagner Givens, and Daniel C. Matisoff. Contagious COVID-19 policies: Policy diffusion during times of crisis. *The Review of Policy Research*, page 10.1111/ropr.12487, May 2022.

[22] R. Kinney, P. Crucitti, R. Albert, and V. Latora. Modeling cascading failures in the North American power grid. *The European Physical Journal B - Condensed Matter and Complex Systems*, 46(1):101–107, July 2005.

[23] Priyabrata Chowdhury, Sanjoy Kumar Paul, Shahriar Kaisar, and Md. Abdul Moktadir. COVID-19 pandemic related supply chain studies: A systematic review. *Transportation Research Part E: Logistics and Transportation Review*, 148:102271, April 2021.

[24] William Ogilvy Kermack, A. G. McKendrick, and Gilbert Thomas Walker. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721, January 1997.

[25] M. E. J. Newman. Spread of epidemic disease on networks. *Physical Review E*, 66(1):016128, July 2002.

[26] Eben Kenah and James M. Robins. Second look at the spread of epidemics on networks. *Physical Review E*, 76(3):036113, September 2007.

[27] Kim Christensen and Nicholas R. Moloney. *Complexity and Criticality*. Imperial College Press, 2005.

[28] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, 12(3):211–223, August 2001.

[29] Jacob Goldenberg and Eitan Muller. Using Complex Systems Analysis to Advance Marketing Theory Development: Modeling Heterogeneity Effects on New Product Growth through Stochastic Cellular Automata. *Academy of Marketing Science Review*, 2001:1, 2001.

[30] Peter Sheridan Dodds and Duncan J. Watts. Universal Behavior in a Generalized Model of Contagion. *Physical Review Letters*, 92(21):218701, May 2004.

[31] J. Chalupa, P. L. Leath, and G. R. Reich. Bootstrap percolation on a Bethe lattice. *Journal of Physics C: Solid State Physics*, 12(1):L31, January 1979.

[32] Damon Centola and Michael Macy. Complex Contagions and the Weakness of Long Ties. *American Journal of Sociology*, 113(3):702–734, 2007.

[33] Casper van Elteren, Rick Quax, and Peter Sloot. Dynamic importance of network nodes is poorly predicted by static structural features. *Physica A: Statistical Mechanics and its Applications*, 593:126889, May 2022.

[34] Petter Holme. Three faces of node importance in network epidemiology: Exact results for small graphs. *Physical Review E*, 96(6):062305, December 2017.

[35] Aviral Chawla and Nick Cheney. Neighbor-Hop Mutation for Genetic Algorithm in Influence Maximization. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, GECCO '23 Companion, pages 187–190, New York, NY, USA, July 2023. Association for Computing Machinery.

[36] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, December 2020.

[37] B. K. M. Case, Jean-Gabriel Young, and Laurent Hébert-Dufresne. Accurately summarizing an outbreak using epidemiological models takes time, January 2023.

[38] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, New York, NY, USA, August 2003. Association for Computing Machinery.

[39] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 57–66, New York, NY, USA, August 2001. Association for Computing Machinery.

[40] Zhen Wang, Chris T. Bauch, Samit Bhattacharyya, Alberto d'Onofrio, Piero Manfredi, Matjaž Perc, Nicola Perra, Marcel Salathé, and Dawei Zhao. Statistical physics of vaccination. *Physics Reports*, 664:1–113, December 2016.

[41] Nicholas A. Christakis and James H. Fowler. Social network sensors for early detection of contagious outbreaks. *PloS One*, 5(9):e12948, September 2010.

[42] Marios Papachristou, Siddhartha Banerjee, and Jon Kleinberg. Dynamic Interventions for Networked Contagions. In *Proceedings of the ACM Web Conference 2023*, pages 3519–3529, Austin TX USA, April 2023. ACM.

[43] Stefan Thurner and Sebastian Poledna. DebtRank-transparency: Controlling systemic risk in financial networks. *Scientific Reports*, 3(1):1888, May 2013.

[44] Paolo Crucitti, Vito Latora, and Massimo Marchiori. Model for cascading failures in complex networks. *Physical Review E*, 69(4):045104, April 2004.

[45] Charles D. Brummitt, Raissa M. D'Souza, and E. A. Leicht. Suppressing cascades of load in interdependent networks. *Proceedings of the National Academy of Sciences of the United States of America*, 109(12):E680–689, March 2012.

[46] Flaviano Morone and Hernán A. Makse. Influence maximization in complex networks through optimal percolation. *Nature*, 524(7563):65–68, August 2015.

[47] Benjamin F. Maier and Dirk Brockmann. Cover time for random walks on arbitrary complex networks. *Physical Review E*, 96(4):042307, October 2017.

[48] Sinan Aral and Paramveer S. Dhillon. Social influence maximization under empirical influence models. *Nature Human Behaviour*, 2(6):375–382, June 2018.

