Persistent Cascades and the Structure of Influence in a Communication Network

by

Steven T. Morse

B.S. Mathematics, U.S. Military Academy

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Master of Science in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 2017

Author .....
Sloan School of Management
May 18, 2017

Certified by .....
Dr. Natasha Markuzon
The Charles Stark Draper Laboratory
Technical Supervisor

Certified by .....
Prof. Marta C. González
Associate Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by .....
Prof. Dimitris Bertsimas
Boeing Professor of Operations Research
Co-Director, Operations Research Center

THIS PAGE INTENTIONALLY LEFT BLANK

Persistent Cascades and the Structure of Influence in a Communication Network

by

Steven T. Morse

Submitted to the Sloan School of Management
on May 18, 2017 in partial fulfillment of the
requirements for the degree of
Master of Science in Operations Research

Abstract

We present work in identifying, modeling, and predicting the structure of influence in a communication network. We focus on cellular phone data, which provides a near-global population sample (in contrast to the relatively limited scope of social media and other internet-based datasets) at the expense of losing any knowledge of the content of the communications themselves.

First, using inexact tree matching and hierarchical clustering, we propose a novel method for extracting persistent patterns of communication among individuals, which we term *persistent cascades*. We find the cascades are short in duration ("bursty"), exhibit habitual hierarchy and long-term persistence, and reveal new roles in weekday vs. weekend spreading. We show that the persistent cascades in the data are significantly different than what is found in a random network, which we illustrate both analytically and through simulation. We show that persistent cascade membership increases the likelihood of receiving information spreading through the network, even after controlling for overall call activity. Finally, we show that the method is extensible to other communication datasets by applying it to an email dataset. In this case study, we find our approach correctly identifies key individuals, ignores noise, and identifies several interesting email chains.

Second, we propose a probabilistic model for the influence structure of a network, based on a multivariate stochastic process called a *Hawkes process*. We develop a novel approach for parameter estimation in this model that uses a Bayesian expectation-maximization (EM) scheme with a network prior. We first apply the model in the univariate case to the group conversations identified using the persistent cascades methodology. We find that the model performs well as a predictor, and also that the estimated parameter values reveal two types of persistent cascades: low-activity conversations with high temporal clustering, and high activity conversations with moderate temporal clustering. We then apply the model in the multivariate case to samples of the cell phone data, finding that the resulting estimate of the *influence matrix* extends our findings with the persistent cascades.

Technical Supervisor: Dr. Natasha Markuzon
The Charles Stark Draper Laboratory

Thesis Supervisor: Prof. Marta C. González
Associate Professor of Civil and Environmental Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

I thank Dr. Natasha Markuzon, my supervisor at Draper, for her guidance and mentorship. With endless patience, she helped me learn the importance of starting small and working up, making my conclusions clear and not missing the forest for the trees, and organizing technical exposition into a cohesive narrative.

I thank Dr. Marta González, my MIT advisor. I am extremely thankful that she encouraged me to present our work at academic conferences, supported me in these efforts, and helped me navigate the peer review process — these experiences were invaluable to my development as a researcher. I am thankful for her flexibility in allowing me to pursue my interests, and for her experience and mentorship in keeping me grounded in the field.

I am grateful for my fellow students in the Operations Research Center and HuMNet Lab, who provided a sounding board for my ideas, challenged my assumptions, and immeasurably improved my experience as a student and researcher. I would like to specifically thank Shwetha Mariadassou, Phil Chodrow, Deeksha Sinha, Tamar Cohen, Brad Sturt, Sébastien Martin, and Riccardo DiClemente for their help and suggestions with my research and coursework.

Lastly, I thank my wife Kali for her love, support, and sacrifice.

Contents

Introduction

1

In this thesis, we propose novel methodology for *identifying* and *modeling* the structure of influence in a communication network, and we present findings of our methods in several city-scale mobile phone datasets. This chapter will motivate and introduce our research question, introduce several key concepts to our approach, and finally summarize our contributions and the outline of the thesis itself.

1.1 Motivation

Imagine a community of individuals, either small-scale (like a club or classroom) or large-scale (like a city or customer base). They are connected through various social, work, and familial relationships, they communicate both in-person and over a growing array of digital mediums, they have opinions and patterns of behavior — a population described in this way, as a set of measurable, describable properties, we may broadly refer to as a *network*.

Now imagine we are interested in how an idea spreads through this network (or how we might spread an idea ourselves). Who should we talk to? Who are the most effective information spreaders? Who can influence his or her friends' opinion?

Answering these questions requires something more than knowledge of who is friends with who, or who communicates with who, that is, it requires more than understanding just the social network. We are instead searching to understand the *influence* structure of the network.

And indeed, understanding the structure of influence of a network is at the heart of a broad range of applications. For example, diffusion modeling (such as information or epidemic spread) depends on an accurate depiction of the interpersonal influences that can drive the diffusion. Influence maximization seeks a subset of individuals, under such a diffusion model, to then "target" with an idea to maximize the spread of some idea or behavior (such as adoption of a product or a political viewpoint). The entire field of network centrality measures, which aim to provide a quantitative measure of an individual's importance in the network, depend on the fact that the structure they are measuring is meaningful in the first place.

Therefore we focus on the question of understanding the structure of influence, and now elaborate our specific problem statement.

1.2 Problem Statement

The central question of this thesis is *how can we identify and model the structure of influence in a communication network?*

In particular, we are interested in *who influences who*, as measured through observation of interpersonal communication. For example, if A talks to B, does that increase the probability that B will talk to C? How can we identify and model these relationships from large-scale communication data? What is the effect of the resulting analysis on diffusion or centrality?

We divide this question into two parts. First, we examine how to *identify* meaningful communication patterns, and the resulting effect on our understanding of diffusion, the role of individuals in information spread, and their influence in the network. Second, we pursue a *model* of the communication network that allows us to describe the influence structure in a probabilistic way.

This is certainly a challenging problem, although the increasing availability of large-scale communication data (for example: social media, emails) makes analysis possible at a larger scale than ever before. We focus on cellular phone data, which provides a near-global population sample (in contrast to the relatively limited scope of social media and other internet-based datasets) at the expense of losing any knowledge of the content of the communications themselves.

1.3 What is the data?

The advent and popularization of mass-usage technology like cellular phones, social media, and wireless internet has accompanied advances in computing power and increasingly sophisticated theoretical machinery to facilitate a recent explosion of research in the field of complex networks, in particular the study of human communication dynamics and social structure. We can use the observed behavior of individuals in these mediums (cell phone use, social media, etc.) to *infer* the answers to questions like: is A in contact with B? is A friends with B, and if so, how strong? are A and B in the same community of friends? We can move beyond these first-order observations to ask deeper questions like: does A influence B's behavior (to adopt an opinion, change a behavioral pattern, buy a product)? if A receives a piece of news, how many others will hear about it, and how long will it take?

But what data is actually available for this task, and how can we use it to answer these queries?

1.3.1 The shortcomings of social media data

Consider the social media platform Twitter. On this platform, an individual can post short, public messages ("tweets") which then appear in a time-stamped, scrolling "feed" to other individuals who have chosen to follow him/her. This translates easily to a network interpretation: if A follows B, there is a directed edge from B to A (indicating B may exert influence on, or pass information to A — sometimes termed: B "dominates" A). Twitter also provides a rich sense of information spread: we can directly, explicitly monitor the spread of an idea (often helpfully codified with a hashtag by users, e.g. `#Election2016`). We can observe, for example, who of an individual's followers "retweets" (reposts) his/her content. We can build models based on these observations to predict what type of content is most likely to be spread, or most likely to generate new followers, etc.

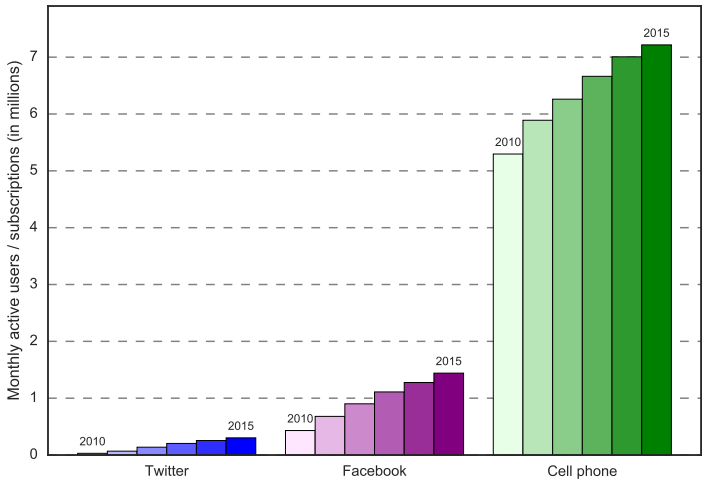


Figure 1.1: **Why cell phone data?** Cell phone users constitute a larger fraction of worldwide population than any other communication platform, in particular social media. Depicted above is a comparison of monthly active users on two social media sites against cellphone subscriptions, worldwide, by year, 2010–2015. Source: ITU, `worldbank.org`

Furthermore, all this data is publicly available (with some rate-limiting constraints) with a publicly available API to interact with the platform.

However, there is a catch. The size and scope of the individuals comprising the Twitter network (the "Twitterverse") is limited. There are approximately 330 million monthly active users worldwide — this is an enormous sample size by comparison with more traditional methods like surveys, but it is still small compared to other available datasets that are on the order of billions, as we will see. More critically, however, it represents a limited demographic. Twitter users are a narrow subset of the population, they are underrepresented especially in developing parts of the world, and use of the service requires internet access.

Other large-scale social media has similar strengths and shortcomings. Facebook has greater worldwide penetration than Twitter, but it is again dependent on internet access, it is unavailable in many parts of the world, and its data is strictly proprietary. Internet blogging data is public (in the sense that it is collectable), and rich in content information, but extremely limited in size and scope.

1.3.2 The case for cellular phone data

By contrast, consider mobile cellular phones. The cellular phone was invented in the early 1970s and has steadily gained in popularity since, to the point that by 2015 the number of cellphone subscriptions account for over 95% of the world's population. Even with careful consideration for the inevitable double-counting that goes on with accumulating this kind of statistic (for example, in India it is common to use two SIM cards per phone), estimates place the number of individual mobile phone users in the world at well over 5 billion. Cellphones have become pervasive as they have become the epicenter of interpersonal communication. For one, the growing availability of internet-enabled phones ("smartphones") brings phones into the "internet of things" and allows communication over a wide variety of internet based messaging mediums. However, smartphones

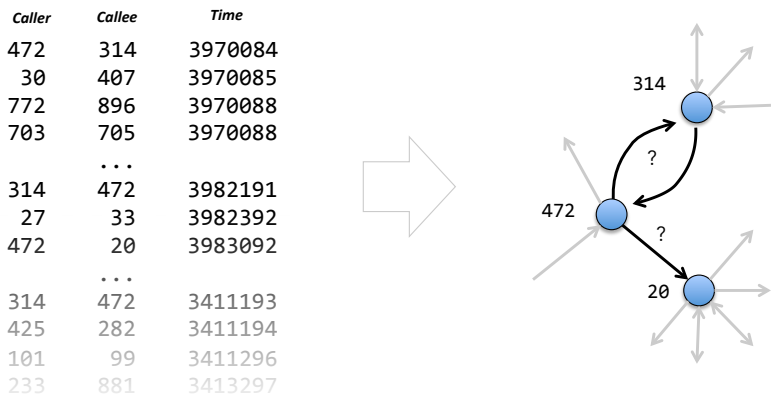


Figure 1.2: **How can we infer network influence structure?** In large-scale communication metadata (e.g. cell phone data), we have information about only the sender, the recipient, and the timestamp for each event. The number of unique individuals is in the hundreds of thousands, and the number of events is in the tens to hundreds of millions. The problem is to infer not only the social relationships, but the structure of influence, from these anonymized, content-less sequences of events. That is, we are interested in not only that A often contacts B, but instead questions like: does A calling B increase the probability that B calls C? does A's call carry more meaning for B than a call from C? do A or B have identifiable roles in spreading information to others?

aside, the use of basic cellular services — namely, calls and SMS texts — over a cellular network constitute a staggering amount of communication events on a daily basis, across the world.

Advantages. This omnipresence provides a near-universal sample of individuals in any demographic category or geographic area. In particular, it provides a sample of unfiltered communication (and often, movement) activity of populations in otherwise-unreachable parts of the globe, such as developing countries or impoverished areas without internet access. Other forms of data (like social media, or census surveys) is sparse or non-existent in these areas.

A second advantage of cellular phone data is its anonymity. In social media, individuals know that their activity is public, to some degree — your friends, followers, and in some cases (such as Twitter or Instagram) anyone with an internet connection, can view and monitor your activity. This creates an inherent *filter* on behavior that can be beneficial, in the sense that there is a higher amount of social capital invested in each post or tweet which heightens the meaning and importance of each event, but is also limiting, since we are seeing a carefully curated version of the real underlying social, influence, or communication structure of the network.

Cell phone data does not have this limitation. Cell phone users make their calls with the assumption that it is a private call between two people. They trust that their service provider keeps this information confidential, if they even think about it at all, and indeed, service providers only provide researchers an anonymized version of the data with names and identities stripped in order to maintain this trust. This allows the researcher an unfiltered and extremely granular perspective into the daily interactions of individuals with their social and business contacts.

Limitation. The drawback, then, to using mobile phone data is that we do not have the luxury of *content knowledge* that we do with social media data, or blog monitoring, or email datasets. We have only an anonymized dataset giving the **caller**, the **callee**, and a **timestamp**. In most cases we also have information about the call **duration**, and many times the **location** of the call, for example

what cellular tower the calls went through. So we may know A called B at 10:46 a.m., but we do not know if it was to discuss a business merger, lunch plans, or if it was simply a wrong number. Figure 1.2 illustrates this puzzle.

This is a serious limitation. Even in rich datasets with content knowledge, or demographic information, or survey data, correctly representing the real underlying network structure is a tall order. Removing all such layers of supplemental information increases the difficulty, and researchers have spent much effort developing methodology that attempts to infer the correct network structure and dynamics from this content-less metadata, as we will discuss in the next chapter.

Previous approaches and way ahead. We can imagine a few approaches to address this limitation:

- **Counting calls.** A straightforward first-estimate is to count the number of observed calls between two individuals, and set some threshold to infer there is a meaningful connection between them: for example, at least 2 per month, or at least one pair of reciprocated calls (A must call B and vice versa in order to establish a connection, [57]). This certainly implies the connection is not just an accidental call, but it still does not give us a very good idea of the strength, or influence, of the connection. We know A calls B three to four times every month, but we do not know if it is a manager doing check-ins with a project leader, a student calling to his favorite takeout restaurant, or just a resilient telemarketer.

- **Observing post-emergency.** Another technique is to focus observation on calling patterns after an emergency event, such as an earthquake or bombing or major sporting event. It seems reasonable to assume that calls made after such an event are more meaningful. So we can focus our search, such as what subset of contacts does someone call post-emergency as opposed to other times? who tends to initiate calls in crisis situations? if we observe a heightened volume of call activity, how long does it last? Et cetera. Works like [8] in this vein also focus on the post-crisis pattern of calls, which tends to be rapid *cascades* of communication (tree-like spreading). However, limitations of this approach are that it restricts us to an overly specific kind of information spreading, and, even worse, limits us to an extremely small portion of the available data.

- **Recurring patterns.** The previous ideas of seeking recurring calls or meaningful patterns can be generalized to looking for recurring patterns: when A calls B, B always calls C and D. This type of pattern-mining approach allows us to use the entire dataset, it gives us ideas about who tends to initiate different types of patterns, and we can start to approximate the structure of influence in the network. A common line of study is to search for recurring *motifs*, such that the object of interest is the shape of the pattern, not the particular individuals involved (e.g. [33]). Also common is to focus on identical (*isomorphic*) patterns. However, the focus on motifs gives us only an abstract picture of influence structure at a population-level, and the focus on isomorphic patterns seems to forget the noisy nature of human communication, casting aside relevant information in the process.

- **Probabilistic model.** The pattern-mining approach unfortunately does not provide us with any model of the patterns it finds, or of the connections themselves. For this, we must construct

a probabilistic model. For example, we might imagine there is some probability that A calls B, and then some conditional probability that given A called B, what is now the probability that B will call C. We formulate this model and estimate its parameters. In return, we have a way of quantifying the network and, importantly, a way of predicting future behavior. (E.g. see [19].) But we must be cautious with this approach: a network of hundreds of thousands of individuals exchanging millions or billions of calls can quickly lead to an explosion of model complexity.

Our thesis extends work in the latter two approaches by proposing new methodology which answers the concerns mentioned, as we describe in the next section.

1.4 Approach and Contributions

Two related questions guide this thesis, as elaborated in the problem statement, and we address each in its own chapter.

- In **Chapter 2** (Background), we set the stage by introducing the rich history of researching social and communication networks. We try to cast a wide net, while focusing on a select few themes, namely: the strength of interpersonal ties, temporal networks, and network influence structure.

- The first main chapter, **Chapter 4** (Persistent Cascades), introduces a novel method for finding patterns of information spread when we know nothing about the content of communication. We frame information spread as a cascading structure, and use methods of inexact tree matching and hierarchical clustering to extract long-term, recurring group conversations we term *persistent cascades*. Analysis of these persistent cascades reveals new roles in information spreading and the influence of certain individuals. We also show the effect these group conversations have on notions of information spreading or centrality in the network. We perform the majority of the analysis on three large mobile phone datasets. Finally, we show that the methodology is extensible to more general datasets by demonstrating its use in an email dataset — this also allows us to test the claim that the persistent cascades are indicative of information spread, since we have knowledge of the emails' content.

- **Chapter 5** (Modeling Influence Structure) takes a probabilistic modeling approach to this problem of determining the influence structure and information spreading dynamics of a communication network. Specifically, we frame the interactions of individuals as instantiations of a multidimensional stochastic process, and show that by incorporating mutual-excitation in this process we can capture the influence structure of real networks. We introduce two novel extensions to existing work in this area: we derive a regularized expectation-maximization (EM) algorithm that allows incorporation of a Bayesian prior on the influence structure, and we apply a *dyadic* version of the model, that is we model each dimension as the pairwise interactions of individuals.

The final chapter concludes and proposes avenues for future work, in the areas of pattern-mining, identifying and modeling influence structure, and predicting behavioral influence.

1.4.1 Notes on style

This thesis is written in the first-person plural, both to keep a technical tone and to reflect the collaborative nature of the work. None of the methods or analysis in this thesis would have been possible without the constant advice and feedback of my advisors and colleagues in the MIT Operations Research Center. The document was compiled with LaTeX, using the free Bembo-like font `fbb`, and with selected formatting from the `classicthesis` template. Nearly all the code was written in Python, with the `networkx` package for networks and the add-on package `seaborn` for plots.

Background 2

This background chapter provides an overview of the relevant highlights of the history of applied network science as it applies to our study of influence structure in a communication network. We focus our review on three interrelated themes: (1) the strength of interpersonal ties (or, not all connections are created equal), (2) the phenomenon of temporal clustering in interpersonal communication and its effect on diffusion dynamics (or, human beings are "bursty" and why that matters), and (3) extracting the latent network using deterministic and probabilistic methods (or, how to infer the real underlying structure from limited observations).

We aim to keep the discussion at a high level throughout, and leave more technical discussion for the subsequent chapters to expand upon as it relates to their content.

2.1 Complex networks

The study of complex networks has its origins in the field of *graph theory*, which most consider to have begun in the 18th century with Leonhard Euler's well-known problem about navigating all seven bridges of Königsberg without crossing any bridge twice (he proved it was not possible). Graph theory introduced the concepts of *nodes* connected by *edges*, giving a powerful method of abstracting a wide variety of problems. Many famous questions, such as the Traveling Salesman problem (how can we find the shortest path that visits all cities in some geographic area?) or the Four Color problem (is it possible to color the countries on a map with four or fewer colors without any adjacent countries sharing a color?) can be readily reframed as graph theoretic problems.

By the 20th century, researchers in a large variety of fields — social science, biology, computer science, economics, transportation, to list a few — were applying the high-level concepts in graph theory to model and analyze problems in their discipline, under the more applied moniker of *network theory* and *complex networks*. The transportation system of a city, the neural system of the brain, the friendships in a social club, the trade agreements between countries, the structure of the internet, the predator-prey interactions of an ecosystem — these, and countless others, are examples of applied problems we can express and analyze using the rich field of network theory. That is, we can reimagine individuals (or countries, neurons, etc.) as nodes in a graph, and we can represent relationships (or roads, treaties, etc.) as edges between them.

Indeed, since these types of applied problems are typically large-scale (thousands or millions of nodes), with non-trivial structure and nuanced interaction dynamics, we often refer to them as *complex networks*, or more broadly, *complex systems*. This paradigm at once provides a beautiful

abstraction and a large toolset of mathematical techniques for analysis.

In this thesis, we focus on a particular subset of problems dealing with the structure and dynamics of influence in a human social network. We are interested in understanding things like the dynamics of information spread, the role of individuals in their social communites, and the structure and interplay of influential relationships.

Our line of study also leads us to emphasize two particular types of complex networks. First, we will attempt to always incorporate *temporal* knowledge of the network; for example, we are interested in not only that two individuals are social contacts, but when and for how long. This focus on *temporal networks* has only recently become a common approach: early work tended to rely on aggregated information, as we will discuss in the next chapter, despite its insufficiency to describe essentially temporal problems like influence and information spread dynamics. Second, we will focus on the idea of *communication networks* (as opposed to social networks), to emphasize the fact that we are doing all our inference (of influence, spreading dynamics, etc.) based on data that is purely communication between individuals. Also, we will tend to avoid the term "social" to emphasize that our aim is not to understand friendship and community, as much as to infer influence through observation of interpersonal communication.

2.2 Strength of ties

In understanding the interactions of individuals in a communication network, at the most basic level we seek to understand the dyadic, interpersonal relationship between two people. These two individuals' "strength of tie" simply refers to the flexible notion of the degree of friendship, or trust, or collaboration present between them. It may be directed, it may be temporally dependent, it may be binary or discrete or on a continuous spectrum. In networks, where individuals are nodes and relationships are edges in a graph, we may attempt to codify this tie strength as an *edge weight*.

Granovetter, in a landmark 1973 paper [22], introduced the idea that interpersonal ties vary in strength. He explored this idea out of an interest of understanding how micro-level processes could affect macro-level change, and he postulated that *weak ties* are actually *strong* because of their importance in spreading processes by connecting distant, tightly connected cliques. There are several critical ideas here: first, the idea that "triadic closure" is inevitable in the presence of strong ties. Second, the extension of this line of reasoning to the hypothesis that since strong ties beget strong ties, creating tightly clustered communities, then the only ties connecting these communities (the "bridges") are weak. And so, these often-ignored weak ties are actually responsible for much of the dynamics we see in spreading of information/disease/chain-letters; thus the "strength" of weak ties.

Researchers extended and riffed on this idea for several decades (see [14, 69, 3, 61, 4, 15, 23]). Notably, Burt [13] introduced the concept of "structural holes" by pointing out that the "causal agent in the phenomenon is not the weakness of the tie but the structural hole it spans." So he frames the discussion in terms of competitive advantage: one should position himself to be connected to non-redundant communities, and in some way the gate-keeper for the inter-community bridges.. This immediate interpretation in terms of competition extended to economic ideas like *embeddedness*, such as in Uzzi's finding [69] that strong interfirm networks rely on a mixture of "arms-length" (contractual) and "embedded" (personal) relationships (e.g. lower detail and complexity through arms-length ties vs. less robustness to change with only personal ties).

However, Aral et al. in [6] point out that although weak bridging ties may provide the most novel information, their "bandwidth" is inherently lower (fewer or less meaningful interactions), while a strong tie has higher bandwidth but more redundant information. They test this idea on a dataset of email exchanges and show that often, higher bandwidth access to redundant information will outgain low bandwidth access to diverse information. This interesting counterpoint actually has parallels in the literature of human communication patterns in the statistical physics community; for example the influential work [57] finds weak social contacts are less important than mid-strength social contacts in spreading dynamics due to the same issue of bandwidth.

In the context of collaboration and group diversity, Hansen [23] similarly observes that *complex* information must travel over *strong* ties, so in general terms: weak ties speed up simple information transfer, but slow down complex projects. His study used new development projects at an electronics and technology company (c. 1994–8), and at heart, used a simple network survey to the R&D leads (incorporating directionality), and enriches it with years active, licensing agreement info, budgets, patents, etc., and using this structure to fit against the dependent variable, project completion time. We note that here, Hanson focuses on the *ability* of ties to transfer knowledge across clusters, but only mentions another challenge, that of *unwillingness*. This plays to the competitive subtext at play in this sort of analysis, since for example, one cluster (team) may not want to let knowledge transfer across a weak/strong tie because it lessens their competitive advantage.

To give another example, in his work on the role of peer thresholds in group dynamics, Granovetter [21] points out the surprising importance of individual decisions (and distributions of thresholds) on aggregate outcomes, with the colorful example of the 100 rioters in a square. He then shows that social structure (heavier weights on strong ties) — among other considerations such as sampling differences (missing key individuals in the group) and spatial consideration (individuals moving from one area to another) — complicates the matter.

Nevertheless, often we decide to treat all ties are the same, typically for practical reasons. For example, Watts [74] gives a mathematical formulation to the basic model presented in Granovetter [21], and analyzes the effect of the distribution of thresholds, but not of different tie strengths, for reasons of analytic clarity. Domingos and Richardson [16] encode the effect of an individual's neighbors on his probability to adopt as $P(X_i|N_i)$ where X_i is the boolean random variable constituting i's decision, and N_i are i's neighbors; we note that this treats N_i as a single block entity, without specific weighting for strong or weak interpersonal ties.

To summarize thus far, it is clear that central to the understanding of diffusion, collaboration, information flow, opinion spread, and other processes on human networks, is an understanding of the underlying interpersonal dynamics and tie strength. Although we can (and often do) encode these interpersonal dynamics as a static, binary edge (i.e. present or absent), the studies just summarized (and others) claim adamantly that this is an oversimplification in many cases, and at the very least we should attempt to encode the interpersonal tie on a spectrum.

2.3 Interaction dynamics

Just as the non-homogeneous strength of ties is often assumed away out of interest for lack of data or desire for a parsimonious model, researchers often make an assumption of Markovian behavior when modeling human interaction patterns. That is, they model each interaction as depending only

on the last, with no "memory" of the history of past events. This can be stated equivalently in many cases as a "memoryless" or "Poisson" assumption, after the Poisson process which adheres to these same properties (for example see [62, 21, 15, 29, 74]).

Consider some of the early work on diffusion, such as the ideas in Shelling [62] on "sorting and mixing" which studies social sorting patterns under the over-arching hypothesis that individual decisions lead to collective dynamics. His running example is with race: individual decisions of where to sit at a cafeteria lead to tipping points, causing macro-patterns of segregation. He conjures a model to explain this, using dimes and pennies on a grid, or in more technical parlance we might say agents with two classes of decision rules in a lattice, and shows by example how equilibrium or total segregation might arrive. This was a fascinating analysis that is at the heart of much later research on behavioral contagion; however, it interestingly neglects the importance that the *order* of these decisions has on the eventual outcome. In fact, Shelling says "it usually turns out the precise order is not crucial to the outcome," but we can imagine this is rarely actually the case — for example, a chain reaction ripping through the center of the grid to start the game, seems like it will have a very different outcome than an equal number of interactions scattered uniformly across the grid in random order. In fact, if these two scenarios did give the same final outcome, that would be a surprising result.

It seems the timing of the decision is as important to the collective dynamics as the decision itself, and the two shape each other. However, we must ask if this idea is borne out in reality. This question leads us to the large body of work on the so-called "burstiness" of human activity patterns, and its effect on network dynamics.

2.3.1 Temporal clustering ("burstiness")

Similar to the shift in thinking about network density in the early 2000s resulting from the observation that real networks' connectivity tend to follow a power-law distribution [10], or are small-world [73], etc, there was a few years later a change in thinking about *interaction patterns* in works like [9, 27, 28, 50, 48, 71]. Specifically, researchers observed that the time between interaction events for a given individual was not exponential, but power-law distributed. In other words, individuals tended to have bursts of communication activity followed by long periods of inactivity. Information spread happens in bursts, the argument goes, because receiving (or generating) a piece of information causes us to send it to others: an email about a meeting time change causes us to forward it to several peers, a decision to change dinner plans causes us to call the other guests.

Barabasi [9] was one of the first to report on this phenomena, based on a dataset of email correspondence, and he referred to it as "burstiness" of activity pattern. He focused on *individual* burstiness, and postulated that the mechanism was humans' natural tendency to prioritize: the highest priority tasks get executed first (short inter-arrival times), while the low priority tasks sit for long periods of time while the high priority tasks are completed (thus creating the long tails). Later work focused on *group* burstiness, with the postulated mechanism being the causal nature of receiving and relaying information ([50, 71, 27]).

Regardless of the mechanism, the basic observation of *temporal clustering* in human activity patterns has been observed in widely different contexts, and we note may refer to a pattern of clustering at the individual level, dyadic (interpersonal) level, or even at a group level.

2.3.2 Effect of temporal clustering on spreading dynamics

Much of this early work on the temporal patterns of human activity patterns also pointed out the *slowing effect* the patterns have on spreading dynamics. In other words, we tend to observe less total spread (epidemic spread, reach of a viral email, etc.) under the actual order of events than we would under a random reshuffling of the same events.

Iribarren and Moro [27] showed the activity patterns in viral information spreading was well-modeled as a non-Markovian branching process (in particular a Bellman-Harris branching model) borrowed from biology, such that each email is a potential "ancestor," with some probability of creating offspring, and so on. Branching processes naturally exhibit temporal clustering, as we will investigate later, but can be understood intuitively by thinking of the treelike structure of such a process and the densification that comes from multiplying offspring. It also seems straightforward, but is a stark shift from a typical model of population change such as the susceptible-infected model from epidemic spread, $i(t) \sim i(0)e^{a_0 t}$, where $i(t)$ represents the fraction of infected individuals. This classical model makes the critical assumption that most of the diffusion occurs around the average interarrival time, and therefore new diffusion by individuals that have already spread information (or, have already infected others), is highly unlikely for large response times or interarrival times. This type of branching process analysis in work like [71] and is related to the non-Poissonian stochastic processes of a later section ([7, 24, 70]).

In [50], the authors give an interesting mathematical argument for why spreading may appear "slowed" under the true, bursty dynamics compared to random interactions, and show why under certain conditions it is actually more effective. In essence, when the infection rate is low, bursts of activity are a more effective mechanism of spreading than random mixing, while the opposite is true as the infection rate gets higher. (We cover this particular idea in more detail in Chapter 3.)

Now let us return to the effect this may have in research that makes the Poissonian assumption. An example is the classic study by Steglich et al. [64], which attempts to disentangle *selection* (i.e. edge formation) from influence. (E.g. did the teenager start smoking because he was friends with smokers, or is that why he befriended them in the first place?) They examine this problem with a classic experiment, a 3-year study of Scottish teenagers and substance use (drugs and alcohol), and a model that incorporates both binary opinion spread and tie formation. They essentially try to identify the magnitude of network influence (vs. the simple predisposition already present in an individual) by fitting a network behavior model to survey data at multiple snapshots. However, their network model which is meant to "fill in the blanks" makes the assumption of Poissonian interaction, and thus we know is allowing much more mixing of behavior than is likely to be occurring in reality. In this way, it seems their model would systematically underestimate the role of network effects, since they would be able to use very conservative parameters on the network's role to capture the real observed effects.

Lastly, however, we must temper our train of thought by remembering that observation of temporal clustering does not equate to causal inference. Aral et al. [5] point out that much of the temporal clustering observed in behavioral contagion of product adoption can be attributed to the preexisting tendency for friends to think and act alike anyway — that is, we may observe several friends buying the new iPhone within days of each other, and false attribute this temporal clustering of adoption to peer influence, when in fact the friends were friends in the first place because they

all share a love of the Apple ecosystem and buy any new products the day they are released. (This property of friends being by nature like-minded is termed *homophily*.) On the other hand, this finding of homophily clouding our ability to infer causality from temporal clustering is particularly aimed at the scenario of product adoption and peer influence, and not the more general problem of information spread.