[49] Duncan J. Watts. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9):5766–5771, April 2002.

[50] A. Adiga, C. J. Kuhlman, H. S. Mortveit, and A. K. S. Vullikanti. Sensitivity of Diffusion Dynamics to Network Uncertainty. *Journal of Artificial Intelligence Research*, 51:207–226, September 2014.

[51] Şirag Erkol, Claudio Castellano, and Filippo Radicchi. Systematic comparison between methods for the detection of influential spreaders in complex networks. *Scientific Reports*, 9(1):15095, October 2019.

[52] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 75–86, New York, NY, USA, June 2014. Association for Computing Machinery.

[53] Siddharth Patwardhan, Filippo Radicchi, and Santo Fortunato. Influence maximization: Divide and conquer. *Physical Review E*, 107(5):054306, May 2023.

[54] Qixia Jiang and Maosong Sun. Fast Query Recommendation by Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 25(1):1192–1197, August 2011.

[55] Şirag Erkol, Ali Faqeeh, and Filippo Radicchi. Influence maximization in noisy networks. *EPL (Europhysics Letters)*, 123(5):58007, October 2018.

[56] Amit Goyal, Francesco Bonchi, and Laks V. S. Lakshmanan. A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment*, 5(1):73–84, September 2011.

[57] Xinran He and David Kempe. Stability of influence maximization. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1256–1265, New York, NY, USA, August 2014. Association for Computing Machinery.

[58] Laurent Hébert-Dufresne, Antoine Allard, Jean-Gabriel Young, and Louis J. Dubé. Global efficiency of local immunization on complex networks. *Scientific Reports*, 3(1):2171, December 2013.

[59] Maksim Kitsak, Lazaros K. Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H. Eugene Stanley, and Hernán A. Makse. Identification of influential spreaders in complex networks. *Nature Physics*, 6(11):888–893, November 2010.

[60] M. E. J. Newman and R. M. Ziff. A fast Monte Carlo algorithm for site or bond percolation. *Physical Review E*, 64(1):016706, June 2001.

[61] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, 12(3):211–223, August 2001.

[62] P. Grassberger. On the critical behavior of the general epidemic process and dynamical percolation. *Mathematical Biosciences*, 63(2):157–172, April 1983.

[63] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, December 1978.

[64] Pierre-André Noël, Antoine Allard, Laurent Hébert-Dufresne, Vincent Marceau, and Louis J. Dubé. Propagation on networks: An exact alternative perspective. *Physical Review E*, 85(3):031118, March 2012.

[65] Donald Ervin Knuth. *The Art of Computer Programming*. Addison-Wesley, 1998.

[66] Laurent Hébert-Dufresne and Antoine Allard. Smeared phase transitions in percolation on real complex networks. *Physical Review Research*, 1(1):013009, August 2019.

[67] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[68] Xinran He and David Kempe. Robust Influence Maximization, June 2016.

[69] H. A. Bethe and William Lawrence Bragg. Statistical theory of superlattices. *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, 150(871):552–575, January 1997.

[70] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the Second AAAI Conference on Artificial Intelligence*, AAAI'82, pages 133–136, Pittsburgh, Pennsylvania, August 1982. AAAI Press.

[71] Marc Mezard and Andrea Montanari. *Information, Physics, and Computation*. Oxford University Press, Inc., USA, February 2009.

[72] Brian Karrer, M. E. J. Newman, and Lenka Zdeborová. Percolation on Sparse Networks. *Physical Review Letters*, 113(20):208702, November 2014.

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

[74] S. Yoon, A. V. Goltsev, S. N. Dorogovtsev, and J. F. F. Mendes. Belief-propagation algorithm and the Ising model on networks with arbitrary distributions of motifs. *Physical Review E*, 84(4):041144, October 2011.

[75] M. E. J. Newman. Message passing methods on complex networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 479(2270):20220774, February 2023.

[76] Brian Karrer and M. E. J. Newman. Message passing approach for general epidemic models. *Physical Review E*, 82(1):016101, July 2010.

[77] Munik Shrestha, Samuel V. Scarpino, and Cristopher Moore. Message-passing approach for recurrent-state epidemic models on networks. *Physical Review E*, 92(2):022821, August 2015.

[78] F. Altarelli, A. Braunstein, L. Dall'Asta, J. R. Wakeling, and R. Zecchina. Containing epidemic outbreaks by message-passing techniques. *Physical Review X*, 4(2):021024, May 2014.

[79] Sergey Melnik, Adam Hackett, Mason A. Porter, Peter J. Mucha, and James P. Gleeson. The unreasonable effectiveness of tree-based theory for networks with clustering. *Physical Review E*, 83(3):036112, March 2011.