2.4 Extracting the latent network

But how do we determine which ties are strong or weak, and their temporal influence on each other? Many of the papers just presented had access to rich data where we had access to things like personal interviews asking "how much do you trust this friend," but even this information doesn't tell us much about a question like: "Given that trusted friend A told you X, what and when do you tell somewhat less trusted friend B?" Worse, if we have data like call records, or email communication, or even if we have interviews and surveys but believe we cannot trust them completely, or if we are doing thought experiments on a theoretical network where data does not exist, how do we make any inference to the true underlying tie strength or temporal relationships?

This line of thinking leads us to frame the question as one of finding the *latent network*. We can imagine there is a true network generating our observed data, with true interpersonal weights and temporal influence dependencies, that we must infer.

2.4.1 Static networks: aggregated approach

A straightforward approach is to consider the network a static object and aggregate observations over some time window to determine the edge weights. Much of the early work in analyzing large-scale human activity and communication patterns through mobile phone records, or call detail records (CDRs), used some form of the this approach. For example, Barabasi, Onnela et al [57] required that calls were reciprocated over the course of say, two weeks, to assign the two individuals an edge. In [35] they experiment with different time windows and find that one month provides the most stable network.

2.4.2 Incorporating temporal knowledge

Later, the temporal nature of human social networks reentered the picture. Aggregating over a month or year gives a falsely inflated sense of the number and strength of contacts that a person maintains, as most people are constantly shifting the groups we most closely associate with — we make new acquaintances, old ones move away, etc. Miritello et al. [49] give a remarkable exposition of several of the effects of this temporal consideration, again using CDRs, by showing that people have a "social capacity," i.e. a relatively small number of friends they are actively communicating with at any one time, even though in aggregate their contact list may be very large. They also showed that people had tendencies to be "social explorers" or "social keepers," with high or low friend turnover, respectively. These roles have a direct effect on diffusion dynamics, and — perhaps surprisingly — the social explorers tend to receive information (or infection) at a delayed clip than the social keepers, evidently since their constant shifting of contacts is a slower mechanism than the rapid, deep penetration seen among social keepers.

2.4.3 Temporal motifs and deterministic methods

The importance of temporal consideration thus established, we can still imagine that we have a poor approximation of the true network with this approach. For example, the chief executive may call her office front desk 2-3 times a week, but in terms of influence, this probably carries less weight than the twice a month call to her regional director. How can we take the temporal knowledge into account to learn the causal structure of the network?

One line of research into studying causal structure in the field of temporal networks is through finding *temporal motifs*. If we see person A contact person B, who contacts person C, who calls back person A, and we lots of other such time-respecting 3-cycles involving other users ($D - E - F$, $X - Y - Z$, etc.), we might be interested if this motif occurs more than we expect under some null model, and if so, why? This is essentially an extension of the older idea of (static) network motifs popularized by M.E.J. Newman and others, to temporal networks. Notable works in this vein are [76] who introduce the idea and examine patterns in CDR and Facebook wall-post history data, and find that certain communication motifs (such as the "2-person ping pong") occur at a much higher rate than found in a randomly generated network of the same size. In Kovanen et al. [33] they provide a more robust framework for this problem, and in [34] follow up to link demographic patterns to observed communication motifs (such as that all-female "star" and "chain" motifs are more common than the respective all-male motifs). Leskovec et al. perform an analysis of motifs on blog post data [39].

These are fascinating studies but, by focusing on population-scale patterns and abstract motifs, still lack an ability to identify causal relationships at the individual and interpersonal level. The chain motif tells us a meso-scale story, but very little about any of its constituent members. There are some deterministic approaches to using this idea of recurrent patterns to examine temporal structure at the individual level, for example finding frequently recurring "dynamic graphlets" such as [26] or "heavy subgraphs" such as [12]. In addition, we will present a deterministic method of this family in the next section as precursor to our proposed work.

2.4.4 Probabilistic models

However, this graph-mining and deterministic approach lacks an ability to model the network, or quantify the observed structure in a probabilistic sense. We may extract a recurrent temporal structure, but we have no way of describing how sure we are about its various parts (back to the problem of tie strength), or much less being able to predict the occurence of the structure in the future. For this we turn to two more recent, and closely related approaches to the problem: (1) modeling the network with conditional probabilistic structure (such as [19, 16, 20]), and (2) modeling the network as a point process (such as [63, 58, 43, 72, 78]).

The first approach essentially views a recurrent pattern as a Bayesian network, or decomposition of a joint distribution: given that A called B, what's the probability that B will call C? This tells us something about each interpersonal relationship, and the joint distribution tells us about the group as a whole. One well-known approach in this vein is due to Gomez-Rodriguez, Leskovec, and Krause [19] who explore the closely related problem of having observed that A, B, and C all received some piece of information, what is the most likely path that they received it? They avoid the combinatorial difficulties of this search space through a clever application of the Independent

Cascade model and the assumption that the information spreads in tree-like shapes. The early work in influence-maximization, such as Domingos-Richardson [16] also took a version of this approach, as they recognized that the probabilities of influence were inherent to the structure itself. Goyal et al. [20] extend these models to both continuous- and discrete-time, and are able to effectively *predict future actions* to within a time interval.

The second approach is to model the network as a point process: i.e. that each individual, or dyadic relationship, or group, represents a stochastic process, and "events" (such as calls, or emails, or blog reposts) are modeled as "arrivals" on that process. In some sense this is simply adding the flexible modeling structure of stochastic processes to the idea of conditional interdependence from before, since the probability of an arrival in the process is conditioned on previous arrivals, and possibly even other arrivals in other parallel processes. Simma and Jordan [63] provide a model where each event triggers a Poisson process of successor events, and they learn the parameters of each using a (distributed) expectation-maximization approach. Perry and Wolfe [58] implement a multivariate point process such that each process represents pairwise interactions (A to B or vice versa), and each process' intensity rate is influenced by its own history *and* select other processes. (They apply their method to a corporate email dataset, and so also interestingly extend it for "multicast" events, such as mass emails where A calls B, C, and D simultaneously, which is not a concern with other mediums.)

The Hawkes process, which we will use in this thesis, was first described in 1971 [24] and became popular in microeconomics for modeling the interdependencies and volatility of stock fluctuations. It is a highly flexible and robust framework, with widespread application: Veen and Schoenberg [72] use a spatial version to predict seismic activity, Stomakhin, Short, and Bertozzi [65] predict gang activity, Zipkin et al. [79] apply it to email correspondence among cadets at the U.S. Military Academy, Pinto et al. [60] uses it for trend detection, Valera and Gomez-Rodriguez [70] for product adoption, and others.

Interestingly to both these approaches, they are able to capture temporal clustering and tie strength in an intrinsic way. Without specifying a model of interarrival times (e.g. power law, exponential), these probabilistic models will settle upon parameters that capture the true temporal clustering dynamics. Also, the resulting probabilities (or in the case of point processes, the process intensities) give us an immediate, meaningful, and robust quantification of the tie strength. With only observations of network activity, and no knowledge of content, we can infer simultaneously strong and probabilistically grounded notions of both temporal activity patterns *and* interpersonal tie strengths.

In summary, the problem of modeling network interactions, taking into account temporal structure and interpersonal tie strength, is gaining attention as researchers realize the importance to understanding the role and mechanisms of networks in a wide variety of applications.

Data

3

The datasets we use throughout this thesis are mobile phone datasets, sometimes referred to as Call Detail Records (CDRs), from three cities and their greater metropolitan area. In this short chapter, we give a description of the data to frame the analysis which follows in subsequent chapters.

3.1 Summary

The CDRs come from two mid-size European cities ("City A" and "City B") and one Central American city ("City C"). The data for City A and B covers a period of 13 months, while City C covers 5 months (although it is higher volume).

Each event in the CDRs contains at minimum the **caller**, **callee** (who the caller called), **timestamp**, and **duration**. This level of information is standard in this type of data. A single event in the CDRs corresponds to a single phone call or SMS event, as recorded by the carrier. (Specific technical information about determining between these two type of events is proprietary to the carrier and not disclosed.)

In two of the three CDRs (City A and B), we have additionally the nearest tower **location** for the caller and also for the callee; this information is partially available for City C. We also have access to the latitude/longitude for these tower locations indexed in a separate file.

For a given month, in City A, there are about 331k (331×10^3) unique users, making a total of 6.3 million call/SMS events. City B has about 258k unique users making 3.9 million call/SMS events. City C sees about 1.7 million unique users each month, making 154 million call/SMS events. Individuals in City A and B make an average of about 10–11 calls per neighbor per month, however this is skewed high by a group of users with high activity. (This information was not collected for City C.) See Table 3.1 for a complete listing of summary statistics for Cities A and B.

Figure 3.1 illustrates the predictable population-level patterns in overall call activity at a week scale. We see that in general, individuals are about half as active on weekends compared to weekdays, and that this pattern is highly predictable. (The only two weekday outliers (in mid-June and mid-

Table 3.1: Summary statistics for CDR datasets.

City	Unique IDs ($\times\mathbf{10^3}$)		Calls ($\times\mathbf{10^3}$)		# months	Degree	Calls /edge/mo.
	avg. / mo.	total	avg. /mo.	total		(k), avg.	(w), avg.
A	331.2	648.1	6,334.6	82,350.1	13	3.88	11.59
B	258.0	523.5	4,172.2	55,747.5	13	3.62	10.52

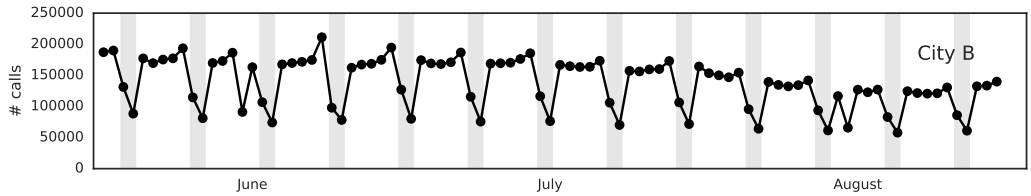
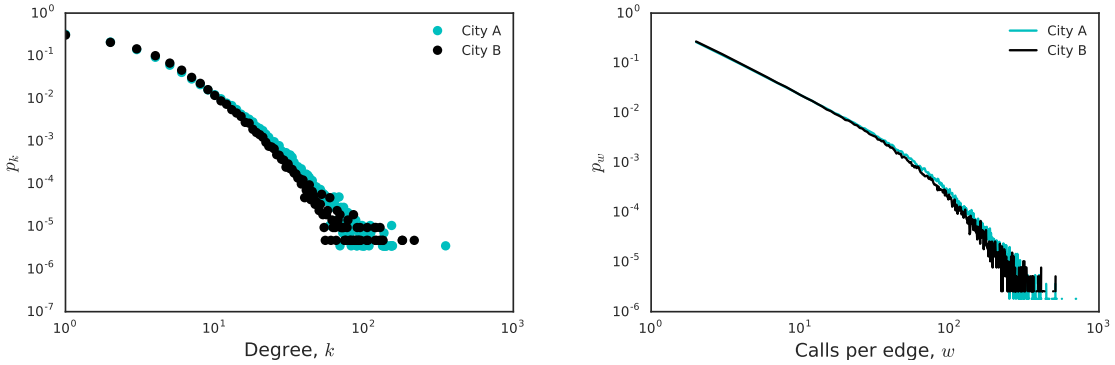


Figure 3.1: **Call activity follows predictable population-level patterns.** This chart depicts the weekly rhythm of weekend dips in overall activity, as compared to weekdays. Each point in the charts above corresponds to the total activity for a single day in one of the CDR datasets. Two cities are shown, over a 3-month period. Weekends are highlighted with gray bars. We see that in general, individuals are about half as active on weekends compared to weekdays, and that this pattern is highly predictable. The only two weekday outliers (in mid-June and mid-August) correspond to national holidays.



(a) (b)

Figure 3.2: Distributions of degree (number of neighbors) and edge weight (number of calls per neighbor) for networks formed from datasets in City A and City B, on a log-log scale, follow well-known forms. Specifically, the degree distribution follows the "power law," with exponential cutoff (i.e. it has the functional form $p_k \propto k^{-\alpha} e^{-k/\kappa}$), that was introduced in [10] and has been found in a wide array of social and communication networks (the internet, coauthor networks, social media). The edge weight distribution also exhibits this power-law behavior.

August) correspond to national holidays.)

3.2 Network construction

We can readily construct a network of these users: let each unique user i be a node, and add an edge between i and j whenever i and j have at least 2 calls between them in some period, say a month. This construction follows conventions developed through experiment and described in great depth in previous work (the requirement for two or reciprocated calls e.g. from [56] and a one month period from [35]). In Figure 3.2 we see that the degree distribution generally follows the so-called "power law," with exponential cutoff (i.e. it has the functional form $p_k \propto k^{-\alpha} e^{-k/\kappa}$), and the number of calls per neighbor (sometimes termed the *edge weights*) also follow a power law, matching many previous findings. We also note the distributions are remarkably similar between cities — this consistency of population-level properties is a common trend observed in these type of datasets (e.g. see [11]).

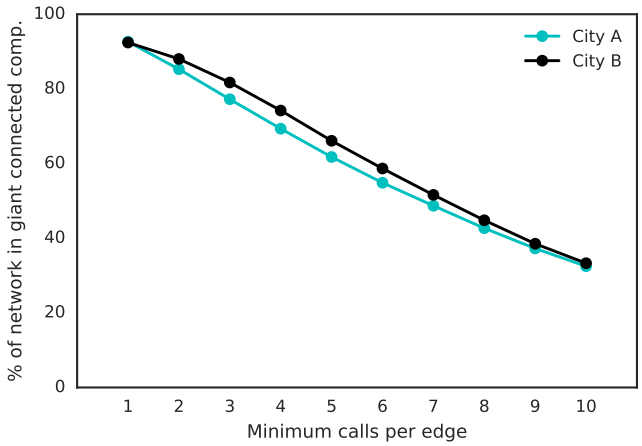


Figure 3.3: **Connectedness of the network.** Choice of the minimum number of calls needed between two individuals in order to place an edge between them affects the connectivity of the network. Depicted is the proportion of the network that is part of the largest connected component, against the minimum number of calls *per month* required for edge connectivity. (We remove any nodes with degree zero.) For low minimum values, we find the majority of the network is part of a single "giant connected component" which constitutes over 85% of the total individuals in the network. As we increase this threshold, the giant connected component shrinks. We will follow previous literature and use small thresholds for edge connection, usually two calls per month as validated in [56].

3.2.1 Large connected component

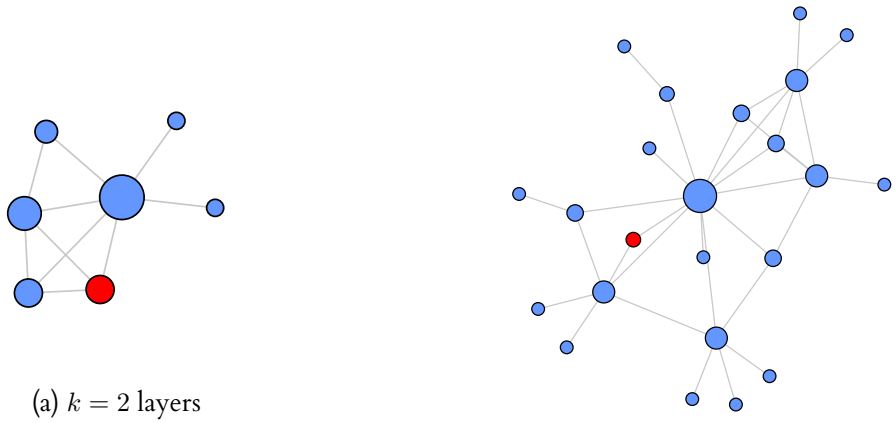
It is not obvious that a particular city-scale network constructed in this way should be fully connected; i.e. that there exists a path from any i to any j. And in fact, we find that it is not *fully* connected, but there does exist a giant connected component (GCC) that makes up about 80-90% of the nodes in the graph. The size of this GCC is dependent on what assumptions we make on network construction, such as how many calls are required to place an edge.

In Figure 3.3 we show this tradeoff in size of the GCC as we increase the minimum edge connectivity threshold. For low minimum values, we find the majority of the network is part of a GCC which constitutes over 85% of the total individuals in the network. As we increase this threshold, the giant connected component shrinks to less than a majority of the population, and the network becomes an archipelago of small- to medium-sized clusters. We will follow previous literature and use small thresholds for edge connection, usually two calls per month, similar to the requirement for reciprocal calls validated in [56].

3.2.2 "Snowball" sampling

We may choose to use samples of the network, either for clearer illustration of a concept or for computational reasons when dealing with large graphs. In general we will sample the network using a "snowball sampling" technique. This allows us to sample a network centered around a particular individual, which in general is conducive to our analysis.

To extract a snowball sample, we will select some node c_0, and collect the set of all individuals $\{c_1^{(i)}\}$ who communicated with c_0, then all individuals $\{c_2^{(i)}\}$ who communicated with any of the $c_1^{(i)}$, etc., to a final set $\{c_k^{(i)}\}$. This creates k "layers" around c_0, and is sometimes referred to as the *ego-k network*. (For example, the ego-1 network of a node c_0 is simply c_0 and those he contacts.)



(a) $k = 2$ layers

(b) $k = 3$ layers

Figure 3.4: **Sampling the network.** We will often use samples of the network, either for clearer illustration of a concept or for computational reasons when dealing with large graphs. In general we will sample the network using a "snowball sampling" technique. Shown are two example snowballs from the City A data: we select a central individual (shown in red), and successively add those he contacts, those they contact, etc. The left network has $k = 2$ layers, the right network has $k = 3$ layers. Nodes are sized according to their degree (# of contacts/friends/neighbors).

In Figure 3.4, we show two examples of snowball samples from City A, with $k = 2$ and $k = 3$. We see that in general the first layer has many interconnections, exhibiting the "triadic closure" described in [22] and others, but that this strongly connected tendency wanes in the second and third layers.

3.3 Conclusion

Mobile phone records provide an unfiltered view into the daily communication patterns of medium-to large-scale populations. The process of transforming raw cell phone record metadata into a network abstraction is well-studied, and so there are many useful previous results to guide our efforts, such as power-law degree distributions, sampling methods, and connectedness.

Persistent Cascades

4

In this chapter we pursue the first part of our problem statement: how can we identify the structure of influence and information spread in a communication network, given only large-scale and content-less data interpersonal communication? Furthermore, what does this tell us about the role and influence of individuals in the network on information spreading? We will initially focus on extracting recurring patterns of interaction as a means of separating meaningful activity from random or inconsequential events; we will then turn to examining the effect this view of the network interactions has on analysis of spreading dynamics and centrality.

4.1 Introduction

4.1.1 Motivation

Our methodology is driven by a desire to better characterize information spread in *temporal networks* using *large-scale metadata*. In general, much research on information spreading assumes a static network and/or employs knowledge of the content of information to develop its model, as described in previous chapters. We find that including temporal patterns of human communication significantly changes our understanding of the network diffusion dynamics and centrality, and that these patterns are evident even without knowledge of content.

Let us review the context and motivation behind these two focuses in more detail.

Large-scale metadata. In many cases, the available data is devoid of content knowledge — that is, we have no Twitter hashtags, or email text, or blog content to guide our understanding, as in such studies as [38, 39, 41] and others. Data in this category we refer to as *communication metadata* and includes datasets like cell phone records or text/SMS messaging. Our methods attempt to bring some knowledge of the true network into the data by extracting persistent structure from these typically noisy datasets.

We focus our attention on mobile phone records, also termed call detail records (CDRs), because they provide a unique opportunity to study the large-scale, unfiltered communication patterns of individuals among their friends. Unfortunately, this breadth of knowledge — in time, space, and demographics — comes at the expense of depth, since we have no information about the purpose or content of communication as we might in social media or email records. Our approach attempts to solve this problem by finding persistent patterns that strongly imply meaningful communication is taking place.

However, although our methodology is created with communication metadata (and specifically CDRs) in mind, by abstracting our focus to rely only the *individuals* and *events* of a network, the methods we describe extend to a wide variety of data and problems: emails, text messaging, product adoption, stock fluctuations, gang activity — in all these cases we have an individual entity (person, stock, gang) which generates some observable event (text, call, price change) that has an effect on other individuals in the network, which we wish to detect. As evidence of this extensibility, we will consider a case study in the final section using email data where content is known.

Temporal networks. Further, we are interested to go beyond a static understanding of our network and better characterize the temporal structure of interactions. Early research in large scale networks typically *aggregated* the observed interactions into a static network — for example, if we saw A communicate with B at least n times in a period of length T, we connect A and B. This threshold n and T is explicitly studied in works like [57], which requires communication to be reciprocated in order to connect two individuals, and [35] which studies the stability of the static network under different period lengths. It is also implicitly assumed in many other studies on this style of data, where the threshold is not always evident from the analysis, and is a sort of unstated hyperparameter.

It is many times enough to simplify a communication network into a static object. However, since information spreading is fundamentally time-dependent in nature, later research introduced the importance of including temporal knowledge. As a simple example, perhaps A communicates with B 10 times in a 3 month period, but all 10 events occurred in the first week and A has not spoken to B since — this gives a very different picture of the network structure that we lose under an aggregated approach. Revelations abound under this new paradigm: [9] showed that contact tends to be heavy-tailed (or "bursty"), [50, 28, 27] described the effect this has on information spread, and [49, 48] showed that temporal consideration reveals new roles of individuals in the spreading process. These and other works are described in more detail in the preceding chapter, but we emphasize it again here to justify our focus on this more nuanced approach.

4.1.2 Approach

We will first take the view that information spreading is by nature a cascading pattern: an individual has a piece of information that he/she spreads to others, who then pass to still others, etc. This eliminates the possibility of loops (since we are only interested in the information-passing edges, and we can assume individuals received the information at the earliest possible call), and creates a rooted, directed, tree structure. This is a well-worn approach: this structure is sometimes called a *minimum spanning temporal tree* (see [25]), it follows from similar assumptions in [19, 59] on information spreading, and has a close analogy to the rich field of epidemic spreading.

Our claim is that observation of similar information spreading structure among similar individuals over a long period of time is a strong indicator of information spread, and reveals new dynamics of communication among the individuals involved. For example, consider an observed pattern where person A calls persons B and C, who then call persons D, E and F, and then we observe this same pattern, or something similar, repeated every few days or weeks.

We term these *persistent cascades*, and claim the pattern leads to two very reasonable claims: (1)

it is more likely that calls in a persistent cascade indicate meaningful social interactions than calls not observed in one, and (2) it is highly likely that persistent cascades correspond with information spread.

4.1.3 Contributions

We first introduce a novel method to detect such recurrent patterns using techniques of inexact tree matching and hierarchical clustering. This differs from existing work in that (1) we are focused on recurring patterns among specific users, not network motifs (e.g. the prevalence of triangles or other structures in the graph, regardless of individual), and (2) we allow for inexact matching (not necessarily isomorphic graphs) to better account for the noisiness of human communication patterns. We then analyze the resulting patterns, termed *persistent cascades*, finding short-duration indicating burstiness, habitual hierarchy in the order that groups communicate within persistent cascades, revealed roles in weekday vs. weekend spreading, and long-term persistence. We justify several simplifying assumptions of our approach by comparing against an exhaustive search, finding that only 2% of all data is affected by our assumptions.

Next, we show the significance of our findings by comparing them against a random network model (specifically, a configuration model with interactions captured as an average rate). We represent this null model both through simulation and analytically. We find that the data exhibits significantly more and larger persistent activity than is evident in a random model. We argue that this result is evidence that the data necessitates a model which can capture the inherent temporal clustering which we are observing in persistent cascades.

We then show the role of members of persistent cascades in information diffusion, borrowing the susceptible-infected-recovered (SIR) model from epidemic spread. We find, through simulation, that members of persistent cascades are more susceptible than non-members, and that this effect is not simply correlated with overall call activity. We give a mathematical argument for why this is so, which illustrates that when information is resistant to spread, persistent cascades provide the necessary repeated exposure to cause spreading, whereas when information can spread freely this effect is masked by the random mixing between non-cascade-members.

Finally, we show that the method is extensible to other communication datasets by applying it to an email dataset. In this case study, we use the publicly available emails released during the government's investigation into Hillary Clinton's use of a private email server, and find that the persistent cascades approach as outlined correctly identifies key staff members, ignores known "noise" in the dataset (such as unlabeled emails), and identifies several interesting persistent email chains.

4.2 Methodology

We now present a methodology for detecting persistent patterns of information spread in large-scale metadata, and present an analysis of findings and results in the mobile phone data.

4.2.1 Defining a cascade

Consider a temporal graph $G = (V, E)$ which represents the communications between users over some large time period $T = [t_{\text{begin}}, t_{\text{end}}]$, such as one month. Let each node $v \in V$ represent a

user who participates in some number of communication events during period T, and let each edge $e \in E$ represent a communication event which we encode as a 4-tuple $e_i = (s_i, d_i, t_i, \delta_i)$ consisting of the initiator (s_i), the receiver (d_i), the time of the event (t_i), and its duration (δ_i).

We define a *time-respecting* path as any sequence of edges $(e_1, e_2, ..., e_k)$ such that for any consecutive pair e_i, e_j in the sequence, we have that $d_i = s_j$ and $t_i + \delta_i \leq t_j$. We define a Δt-*connected* path as a time-respecting path such that $t_k - t_1 \leq \Delta t$. From these definitions, one can construct Δt-connected subgraphs that contain some time-respecting subset of *all* the events within Δt (e.g. [33]).

However, in pursuit of understanding information spread patterns, we make an assumption that the information *originates from a single user*, and every user receives the information at the *earliest possible time*. This implies there is a single in-edge to each user, and creates a rooted, directed tree structure. Intuitively, this shifts focus from the structure of the call patterns to the structure of the information spread, since we will only capture the first occurrence of "information" being passed.

Formally, this assumption leads to the construction of a rooted, directed, Δt-connected tree which we term a *cascade*. This term, and its construction, follows closely that in [59]. (These structures are also called the *minimum spanning temporal tree* and [25] gives some efficient algorithms for extracting them from a network both with and without edge weights.)

Denote a cascade with root r as C_r, denote the set of all cascades for root r with maximum time interval Δt and total time period T as $\mathcal{C}_r(T, \Delta t)$, and use superscripts as necessary to distinguish multiple cascades with the same root. For example, we might have the set of all cascades for some root a:

$$\mathcal{C}_a(T = 1 \text{ mo},\ \Delta t = 24 \text{ hrs}) = \{C_a^1, C_a^2, C_a^3\} \tag{4.1}$$

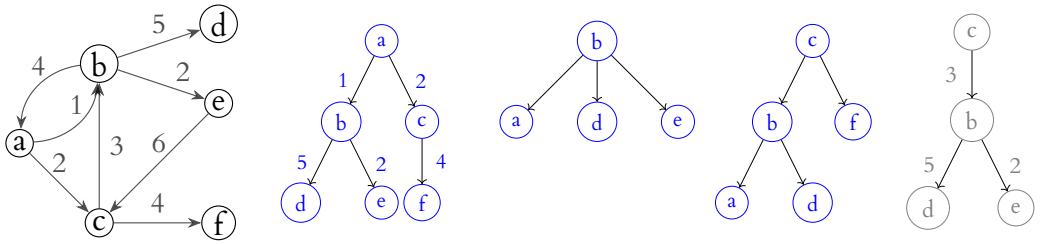
Note we require that the intervals not overlap: i.e. no calls from C_a^1 can also be in C_a^2, etc.

An example of cascade construction from a network with all temporal information is shown in Figure 4.1, and the algorithm for extracting one for a given root r in a time interval Δt is described in Algorithm 1.

We make two notes about this definition before proceeding. First, notice that for any cascade, its subtrees are also (usually) cascades. For example, in Figure 4.1, note that the cascade with root a has a subtree corresponding to the cascade with root b. This is by design: we do not know the true information originator, so we should consider each possible "root" user in his or her own right in the analysis of persistence that follows. We can afford such an exhaustive search because we have already greatly reduced the search space by requiring a cascade to have at minimum three members and roots to have made enough calls to make the later persistence analysis possible. Also, although the number of trees theoretically grows exponentially with the graph size $|V|$, the temporal requirement greatly lowers this bound, and the minimum activity constraints just mentioned lower it again, so the problem is highly tractable for even large datasets. (Typical runtime for extracting all cascades for a network of 300,000 nodes over the course of a month is around 10 minutes.)

Second, consider a root node who is very consistent in the users he calls, but these users are then subsequently very *in*consistent. Then the overall cascades generated from this root will be dissimilar, and therefore ignored in the subsequent analysis. This is again by design: we are concerned with persistent information *spread*, not just consistent calls from a particular user to certain friends. Cascades that are only similar in the first level may not indicate the root is a strong originator of

Figure 4.1: **Simplified illustration of cascade extraction from a temporal graph.** For clarity, we examine a network with only 6 nodes. **(a)** Full temporal information ($\Delta t = 6$ units, times depicted on edges). **(b)** Three valid cascades given this temporal snapshot. Note that there is no time ordering of children within a cascade. **(c)** Invalid cascade because: (c-b-e) is not a time-connected path, and missing the edge (c-f).



Algorithm 1 Cascade construction

Input: r, $G(\Delta t)$, t_{begin}, t_{end}
Output: C_r
  Initialize (global) $C_r = \{r\}$
  Run SUBTREE(r, t_{begin})
  procedure SUBTREE(v, τ)
    edges $\leftarrow e_i \in G(\Delta t)$: $s_i = v$, and $\tau < t_i < t_{\text{end}}$
    for $e_i = (v, d_i, t_i)$ in edges **do**
      if $\exists n \in C_r$ s.t. $n = d_i$ **then**
        if $\tau_n \geq t_i$ **then**
          delete n
        else
          continue
      add (edge) e_i and (node) d_i to global C_r
      assign $\tau_i = t_i$
      SUBTREE(d_i, t_i)
    return

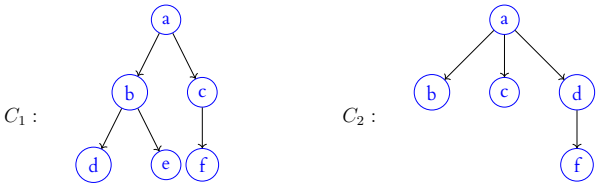
information. However, we will address this concern in a later section and show the minimal impact of this assumption on the analysis.

4.2.2 Measuring similarity

Measuring similarity of cascades, as defined, is now an inexact tree matching problem. We now define two similarity measures, both standard in the literature: normalized tree edit distance, and reach set similarity (measured with a Jaccard index).

Tree edit distance

Edit distance is the process of counting the minimum number of insertions, deletions, or mutations required to transform one string into another. One can extend this concept to trees. Denote the tree edit distance between two trees (or cascades) C_1 and C_2 as $\text{TED}(C_1, C_2)$, which maps two cascades to a nonnegative integer. As an example, consider the following two trees:



To change C_1 into C_2, we can delete d and e, mutate c into d, and add c again, giving $\mathrm{TED}(C_1, C_2) = 4$ (note this is the same to change C_2 into C_1).

A canonical algorithm for computing this distance is due to Zhang and Shasha ([75]), which we implement with the `zss` package available at `https://github.com/timtadh/zhang-shasha`. We can now define a similarity measure using this distance as follows.

Definition 4.2.1. *Tree Edit Distance similarity.* Define the normalized tree edit similarity as

$$s_{\mathrm{TED}}(C_1, C_2) \stackrel{\text{def}}{=} 1 - \frac{2 \cdot \mathrm{TED}(C_1, C_2)}{|C_1| + |C_2| + \mathrm{TED}(C_1, C_2)}. \tag{4.2}$$

and note s_{TED} lies on $[0, 1]$.

This definition is due to [42], who also prove that the corresponding distance metric $1 - s_{\mathrm{NTED}}$ meets the triangle inequality. Note we make every edit operation unit cost.

Using the example trees above, we now compute $s_{\mathrm{TED}} = 1 - \frac{2 \cdot 4}{6+5+4} = \frac{7}{15} \approx 0.47$.

Reach set

Consider the un-ordered set of all nodes in a tree. For a cascade, this corresponds to all users who the root reached during the time period Δt, and potentially received some information. We term this the *reach set* of a cascade (similar to concepts in [59, 19]).