[80] George T. Cantwell and M. E. J. Newman. Message passing on networks with loops. *Proceedings of the National Academy of Sciences*, 116(47):23398–23403, November 2019.

[81] Alec Kirkley, George T. Cantwell, and M. E. J. Newman. Belief propagation for networks with loops. *Science Advances*, 7(17):eabf1211, April 2021.

[82] Andrey Y. Lokhov, Marc Mézard, Hiroki Ohta, and Lenka Zdeborová. Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Physical Review E*, 90(1):012801, July 2014.

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

[84] Benjamin Steinegger, Iacopo Iacopini, Andreia Sofia Teixeira, Alberto Bracci, Pau Casanova-Ferrer, Alberto Antonioni, and Eugenio Valdano. Non-selective distribution of infectious disease prevention may outperform risk-based targeting. *Nature Communications*, 13(1):3028, May 2022.

[85] C. Spearman. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[86] Antoine Allard and Laurent Hébert-Dufresne. On the accuracy of message-passing approaches to percolation in complex networks, June 2019.

[87] Filippo Radicchi and Claudio Castellano. Beyond the locally treelike approximation for percolation on real networks. *Physical Review E*, 93(3):030302, March 2016.

[88] Antoine Allard, Laurent Hébert-Dufresne, Jean-Gabriel Young, and Louis J. Dubé. General and exact approach to percolation on random graphs. *Physical Review E*, 92(6):062807, December 2015.

[89] Jean-Gabriel Young, George T Cantwell, and M E J Newman. Bayesian inference of network structure from unreliable data. *Journal of Complex Networks*, 8(6):cnaa046, March 2021.

[90] Abhijit Banerjee, Arun G. Chandrasekhar, Esther Duflo, and Matthew O. Jackson. The Diffusion of Microfinance. *Science*, 341(6144):1236498, July 2013.

[91] John von Neumann, Oskar Morgenstern, and Ariel Rubinstein. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)*. Princeton University Press, 1944.

[92] Steve Bankes. Exploratory Modeling for Policy Analysis. *Operations Research*, 41(3):435–449, June 1993.

[93] J. L. Borges. *Del Rigor En La Ciencia*. 1946.

[94] Donald T. Campbell. Assessing the impact of planned social change. *Evaluation and Program Planning*, 2(1):67–90, January 1979.

[95] Paul E. Smaldino and Richard McElreath. The natural selection of bad science. *Royal Society Open Science*, 3(9):160384, September 2016.

[96] James C. Scott. *Seeing Like a State: How Certain Schemes to Improve the Human Condition Have Failed*. Yale University Press, 1998.

[97] Robert J. Lempert, David G. Groves, Steven W. Popper, and Steve C. Bankes. A General, Analytic Method for Generating Robust Strategies and Narrative Scenarios. *Management Science*, 52(4):514–528, April 2006.

[98] Robert Lempert and Sara Turner. On Model Pluralism and the Utility of Quantitative Decision Support. *Risk Analysis*, 41(6):874–877, 2021.

[99] Lisette Espín-Noboa, Claudia Wagner, Fariba Karimi, and Kristina Lerman. Towards Quantifying Sampling Bias in Network Inference. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1277–1285, Republic and Canton of Geneva, CHE, April 2018. International World Wide Web Conferences Steering Committee.

[100] Ju-Sung Lee and Juergen Pfeffer. Estimating Centrality Statistics for Complete and Sampled Networks: Some Approaches and Complications. In *2015 48th Hawaii International Conference on System Sciences*, pages 1686–1695, January 2015.

[101] Leonie Neuhäuser, Felix I. Stamm, Florian Lemmerich, Michael T. Schaub, and Markus Strohmaier. Simulating systematic bias in attributed social networks and its effect on rankings of minority nodes. *Applied Network Science*, 6(1):86, November 2021.

[102] Lisette Espín-Noboa, Claudia Wagner, Markus Strohmaier, and Fariba Karimi. Inequality and inequity in network-based ranking and recommendation algorithms. *Scientific Reports*, 12(1):2012, February 2022.

[103] Fariba Karimi, Mathieu Génois, Claudia Wagner, Philipp Singer, and Markus Strohmaier. Homophily influences ranking of minorities in social networks. *Scientific Reports*, 8(1):11077, July 2018.

[104] Grant Schoenebeck, Biaoshuai Tao, and Fang-Yi Yu. Think Globally, Act Locally: On the Optimal Seeding for Nonsubmodular Influence Maximization. *arXiv:2003.10393 [cs]*, page 20 pages, 2019.

[105] Seth A. Marvel, Hyunsuk Hong, Anna Papush, and Steven H. Strogatz. Encouraging Moderation: Clues from a Simple Model of Ideological Conflict. *Physical Review Letters*, 109(11):118702, September 2012.

[106] M. Mobilia, A. Petersen, and S. Redner. On the role of zealotry in the voter model. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08029, August 2007.