A simple first approximation of the similarity of two cascades is by comparing their reach sets. Let $R(C_i)$ denote the reach set of a cascade C_i. Now, given two cascades C_1 and C_2, define the similarity measure s_{RS} as the Jaccard index of the two reach sets, that is

Definition 4.2.2. *Reach Set similarity.* Given two cascades C_1 and C_2, and their reach sets $R(C_1)$ and $R(C_2)$, define

$$s_{\mathrm{RS}}(C_1, C_2) \stackrel{\text{def}}{=} \frac{|R(C_1) \cap R(C_2)|}{|R(C_1) \cup R(C_2)|}. \tag{4.3}$$

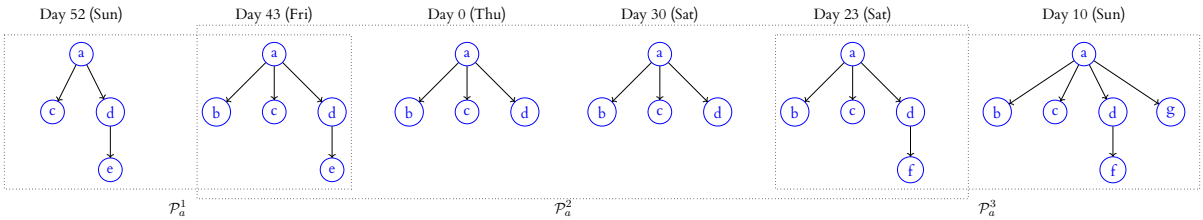
and note s_{RS} lies on $[0, 1]$.

Continuing with the previous example, we have $s_{\mathrm{RS}}(C_1, C_2) = \frac{5}{6} \approx 0.83$.

4.2.3 Persistence

We now would like to group cascades together which all share some minimum pairwise similarity, and so are in a relaxed (but well-defined) sense the "same cascade." This group now represents various incarnations of some fundamental communication structure. We call these groups *persistence classes*, and the elements of each group *persistent cascades*, and they are the main object of our analysis.

Figure 4.2: **Grouping similar cascades into persistence classes.** Shown is an actual set of persistent cascades for a root a over a 60-day period. Six persistent cascades are shown, each from temporal subgraphs with $\Delta t = 24$ hours. Dotted rectangles depict the persistence class groupings. We see a clear set of "core friends" (nodes b, c, d), and slight variations incorporating other groups. We also see the overlap that occurs when a cascade appears to fit in multiple classes. Labeled above each cascade is the day of the week.



Definition 4.2.3. *Persistence class.* Define the i-th persistence class of root r, similarity threshold ℓ in time period T over intervals Δt, as the set

$$\mathcal{P}_r^i(\ell, T, \Delta t) = \left\{ C_r^1, C_r^2 \in \mathcal{C}_r(T, \Delta t) \ : \ s_*(C_r^1, C_r^2) \geq \ell \right\} \tag{4.4}$$

and the collection of all persistence classes for a particular root as $\mathcal{P}_r(\ell, T, \Delta t)$.

Definition 4.2.4. *Persistent cascade.* Define a persistent cascade as any cascade C_r^i such that $C_r^i \in \mathcal{P}_r(\cdot)$, for some r.

Note we may also choose to ignore any persistence classes below a certain size. The minimum size is 2 by construction, but we may decide based on the parameters T and Δt that a minimum size of 3 or more is appropriate.

To find these classes, our definition and Eq. (4.4) leads us directly to an agglomerative clustering approach with complete-linkage — that is, define the similarity between two clusters U and V as

$$s(U, V) = \min s_*(U_i, V_j), \forall i \in U, \forall j \in V$$

where U_i, V_j represent cascades within U and V. Then the clusters at iteration k, such that every pairwise similarity within the cluster is $\geq s_k$, represent persistence classes with $\ell = s_k$.

However, this assumes that each cascade falls uniquely into one class, which we can imagine is not always true: a spreading pattern among work friends may overlap with the pattern among social friends, and there may be cascades that are not clearly in one class or the other.

So we instead adopt a graph-theoretic interpretation of the complete-linkage approach: represent each data point (cascade) as a vertex in a graph $H(s_k)$ such that each any two vertices with similarity $\geq s_k$ are connected. Then the clusters at iteration k correspond to the maximal completely connected subgraphs in H, also known as the maximal cliques.

Now, applying this technique, consider the collection of persistence classes $\mathcal{P}_a$ depicted in Figure 4.2, taken from City A. Here, we see a core pattern consisting of root a calling b, c, and d, captured in $\mathcal{P}_a^2$. Then, we see two variations on this core structure: $\mathcal{P}_a^1$ which incorporates e, and $\mathcal{P}_a^3$ which incorporates f and g. Since they are mostly weekend calls, we might easily imagine this being a core group of social friends, with variations possibly for family or work acquaintances.

We make two notes on our methodology of identifying persistence. First, we are only doing pairwise comparison between cascades which share a root node, leaving out groupings such as dif-

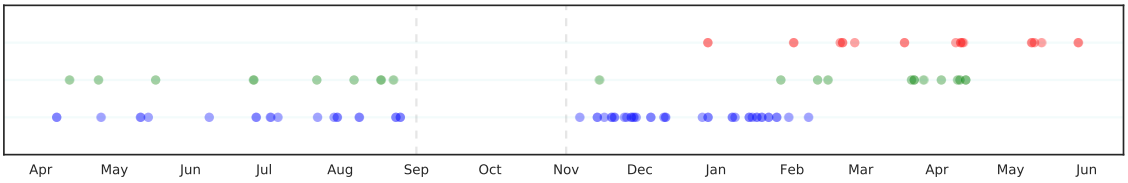


Figure 4.3: **Examples of call activity within a persistence class.** Three example persistent patterns are shown, from City A. Each line corresponds to a persistence class, and each dot corresponds to the occurrence of a cascade within that class (i.e. each dot represents multiple calls). The timelines suggest interpretations of information spreading: for example, the third line appears to be friends whose communication crescendos in Dec-Jan and then sharply drops off (possibly event planning); the first line appears to be a group forming (possibly post-holidays).

ferent initiators who disseminate information to the same people. It has the effect of maintaining focus on analysis of the roots, instead of the broader role or persistence of a cascade pattern itself. Second, note that it is conceivable that unrelated call events could happen consistently in the same order among the same people and get picked up mistakenly as persistent. Not knowing the actual content of the calls, we can only say that persistence, as defined, indicates a very high likelihood of information spreading.

4.3 Findings in the data

4.3.1 Examples

Before proceeding to any thorough analysis of this method's findings, let us take a look at a few example patterns found in the data.

First, we find long-term persistence on the scale of months to a year (the entire length of the available dataset). In Figure 4.3 we show three example long-term patterns in the data (City A). Each line corresponds to a persistence class, and each dot corresponds to the occurrence of a cascade within that class (i.e. each dot represents multiple calls). The timelines suggest interpretations of information spreading: for example, the third line appears to be friends whose communication crescendos in Dec-Jan and then sharply drops off (possibly event planning); the first line appears to be a group forming (possibly post-holidays).

These long-time-scale classes are typically only 3-4 users. However, we see large repeated patterns of 10 or more users at the time scale of months. For example, Figure 4.4 shows two large patterns: one of 10 distinct individuals (1 month period) and one of 9 distinct individuals (2-month period).

Both the long-term persistence and the large cascades are remarkable, and both enable a strong claim for meaningful communication and likely information spread. We now turn to more thorough analysis of these patterns.

4.3.2 Size and connectedness of the persistent subnetwork

We find that most persistent cascades (e.g. 71% of the sample in City A) are among 3 contacts (the minimum necessary to constitute a cascade). The largest persistent structures involve 20-30 people (for example, in City A, we find a persistent class with cascades of 37-39 users, but note persistent

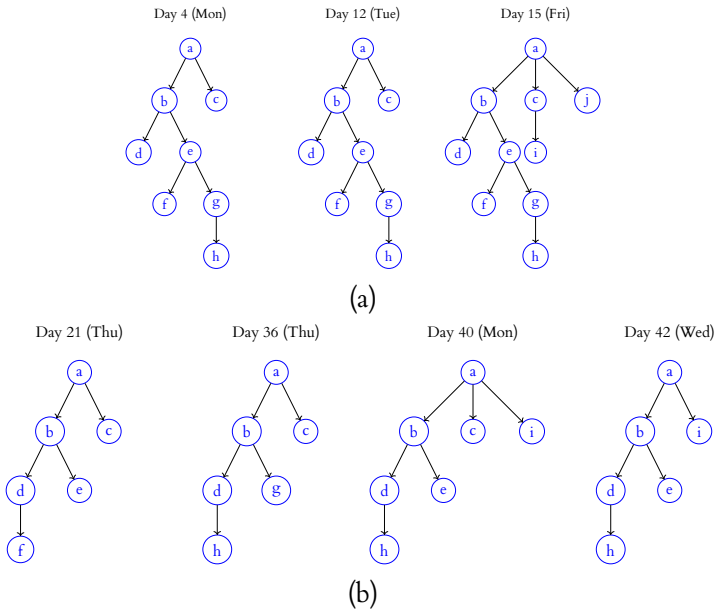


Figure 4.4: **Examples of large persistent patterns.** Persistence class (from the data) of (a) 10 distinct individuals, over 1 month; and (b) 9 distinct individuals, over 2 months.

cascades with more than 6 people constitute less than 1% of the sample).

We also find that calls within a persistent cascade only account for about 10% of the entire dataset for a given number of months. (For example, with $\ell = 0.8$, the average in City A is 9.8% of calls over a 1 year period, and with $\ell = 0.7$, the average rises to 15.1% of calls.) On the other hand, over 20% of the network is involved in a persistent cascade (on average), which indicates that most of the non-persistent call activity is high-volume. This leads us to wonder disconnected the persistent cascades are from each other, and from the rest of the network. How sparse is this phenomenon?

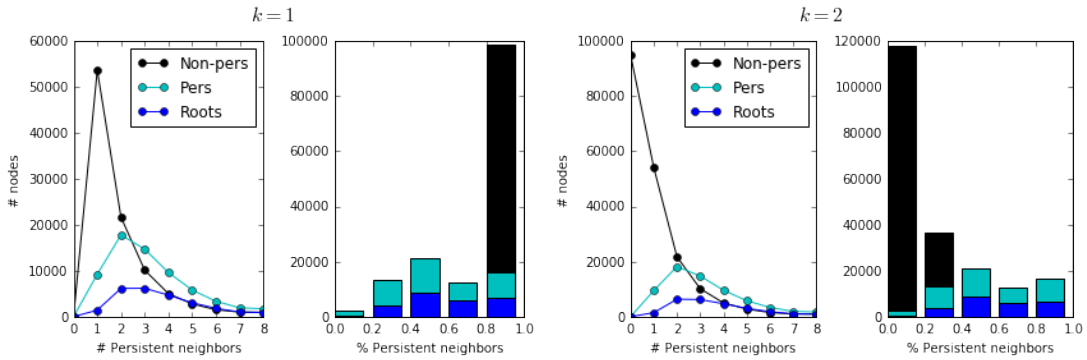
One way to analyze this connectedness is to consider the induced network obtained by only retaining persistent individuals (i.e. nodes that are a member of a persistent cascade). We can then "loosen" the constraint to include individuals that have made (non-persistent) calls to cascade members ($k = 1$ hop away from a cascade), or loosen again to $k = 2$ hops away, etc. We will denote the resulting graph as the *k-connected persistent subnetwork*.

We find that with $k = 0$, the persistent subnetwork is highly disconnected: we only retain about 20% of nodes (as stated before), which are divided into over 12,000 subgraphs of no more than 180 nodes, and with an average size around 5. (These figures are approximations of results culled from several months in both City A and B. Results are very consistent month to month.)

By contrast, with $k = 1$, a giant component suddenly emerges. The persistent subnetwork now contains about half (average 50.6%) of the full network. With $k = 2$, the subnetwork contains nearly 80% of the full network, and about 25.3% of nodes belonging to a persistent cascade. And so with $k = 2$, we have nearly recovered the original network, indicating that despite persistent cascades being an uncommon event, nearly the entire population is within 1 or 2 "hops" from these persistent communication structures.

Figure 4.5 shows the distributions of persistent degree (number of persistent neighbors) and the proportion of persistent neighbors, as we relax the connectedness constraint from $k = 1$ to 2. We note the large jump in the distribution of neighbors for the non-cascade members: non-cascade

Figure 4.5: **Connectedness of the persistent cascade network.** Members of persistent cascades constitute only about 20% of the entire network. However, if we include non-cascade-members who are only $k = 1$ "hop" away, a giant component emerges that includes over 50% of the network. By including those at most $k = 2$ hops away from a cascade member, the giant component constitutes nearly 80% of the full network. The charts in this figure depict the change in distribution of persistent degree (number of persistent neighbors) and the proportion of neighbors which are persistent, as we relax the connectedness constraint from nodes being at most $k = 1$ hop away from a persistent cascade to $k = 2$ hops away. (The distributions are separated by node type: persistent cascade members ("pers"), persistent cascade roots ("roots"), and non-cascade members ("non-pers").)This indicates that despite persistent cascades being an uncommon event, nearly the entire population is within 1 or 2 "hops" from these persistent communication structures.



members link to graph at $k = 1$ with a single persistent contact, but by $k = 2$ this is completely overwhelmed by non-cascade members with large non-persistent degree counts.

4.3.3 Cascade time and duration

Figure 4.6 shows the distribution of first and last call times in a (persistent) cascade, and the resulting distribution of cascade durations. This was done on a random sample of 10^4 root users in all 3 cities over a period of 1 month. The call times follow the expected workday pattern of a morning peak around 9-10 a.m., and another peak before nightfall around 8 p.m.

We also see from the right plot in the figure that most persistent cascades are very short — usually everyone is called within an hour — which echoes earlier work on the burstiness of communication. There is also a large group of cascades with durations between 5-10 hours, suggesting information spread is either very rapid, or unfolding over a morning or afternoon, but rarely lasting all day.

This evident short attention span in the cascades led us to avoid analysis of longer time periods (48, 72 hours or longer). Longer time periods also may decrease the possibility of the cascade representing information spread. It may be fruitful to consider a shorter interval, such as 12 hours, to attempt to capture morning vs. evening cascading action (e.g. work vs. social), or a sliding window approach. We leave exploration to future work.

4.3.4 Similarity measure correlation and habitual hierarchy

We now examine the relationship between the two similarity measures introduced before: tree edit distance (TED) and reach set (RS). Based on a random sample of 5×10^4 pairs of cascades from City B, the measures have a Pearson correlation coefficient of $\rho = 0.91$. (We have chosen a single city for illustration, but this coefficient is similarly high for all cities: in City A it is 0.90 and in City C it is 0.93.)

Figure 4.6: **Persistent cascade activity is circadian and "bursty."** Shown is the distribution of (left) call times and (right) duration among persistent cascades. The left plot shows the distribution of times for the *first* (i.e. earliest) and *last* (latest) calls in a cascade. These follow the well-known 24-hour circadian rhythm of human activity. The right plot shows the resulting distribution of total duration of a cascade, and illustrates that most persistent cascades happen rapidly, over the course of 1-2 hours.

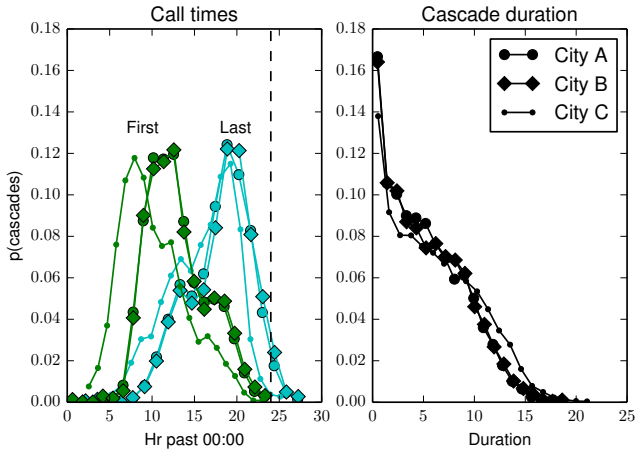


Figure 4.7: **Habitual hierarchy of information spread.** This heat map depicts the correlation between tree edit distance (NTED) and reach set (RS) metrics on a sample of 5×10^4 pairs of cascades with the same root (over approximately 10^4 different roots). Recall that NTED measures *structural* similarity, while RS measures similarity of *individuals* (regardless of structure). The number of pairs where $s_{\text{RS}}(x, y) = 1.0$ but $s_{\text{TED}}(x, y) < 1.0$ is surprisingly small — only about 0.5% of the sample — this suggests that cascades among the same users tend to occur in the same order. Note: the colors are log-scaled for visualization. The Pearson correlation coefficient of this relationship is $\rho = 0.91$. We have chosen a single city (City B) for illustration, but this coefficient is similarly high for all cities: in City A it is 0.90 and in City C it is 0.93.

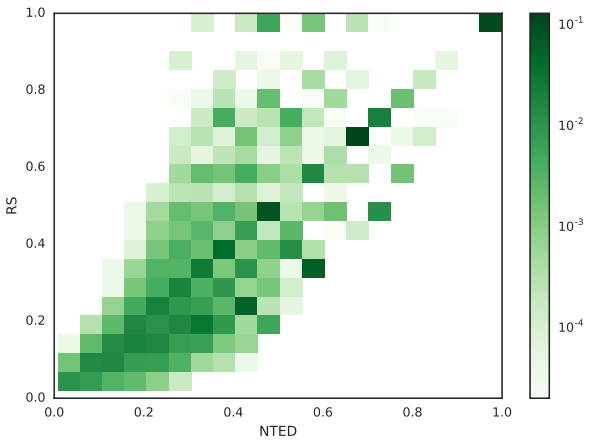


Table 4.1: **Persistent cascades reveal a tendency for weekend or weekday information spreading.** This table gives the percentage of individuals with call activity only on weekends, only on weekdays, or some mix of both. The top half of the table shows this percentage without any analysis of persistence, and simply illustrates that most people ($> 99\%$) make calls throughout the week. The bottom half of the table shows the percentage when we consider only calls *within a persistent cascade*, and reveals two new groups emerging: a group that is only involved in persistent communication on weekdays, and another that is exclusive to weekends. These roles in information spread are not evident without the persistence analysis.

Cascade type	Dataset	Only Weekend	Mix	Only Weekday
All	City A	<1%	99.2%	<1%
	City B	<1%	99.4%	<1%
	City C	<1%	99.8%	<1%
Persistent	City A	1.8%	82.5%	15.6%
	City B	2.6%	83.8%	12.9%
	City C	2.5%	84.2%	13.3%

Note. "Only" weekend/weekday signifies at least 90% of events. Fridays designated as the weekend.

It is possibly surprising that the correlation is so high. For example, consider the group of cascades with s_{RS} of 1.0 and s_{TED} less than 1.0, and note that this group represents less than 0.5% of the sample. This shows that when two cascades involve the same people, they nearly always involve them in *the same order*. (And if not, we would see more pairs with dissimilar structure (low TED) but similar reached users (high RS).) This observation suggests there is a *habitual hierarchy* of information spread among social contacts.

Correlation used for performance speedup. As an important aside, the main performance bottleneck in computing all persistence classes for a particular dataset is the TED measure. However, the correlation between measures shows RS is a close approximation in most cases. It is also much easier to compute. So, if computing $\mathcal{P}_*$ under both measures, one can compute RS similarity first, and only compute TED similarity as necessary for s_{RS} above some low threshold. Finally, since we are only considering classes with the same root, the clustering step is parallelizable. Using these speedups, we could build all persistence classes for a single city, with both similarity measures and $T = 1$ month, in about 30 minutes.

4.3.5 Tendency for weekday vs. weekend information spread

Consider the set of all cascades (not necessarily persistent) that a given (root) user initiates in the course of some period T, for example a month. Since most active users tend to make some calls every day, we might expect these cascades to be evenly distributed over each day of the week.

In Table 4.1 we examine all cascade initiators in each city with at least one persistent class and at least 3 persistent cascades. If we consider all cascades of this group (not just persistent ones), we see that there is an even mix throughout the week, as expected: nearly all users are generating cascades (that is, making calls to multiple people) on some mix of both weekend and weekdays. Very few users ($< 1\%$) are active exclusively on weekdays and/or weekends.

However, if we examine only *persistent* cascades, two new groups emerge: a large portion of root users who only initiate persistent cascades on weekdays, and a slightly smaller portion who only initiate on weekends. These two extremes constitute over 15% of all root users, while the same

extremes measured in all cascades are $< 1\%$. This is a complement to the observation that people have different mobility similarities to weekend and weekday contacts, in [68].

In other words, for these two groups, although they make calls throughout the week, their role in spreading information appears to be specialized: their only persistent patterns of information spread happen during either weekday (i.e., work week) hours or weekend hours, but not both. Their other communication is sporadic, or random, and one might easily conclude, not meaningful.

4.3.6 Long-term persistence

Now we turn our attention to observations of the persistent structures over longer periods of time ($T > 1$ month). One property we expect to see emerge is the idea of *long-term persistence*. Specifically, if the persistent classes represent the fundamental underlying communication structure of the network, we expect them to persist over long periods of time — that is, user's should continue to generate cascades which "fit" into existing classes.

First, in Figure 4.8(a), note the decline in the distribution of persistent classes as we increase the minimum size requirement (i.e., for a user a, enforce that $|\mathcal{P}_a^i(\cdot)| \geq k$, for all i, and increase $k = 2, 3, 4, ...$). This is an expected effect of increasing requirements within a finite time. For a minimum size of 4 cascades, only about a tenth of the population has even one persistent class.

If there were no long-term persistence of these classes, then we would see no class growth over time, and the distributions of persistent classes would decline as we increase their minimum size requirement, regardless of the time period.

However, in Figure 4.8(b), the opposite happens. As we increase the time period and the minimum size requirement, the distribution of persistent classes increases somewhat and stays generally the same, especially for the 90% of the population with 3 or fewer classes. This implies that our intuition is correct, and many (if not most) of the persistent classes continue to grow as time goes on.

We can also be more precise by checking, for example, how many specific users with 1 persistent class after 1 month, still have 1 persistent class after 2, 3, 4, and 5 months, etc. We find that about 65% of users with a single persistent class (of size ≥ 5) after 3 months of observation, will still have a single persistent class (now of size ≥ 6) after 4 months of observation. And about 71% with a single class after 4 months will again have a single class (now of size ≥ 7) after 5 months. This is remarkable consistency, and suggests a strong predictability of calling habits.

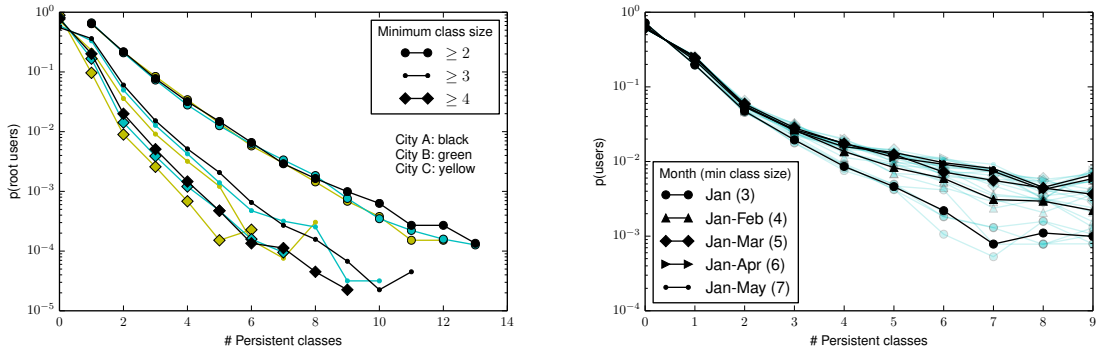
4.3.7 Implementing exhaustive search

There are many families of patterns that we might imagine are present in the data, but that the construction of our algorithm as defined will not "pick up." For example:

1. A manager who makes a morning call to his/her secretary each day before initiating high-importance cascades later in the day. The secretary's subsequent high activity but non-persistent calls will mask any activity from the manager.

2. A root who has two recurring patterns of communication, but they are large and intermingled throughout the day, so even tree matching with relatively low thresholds of similarity will not detect recurrence.

Figure 4.8: **Long-term persistence and predictability.** We find that individuals in persistent cascades tend to continue communicating in the same patterns, and do not generate more and more new patterns over time. To illustrate this, we compare the distribution of the proportion of users with different numbers of persistence classes — first (a) we increase the minimum required cascades in a class, but fix T. We find that, unsurprisingly, the number of people with a given number of classes decrease as we increase this threshold. By contrast, in (b) we again increase the requirement for persistence (from 3 to 7 cascades in the class) and also increase T from 1 to 5 months. Now the distributions are nearly identical, especially for users with 0-2 classes (who constitute over 90% of the sample), suggesting long-term persistence and bounded social capacity. (The black plot depicts the average over 5 samples of 5×10^3 random users; samples depicted in light green.)



3. A working partygoer who makes work-related reoccurring calls in the morning, but social-related calls in the evening. Using a day-long period, we will attempt to group these into a single cascade and likely miss many patterns.

4. A supervisor who makes a call to one project leader in the morning, and another in the evening, with various midday patterns occurring on a more random (but high-activity) basis. We will miss the morning-evening pattern because it is always split by unrelated intermediate calls.

And there are certainly others. We can solve many of these dilemmas with slight modifications to the algorithm: for example, for (1) we can try to eliminate some of the secretary's noise by only keeping high recurring calls, for (2) we can loosen our threshold of similarity to detect these large intermingled patterns and do some post-hoc analysis to suss out the two classes, for (3) we can introduce sliding time windows (instead of static disjoint time periods) to try to maximize stable classes, and for (4) we can try randomly splitting each period and choose the splits that maximize the resulting classes' similarity, or size, etc. (similar to pruning a decision tree).

However, it would be helpful to understand whether these patterns are a large concern in the data or merely rare events. To do this we will explore the only modification that will allow us to detect *any* possible pattern: *exhaustive search*.

Specifically, we will proceed as before and extract each minimum spanning temporal tree from every possible root individual in the network, over the interval of a day, for some period T like a month. However, we will then split these cascades into all possible time-respecting, temporally connected, subtrees rooted at the original root. Finally, we will repeat the clustering analysis of the original algorithm, but now looking for clusters of subtrees, with the constraint that subtrees cannot be clustered together with subtrees in the same interval. (This method still makes the assumption that patterns do not occur across days.)

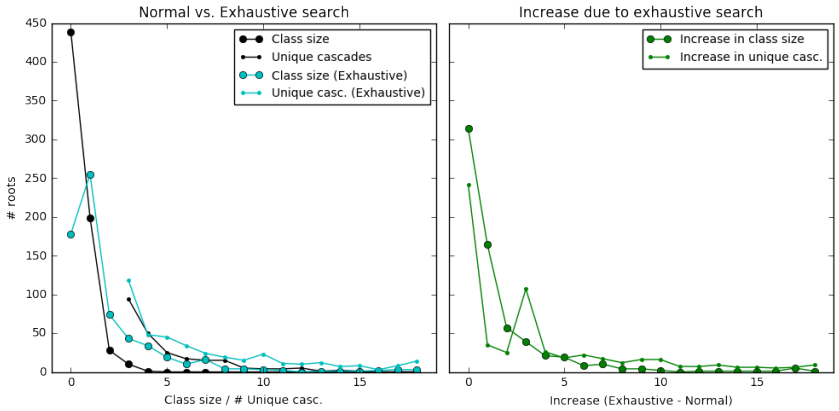


Figure 4.9: Left plot depicts the distribution of class size and unique cascades per root user using the unmodified method (black) and exhaustive search (cyan). Right plot depicts the increase, per root, in these statistics by doing exhaustive search. We notice there is a nonnegative change for every user, but that the majority of the sample sees no change.

This approach is prohibitively expensive in general: the number of temporal subtrees of a cascade of size N is bounded above by 2^{N-1}. Consider a root with 5 cascades, each of about 10 members: this leads to upwards of $\frac{1}{2}(5 \times 2^{10})^2 \approx 12.5$ million pairwise comparisons for the similarity matrix. However, it is tractable on large samples in practice: the vast majority of full cascades are only 3-6 members (as described before), and since we are only doing clustering between days this reduces the computational effort again.

The value of the exhaustive search extension to the algorithm will be to illuminate, in a sense, how big of a problem we might have. That is, since the exhaustive search method will detect the "missing patterns" we are concerned about, we can compare the results to the ones from the previous section and get a sense for how rare these problematic patterns are.

We first implement both the original and the exhaustive search method on multiple samples of 1,000 individuals in the network and compare the distributions of class size and number of unique cascades per root. Results are shown in Figure 4.9. We find that although the exhaustive search finds a large number of new patterns (approximately a 30-40% increase in total cascades), these are in general smaller cascades. We also find that although exhaustive search results, by construction, in a nonnegative change in both class size and total unique cascades for every user, the majority of the population sees no or negligible change.

We can also investigate the number of new call events are identified by this exhaustive search as "persistent" calls. Figure 4.10 shows the percentage of all calls in the sample that are involved in a persistent cascade event for both the original and exhaustive search method. We see a small increase, from about 9-10% to just over 12% of calls. On one hand, this mirrors the 30-40% increase we saw in the distributions of class sizes and unique cascades, but on a population level is a relatively mild change.

In conclusion, although the problematic patterns we imagined are present in the data, they contribute to only about 2-3% of all calls and are arguably not a serious shortcoming to the original method.

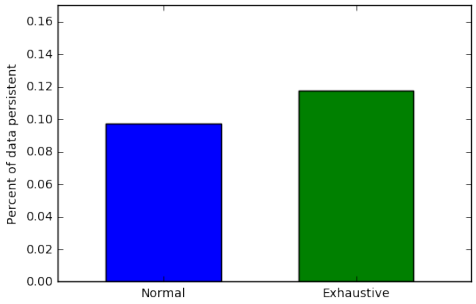


Figure 4.10: **Increase in persistent cascade detection through exhaustive search.** Depicted is the percentage of the data involved in persistent communication, using the original algorithm vs. exhaustive search. This shows that only about 2-3% of the data is involved in persistent activity that we do not detect with our algorithm as proposed.

4.3.8 Discussion

We have now introduced a novel methodology for extracting recurring patterns of information spread, termed persistent cascades, from raw communication metadata, and analyzed the resulting patterns in three city-scale mobile phone datasets. We found the persistent cascades are present on long time scales of months to a year, and found examples of surprisingly large, recurrent structure on the scale of months. We found the patterns tend to be short in duration (the majority last less than 3 hours), which indicates a short attention span in spreading information and echoes previous research in the "burstiness" of human communication. The individuals in a persistent cascade exhibit a habitual hierarchy, in the sense that when the same individuals communicate, they do so in the same order. We also found that our analysis reveals two new groups of individuals who have exclusive roles of information spreading on either weekends or weekdays. Individuals tend to generate more and more instances of the same pattern, and do not create new patterns, indicating predictability of communication. Lastly, we justified several of our simplifying assumptions by comparing our results against those obtained through an exhaustive search, finding that only 2% of the data is affected by our assumptions.

4.4 Comparison to a random model

Motivation. To quantify the significance of the observed patterns in the data, we would like to compare the distribution of their occurrence against some type of null model. Specifically, given a random network with a degree distribution matching the real network, and with average inter-individual call event rates also matching the real data, but without any of the temporal clustering or mutually influencing effects we hypothesize are present in the data, what is the probability of a cascade of s users occurring n times in a month?

We expect to find that in such a random network, recurring cascades of 3 and even 4 individuals still occur with some regularity (2-3 times a month) by sheer chance. However, we also expect that the likelihood of any of the larger cascades, or more frequently occurring cascade patterns, is extremely low. Overall, the distributions of persistent cascade size or persistent class size will be significantly different in the data from the random graph model.

To this end we will adapt techniques from the rich fields of *percolation theory* and *epidemic spread-*

ing. We are concerned with the dynamics of some contagion through a population with network structure. In our application, the contagion is information, the initial infected population are the cascade roots, and the outbreak is the cascade itself. Further, we may assume that the probability of "infection" is only dependent on the rate of interaction between individuals (not on the infectivity of some disease) — in this way we keep focus on the size and structure of the cascade.

Possibly the simplest model in this vein is the susceptible-infected or *SI model*, which considers a population of susceptible individuals S and infected individuals I, who are fully mixing in continuous time (i.e. no network constraint), and with a parameter β representing the probability an infected individual will infect a susceptible individual per unit time. We might think to apply this to our problem by setting a single individual as "infected" (possessing some information), with β representing the probability this seed will interact with his neighbors (e.g. the population average rate of communication). Now let $i(t)$ be the fraction of the population infected at time t, which in our case would be $1/N$, and we have the classic ordinary differential equation and its solution

$$\frac{\mathrm{d}i(t)}{\mathrm{d}t} = \beta i(t)(1 - i(t)) \;\Rightarrow\; i(t) = \frac{i(0)e^{\beta t}}{1 + i(0)(e^{\beta t} - 1)} \tag{4.5}$$

also known as the *logistic equation*, or S-curve. We might approximate the size of an information cascade by simply calculating $i(t)$ after some small time step representing, say, 24 hours.

However, a model like this falls short in modeling our problem for two main reasons: (1) it cannot take into account network effects (other than something like average degree), and (2) it models expected change at a population-level, not the discrete probability of particular outbreak sizes or time periods. As a result, we will introduce some more sophisticated machinery that will allow us to exactly model the probability distribution of recurring cascades, within a random graph model that allows us to closely match the degree distribution and interaction rates of the real network.

Findings. We instead extend the methods in [53, 52, 47] to precisely describe the probability of a particular outbreak (i.e. cascade), and subsequently the probability of its recurrence (i.e. persistence). We will introduce the methods, then compare the analytic distribution to one obtained through simulation, and against the distribution found in the data using the algorithm defined in the previous sections.

We will first see that this analytical form closely matches that of simulation. Second, and most crucially, we find that despite mirroring our real network's degree distribution and pairwise average rates of interaction, the distribution of persistence in the real network is significantly heavier-tailed than the random model (both simulated and analytic).

4.4.1 Simulation model

We first propose a simulation method to test our hypothesis, and later show that this can also be done analytically (with only minor approximations).

Network structure

We will first mimic the structure of the real network by creating a random network with a matching degree distribution. Specifically, given an observed degree distribution $\bar{p}_k$ in the data (i.e. $\bar{p}_k$ is the

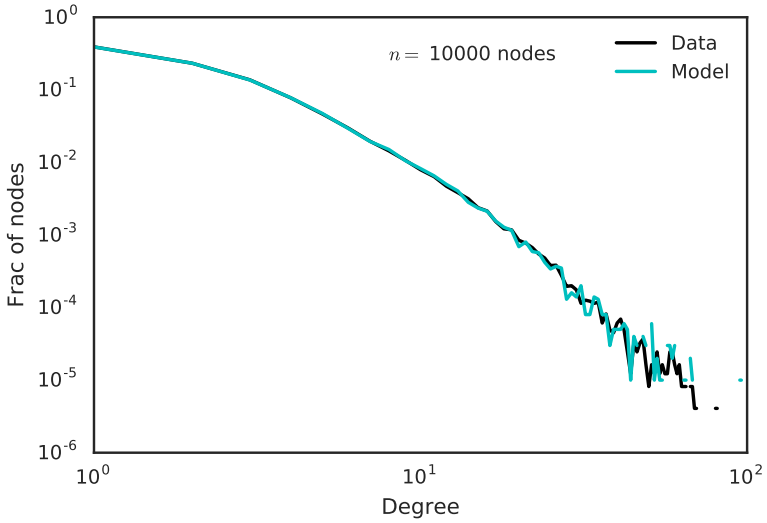


Figure 4.11: Degree distribution comparison of the actual data and a synthetic network generated using the same degree distribution, on 100,000 nodes.

proportion of nodes with degree k, which we construct from the static network over some observed period, and use the normalized histogram of the resulting first-neighbor degree distribution), we seek to generate a graph at random with an identical (or nearly identical) degree distribution p_k. We can accomplish by using the technique of the *configuration model*, which considers the family of all graphs $G(p_k)$ with degree distribution p_k, and is able to sample a graph at random from this family, $G(p_k)_i$.

The configuration model has the benefit of capturing the network structure more closely than, for example, an Erdös-Renyí (ER) model. In the ER model, we generate a network of n nodes such that the probability of any two nodes being connected is independent, with probability λ/n. Then the expected degree is simply λ, for large n. This is a highly tractable model. However, it is not a very accurate depiction of a real communication network: one compelling reason is that it is easy to show the resulting degree distribution is Poisson, whereas we know our network has closer to power-law behavior. The configuration model, by contrast, allows us to specify this distribution exactly to match the data, and as we will see, still provides a tractable framework.

Interaction rates

Second, we will mimic the interactions on each edge in the network by generating random events (simple point process) with a matching *average* rate. These rates, in the data, follow some distribution $\bar{P}(r)$. We fit a Gamma distribution to this to infer a distribution $P(r)$. We use the following form for the Gamma distribution

$$\text{Gamma}(x;\ \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \tag{4.6}$$

This distribution is a common choice for describing distribution of rates, because it is nonnegative and conjugate with the exponential family.

We now consider an arbitrary 2-month period in the data. The comparison of the degree distri-

butions in the real data and fitted model are shown in Figure 4.11. For the Gamma distribution we find $\alpha = 0.82$ and $\beta = 2.94$, and these parameters are very consistent regardless of which 2-month period we choose.

We can now simulate a dataset by creating a random graph with this degree distribution, and generating homogeneous, memoryless point processes on each edge with rate drawn from the estimated Gamma distribution. We then run the persistent cascades algorithm on this generated dataset, and the original data, and compare distributions of cascade size and recurrence. Before making this comparison, we show that this entire process can be described analytically.

4.4.2 Analytical model

Now we will introduce framework to describe this model in a precise way, by extending results in the field of percolation theory and epidemic spreading. We will make the same assumptions as in the previous section (i.e. network structure and average rates of interaction fit to the data, but all events iid). At a high level, we will:

1. derive a probability distribution of a cascade (outbreak) of size s happening after n steps, denoted $P_s^{(n)}$,
2. use this to upper bound the probability of a *particular* cascade occurring among a *particular* set of users,
3. use this in a binomial distribution (coin flipping) to describe the probability of this cascade occurring multiple times (i.e. persistence).

Overview

Recall that we will adapt epidemic modeling to our application such that the contagion is information, the outbreak resulting from a single seed is a cascade, and the probability of "infection" is determined only by the rate of interaction between the two involved individuals (and not any notion of disease infectivity).

In [53, 51, 30] and others, the authors introduce analytic forms for the final size of an outbreak in an SIR epidemic model for graphs with arbitrary degree distributions; that is, the probability of the size of the outbreak after the disease has "run its course," and all individuals are either susceptible or recovered. A surprising finding in these papers (for example [53]) is that the probability distribution of final outbreak sizes, P_s, sums to $u = \sum_s P_s < 1$, with the interpretation that $1 - u$ represents the probability of a population-wide outbreak (or "epidemic"), which is not captured by the model. The argument goes that system-size outbreaks would contain loops that would invalidate the formalism of their model.

Marder [47] proposes instead a model which tracks the *stepwise* size of the outbreak, as a distribution $P_s^{(n)}$ representing the probability of an outbreak of size s after n steps. In doing so, he finds that the distribution actually breaks into two parts: a piece which converges in the limit to a finite outbreak, and a piece which grows exponentially in mean and variance as n increases. It is this second piece which explains the possibility of *epidemic* spreading.

We will adapt Marder's technique of stepwise outbreak tracking for two reasons: (1) it allows us to capture the *depth* of a cascade, and (2) it gives a more accurate probability in the short-term,

where we are interested. We will then extend this distribution to a distribution on the likelihood of a *particular* outbreak happening *multiple* times during a time period — in other words, a persistent cascade.

Epidemic size distribution.

The following derivation follows closely from [53] and [47]. Consider a random network with the degree distribution d_i of any node i given by $\mathbf{P}(d_i = k) = p_k$ for any i. Now define the generating function

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k \tag{4.7}$$

Generating functions are a common workhorse in the study of random graphs, as they encapsulate an entire distribution in a manipulable form, as we will see. As an example, note that, given a generating function G, we can recover the underlying distribution p_k by taking successive derivatives and evaluating at zero:

$$p_k = \frac{1}{k!} \frac{\mathrm{d}^k G}{\mathrm{d}x}\Big|_{x=0} \tag{4.8}$$

We can also recover the moments of the distribution. Note, for instance, that $G_0'(1) = \sum_{k=0}^{\infty} k p_k = \langle k \rangle$, i.e. the average degree.

We may also consider the distribution of the degree of any vertex that we reach by traversing an edge, not counting the vertex we started at (called the *excess degree*), and define its generating function in terms of p_k as

$$G_1(x) = \frac{G_0'(x)}{G_0'(1)} \tag{4.9}$$

see [53] for a derivation.

Now consider a network with a single infected individual. At each step, with unit probability, infected individuals infect their susceptible neighbors. (Later we will extend this to when the probability of infection is < 1.) The probability of having s infected after n steps call $P_s^{(n)}$. The generating function for this distribution define as

$$H^{(n)}(x) = \sum_{s=0}^{\infty} P_s^{(n)} x^s \tag{4.10}$$

So at step $n = 0$, we have $H^{(0)}(x) = x$, and at $n = 1$ we have $H^{(1)} = xG_0(x)$ since $G_0(x)$ gives the probability of a node's degree, and we started with one individual. Continuing, and using the *powers* property in [53], we can show that $H^{(2)}(x) = xG_0(xG_1(x))$, and so we have the recurrence relation

$$H^{(n)} = H^{(n-1)}(xG_1(x)) \tag{4.11}$$

For easier iterative calculation, we will define

$$F^{(0)}(x) = 1, \qquad F^{(n)}(x) = G_1(xF^{(n-1)}(x)), \qquad H^{(n)}(x) = xG_0(xF^{(n-1)}(x)). \tag{4.12}$$

following [47].

We can then extract $P_s^{(n)}$ by taking the appropriate derivative of H as previously mentioned.

However, we need to resort to numerical differentiation here, and in practice, the recursive definition of H and inherent small values leads to machine precision errors beyond the first 10 or so values of s (see Newman, Marder). Instead, [53] recommends applying the Cauchy integral formula to instead derive

$$P_s^{(n)} = \frac{1}{s!}\frac{\mathrm{d}^s H}{\mathrm{d}x^s} = \frac{1}{2\pi i}\oint_\gamma \frac{H^{(n)}(z)}{z^{s+1}}\mathrm{d}z \tag{4.13}$$

with γ the unit circle (in the complex plane) $|z| = 1$. We can use the parameterization $z(t) = e^{2\pi i t}$ to rewrite this as

$$P_s^{(n)} = \int_0^1 e^{-2\pi i s\theta} H^{(n)}(e^{2\pi i\theta})\mathrm{d}\theta. \tag{4.14}$$

Following [47], we can evaluate this integral at some large number of points M around the unit circle, m/M for $m = 0, 1, ..., M-1$, which will approximate the integral closely with the Riemann sum, and is then in the form of an inverse discrete Fourier transform, that is

$$P_s^{(n)} = \frac{1}{M}\sum_{m=0}^{M-1} e^{-2\pi i s m/M} H_m^{(n)} = \frac{1}{M}\mathcal{F}_{\mathrm{DFT}}(H, -1)[s] \tag{4.15}$$

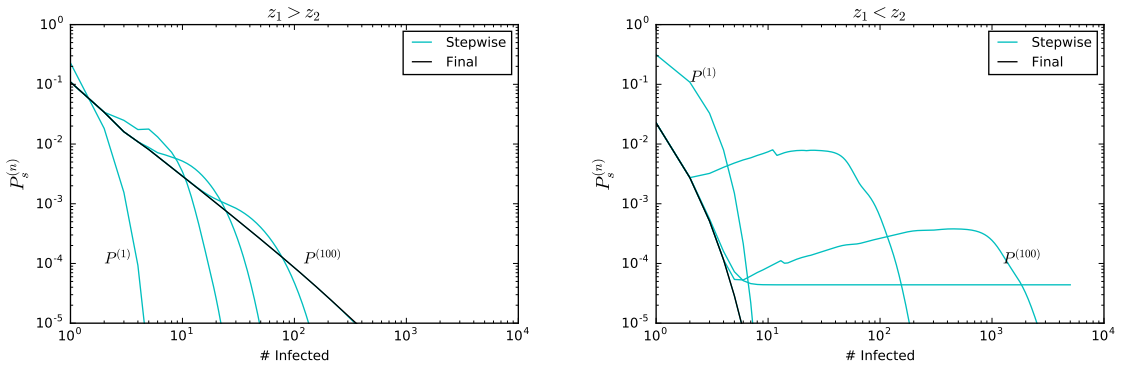
where $H_m^{(n)} = H(e^{2\pi i m/M})$, and the notation $[s]$ simply means retrieving the s-th element from the returned spectra of the transform. We use the Python module `numpy.fft` to carry out this calculation.

Example. As an example, consider a network with generating function $G_0(x) = 0.7x + 0.2x^2 + 0.05x^3 + 0.04x^4 + 0.01x^5$ (taken from [47]). Note that since $z_1 = G_0'(1) = 1.46$ and $z_2 = G_1'(1) = 1.38$ gives $z_1 > z_2$, i.e. the average excess degree is less than the average degree and it can be shown that this implies an epidemic is not possible under the model [47, 53]. Indeed, in Figure 4.12(a) we can see the probability distributions appear to approach a limit as n gets larger (shown are values $n = 1, 5, 10, 100$). In fact, at $P^{(100)}$ the distribution is already indistinguishable from the long-term or "final" distribution given in [53] and others, as mentioned at the beginning of this section. However, we note the distributions are quite different in the short term.

In Figure 4.12(b), we consider a degree distribution such that $z_1 = 3.63$ and $z_2 = 16.1$, which is closer to what we observe in the data. Now $z_1 < z_2$ and we know epidemic spreading is possible. Here the difference between the stepwise method and the "long-term" method becomes more stark. Specifically, we see that the stepwise model consists of two pieces: a finite element that is converging to the long-term method as n increase, and a non-finite element that is increasing in mean and variance exponentially with n, and allows for epidemic (population-level) spreading. [47] gives a more complete treatment of this observation; for our purposes, we are simply interested in the fact that the stepwise model, by being able to capture the entire spectrum of possible outcomes, is able to more accurately capture short-term dynamics.

Transmissibility

Importantly, however, note that we assumed the probability of infection was 1 at each step in the previous derivation, which we do not want to assume. For an infected individual i interacting with a susceptible contact j, the probability of infection should be governed by the average rate of disease-



(a) (b)

Figure 4.12: **Outbreak size modeling.** Comparison of the stepwise [47] and final outbreak [53] models of epidemic spreading, for networks with $z_1 > z_2$ (left) and $z_1 < z_2$ (right). We note that in general, the stepwise model approaches the Newman model as n gets large. However, the stepwise model is able to capture more precisely both short-term and long-term epidemic spreading.

causing contact — or in our case, the average rate of call activity between the users — which we denote r_{ij} and varies from pair to pair. (Again, for purposes of our application, this has only to do with the average rate of interaction, and nothing to do with a notion of the infectivity of the information/disease itself.)

We can model the distribution of these rates as drawn from a distribution $P(r)$, and we will adopt a Gamma distribution form here for convenience. Furthermore, there is a ticking clock on the infection due to our requirement that the cascade occur within a 24-hour period, denote this τ. In the epidemic literature, this captures the *recovery period* wherein i is still "infective." In our application, it captures the short-term importance of the information being spread (and we note this will lead to an upper bound on transmissibility, since in our algorithm we are actually giving each successive member of a cascade a shorter and shorter recovery window before the end of a *fixed* time window arrives).

So, as outlined in [53], denote the probability of transmission from i to j as T_{ij} (and note it may not be symmetric). The probability there is *not* infection is then

$$1 - T_{ij} = \lim_{\delta t \to 0} (1 - r_{ij}\delta t)^{\tau/\delta t} = e^{-r_{ij}\tau} \tag{4.16}$$

and therefore $T_{ij} = 1 - e^{-r_{ij}\tau}$. However, since r_{ij} is iid for each pair in the network, then on a population level it is sufficient to consider the average transmissibility $T = \langle T_{ij} \rangle$ (see [53]), which we can recover by averaging over all possible values of r,

$$T = \langle T_{ij} \rangle = 1 - \int_0^\infty e^{-rt} P(r) \mathrm{d}r \tag{4.17}$$

As a convenient form for $P(r)$ we choose the Gamma distribution, defined in Eq. (4.6). We can

then derive the value for T analytically (in the continuous case) as

$$\begin{aligned} T &= 1 - \int_0^\infty e^{-r\tau} e^{-\beta r} r^{\alpha-1} \frac{\beta^\alpha}{\Gamma(\alpha)} \mathrm{d}r \\ &= 1 - \frac{\beta^\alpha}{(\beta+\tau)^\alpha} \int_0^\infty \frac{(\beta+\tau)^\alpha}{\Gamma(\alpha)} r^{\alpha-1} e^{-r(\beta+\tau)} \mathrm{d}r \\ &= 1 - \frac{\beta^\alpha}{(\beta+\tau)^\alpha} \end{aligned}$$

This makes intuitive sense, as when τ gets larger (our period of interest gets longer), the transmissibility goes toward one, and as the rates skew smaller with larger β, the transmissibility goes toward zero.

As shown in [53], we can simply express the generating function for the degree and excess degree distributions now as

$$G_0(x) = G_0(1 + T(x-1)), \qquad G_1(x) = G_1(1 + T(x-1)) \tag{4.18}$$

to capture this effect. (Derivation involves a straightforward manipulation of the total probability's resulting binomial distribution and is omitted for brevity.)

Extending to persistence

To capture the probability that a particular outbreak (i.e. cascade) happened between the same set of users multiple times (i.e. was persistent), we can take advantage of the fact that, given the outbreak center and size, any set of users is equally likely. We will also discard any notion of approximate similarity, and only consider outbreaks of exactly equal size.

The probability of a specific set $\chi(r)$ of users being in an outbreak rooted at r, with $s_{\chi(r)} = |\chi(r)|$, for any particular seed/root node r, is

$$q^{(n)}_{\chi(r)} = \frac{1}{\#\text{ ways to make } \chi(r)} P^{(n)}_{s_{\chi(r)}} \leq \frac{1}{\binom{k_r}{s_\chi}} P^{(n)}_{s_\chi}$$

with the inequality due to the fact that the actual number of ways to form a set of $s_{\chi(r)}$ nodes is bounded below by the number of ways to form this set from a node's immediate neighbors, k_r.

Now the probability of a particular pattern happening k times over the course of D disjoint periods (for example $D = 60$ days) we will denote as Q^D_χ, and we can upper bound it with a binomial distributed with parameters D and $\hat{q}_\chi$,

$$\hat{Q}^D_\chi \sim \text{Binom}(D, \hat{q}_\chi).$$

As an example, consider a network with $G_0'(1) = 2.9$ and $G_1'(1) = 16.1$ and transmissibility $T = 0.28$ (both of which mirror our real network, as we will discuss in the next subsection). Now consider Figure 4.13. The distribution of outbreak sizes after 2 steps (which is a typical depth of a persistent cascade), $P^{(2)}_s$, is shown on the left. We note this distribution seems to make sense given the degree and excess degree: the probability peaks around the average first degree of 2-3, and sharply drops off after the average count of the first two levels, which is $G_0'(1) + G_1'(1) \approx 20$. For

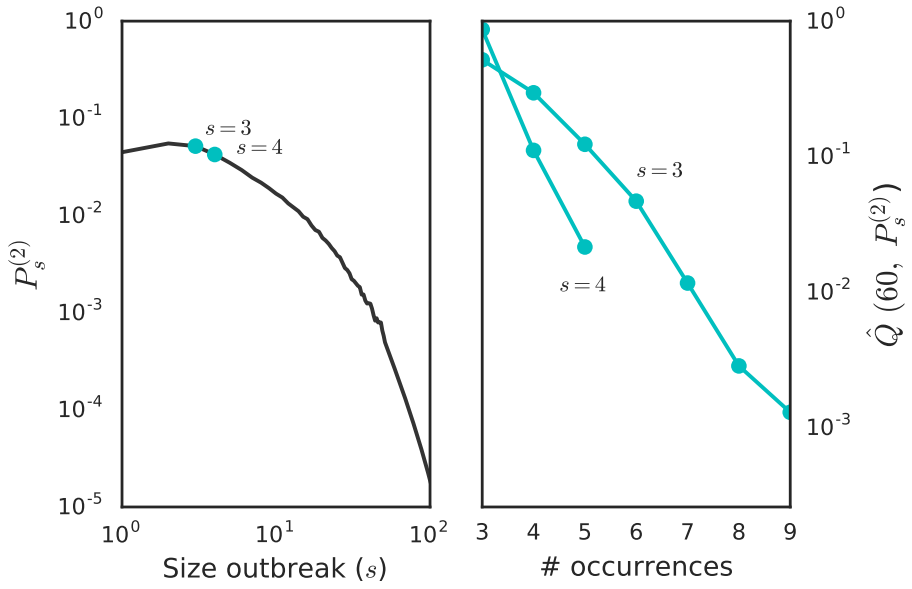


Figure 4.13: **Extending the outbreak model to recurring (persistent) outbreaks.** (Left) Probability of outbreaks of a size s, after 2 steps ($P_s^{(2)}$). (Right) Binomial distribution $\hat{Q}$ for the case of $D = 60$ days and $p = P_s^{(2)}$ for $s = 3$ and $s = 4$, i.e. cascades of size 3 and 4.

cascades of size $s = 3$ and $s = 4$, the corresponding distribution of number of reoccurrences is given on the right of the figure (note we only consider occurrences of at least 3 or more, in anticipation of our persistent cascade analysis that is to follow, and have renormalized the distribution accordingly). This again matches our intuition: there is an approximately exponential dropoff in the likelihood of higher reoccurrence in a random network with non-temporally-clustered interactions.

4.4.3 Findings in the network

We can now compare the resulting distributions of size and frequency of cascades in the real data, simulation model, and analytical model. Recall that we are considering an arbitrary 2-month period of data for this experiment.

Results are shown in Figure 4.14, for both the 3-node case (i.e. P_3 and $\hat{Q}_3^{(k)}$) and 4-node case (P_4, $\hat{Q}_4^{(k)}$). We see the simulated data generally follows the model in both cases, although it drops off slightly more quickly in the 4-node case. In both cases, the real data exhibits a "heavy tail" of cascades occurring 5+ times, in contrast with the random graph models, which exhibit zero probability of reoccurrence past 4-5. In both cases, one can show that the difference in the simulated and real distributions is significantly different — for example using a simple χ^2-test, the significance is at a 99%+-level.

Conclusion. The results in Fig. 4.14 match our expectations at the beginning of this section in both the 3-node and 4-node case — cascades in the data are larger and more recurrent than patterns in simulated data or a modeled distribution even when mimicking exactly the network structure and average rates of interaction. This allows us to reject the hypothesis that these factors are enough to explain the patterns in the data, and leads us to pursue a model which can capture the mutually

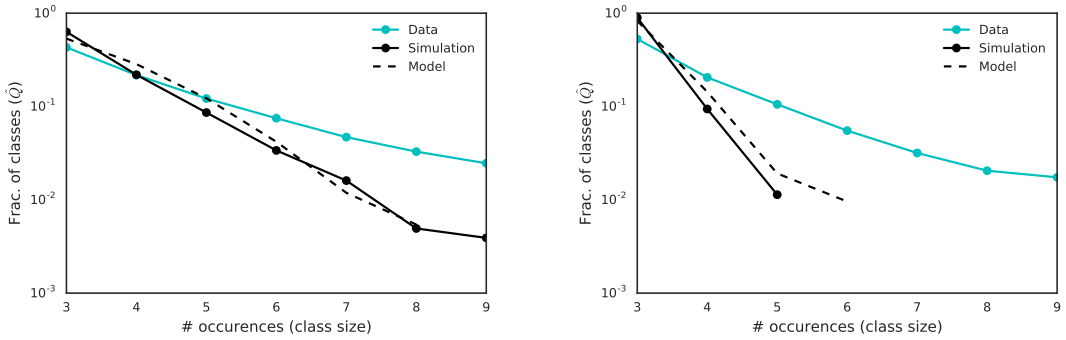


Figure 4.14: **Patterns in the data are significantly different than patterns in a random network.** Comparison of cascade persistence (re-occurence of the same pattern) for cascades of size 3 (**a**) and size 4 (**b**), using results from the actual data, a simulated data set on a scale-free network, and the epidemic model's predicted distribution. The x-axis represents the number of reoccurrences of the same pattern, and the y-axis shows the proportion of all cascades with this amount of recurrence. We see the simulated data closely follows the model in the 3-node case, but drops off more quickly than expected in the 4-node case. In both cases, the real data exhibits a "heavy tail" of cascades occurring 5+ times, in contrast with the random graph models, which exhibit near-zero probability of reoccurrence past 4-5.

exciting nature of group interactions.

4.5 Effects on centrality and diffusion

4.5.1 Diffusion: role of spreaders

What does the presence of cascades, and specifically persistent ones, have on information spreading? For example, if information is seeded at an arbitrary point in the network, does cascade membership increase the probability of receiving it?

We will again frame our problem in terms of epidemic modeling, where the "disease" again represents in our case information, and "transmissibility" pertains to the ability of an individual to pass information to a social contact. Let us first delve into the dynamics of spreading under this model, develop a case for why cascades might see an increased probability of receiving information, and then finally test our hypothesis through simulation on the real data.

Note that this section *simulates* the spread of information but uses the *real* order and timing of interactions observed in the data. We of course do not have access to second-by-second tracking of the spread of some real piece of information or news through the network (as we might in a Twitter or internet blog or email dataset), but we are instead claiming that if such a spreading process were occurring, where the probability of the news being passed was λ, our simulations reveal the dynamics of what that spread would be.

Background

Several papers, as mentioned in Chapter 2, have shown evidence that the "bursty" nature of human interaction actually has a *slowing* effect on spreading dynamics. The intuition is that the long tails of inactivity slow down population-level spreading; compare this with classical models where we assume complete mixing of the population, which is closer to random activity. However, it is

not immediately obvious that long tails of interarrival times would always slow down spreading: wouldn't bursts of rapid, efficient activity *assist* spreading under a ticking-clock model like SIR, since it would help the disease spread before the infected nodes recovered?

In [50], they give an interesting mathematical argument for this apparent paradox. We summarize their findings here to set the stage for results by simulation in the next subsection.

Consider a simple SIR-type spreading process. An individual i is infected by some contact at time t_α, and there is some waiting time τ_{ij} between this event $* \to i$ and the next interaction i has with some j, $i \to j$. Note this is different than the interevent time δt_{ij}, and the distribution can be approximated with

$$P(\tau_{ij}) = \frac{1}{\bar{\delta t}_{ij}} \int_\tau^\infty P(\delta t_{ij}) d\delta t$$

Note that then $P(\tau)$ "inherits" any properties of $P(\delta t)$, such as being potentially heavy-tailed.

Insight #1. The burstiness of human communication *slows down* spreading due to the long tails of inactivity; however, the causal nature of these bursts (since receiving communication usually induces you to communicate to others) actually *speeds up* spreading, since an infected individual will generate several quick secondary spreaders. These are counterbalancing forces.

So then consider (as in the previous section) the *transmissibility* $\mathcal{T}_{ij}$ of an edge $i-j$, representing the possibility of infection from i to j, which is a function of the rate of infection λ and the recovery window T. This can be represented, for a single edge, as $1-(1-\lambda)^{n_{ij}(t_\alpha)}$, where $n_{ij}(t_\alpha)$ is the number of $i-j$ interactions in the time window $[t_\alpha, t_\alpha + T]$. Then, the total probability of infection is

$$\begin{aligned}\mathcal{T}_{ij}(\lambda, T) &= \sum_{\alpha \in \{\alpha *\}} P(t_\alpha = \alpha) \cdot \left(1-(1-\lambda)^{n_{ij}(t_\alpha)}\right) \\ &= \left\langle 1-(1-\lambda)^{n_{ij}(t_\alpha)} \right\rangle_\alpha\end{aligned}$$

where we can take the average if we assume each possible $* \to i$ infection event is iid.

Now, note that when λ is small, $1-(1-\lambda)^n \approx \lambda n$. When λ is big, $1-(1-\lambda)^n \approx 1$. So, we have two "regimes" of transmissibility, depending on the parameter λ:

$$\mathcal{T}_{ij}(\lambda, T) = \begin{cases} \lambda \langle n_{ij}(t_\alpha) \rangle & \text{if } \lambda \ll 1 \\ 1 - P_{ij}^0 & \text{if } \lambda \approx 1 \end{cases} \tag{4.19}$$

where $P_{ij}^0 = P(n_{ij} = 0; T)$, and we can approximate with the density from before as $P_{ij}^0 = \int_T^\infty P(\tau_{ij}) d\tau_{ij}$, basically measuring the probability of a relay time being longer than the recovery period T. When $P(\tau)$ is heavy-tailed, P_{ij}^0 will be larger (for large T) than if $P(\tau)$ is exponentially decaying. Note also that if there is causal correlation between $* \to i$ and $i \to j$, then $n_{ij}(t_\alpha)$ will be higher in "real life" than in a randomized order of events.

Insight #2. We can see, when infection rate is very small, then real order of interactions will actually lead to *higher* transmissibility on edges and thus increase the outbreak size, due to the higher $n_{ij}(t_\alpha)$. When infection rate is larger, the real order of interactions will lead to lower order of interactions because it has a higher possibility of no interactions during T, that is P_{ij}^0 is higher.

Simulation

This analysis by [50] postulates that there are "information cascades" doing the heavy lifting of spreading information, and whose effects are only masked under large infectivity λ. Now we claim to have actually identified conversations displaying these cascading properties, the so-called persistent cascades of this chapter. Our hypothesis is then that when λ is small and we follow the real order of interactions, the persistent cascades will play a significant role in spreading and cascade membership will contribute to higher probability of infection (i.e. receiving information). On the other hand, when λ is large, the cascades' importance will be masked by the high volume of random calls and we will see no significant difference between cascade membership or not.

In another sense, we observe that there is both *random* and *cascading* activity occurring simultaneously in the real data, that we have identified the individuals constituting both groups, and so by tracking the epidemic spread separately for both, we should see the contrast in infective dynamics between regimes of λ without even randomizing the order of calls.

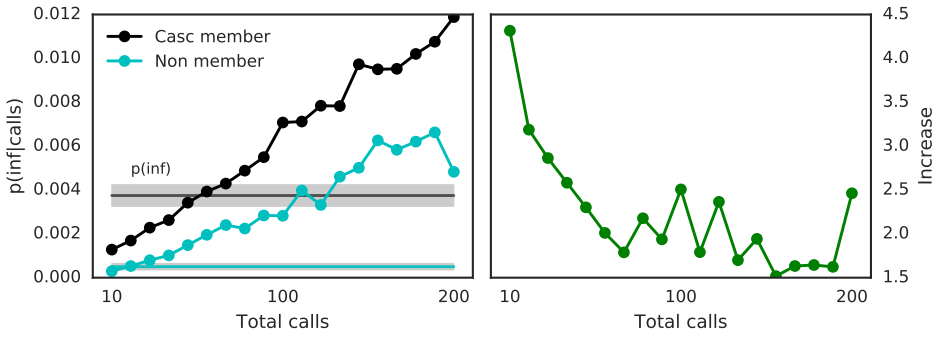
Experiment setup. We simulate the SIR model in the temporal social network resulting from about a month of CDR data. We start each simulation by choosing at random 1,000 nodes, and considering all other nodes as susceptible. We ensure there is an equal probability of cascade members or non-members chosen as seeds in each simulation. We then step through the call data in order, and in each call letting the caller infect the callee with probability λ. Infected nodes recover after a period T, and cannot be infected again. We continue until all nodes are susceptible or recovered. We repeat this for 100 simulations of 1,000 seeds spread across the network.

We consider two regimes of infectivity, $\lambda = 0.05$ and $\lambda = 0.3$, with the recovery period $T = 3$ days. We measure the probability that a node is infected by counting the fraction of times it is infected over all simulations, and average this across all nodes in a particular type of cascade membership and range of call activity. We are controlling for number of total calls since we want to separate out any increase in probability of infection from simply having more exposure in general.

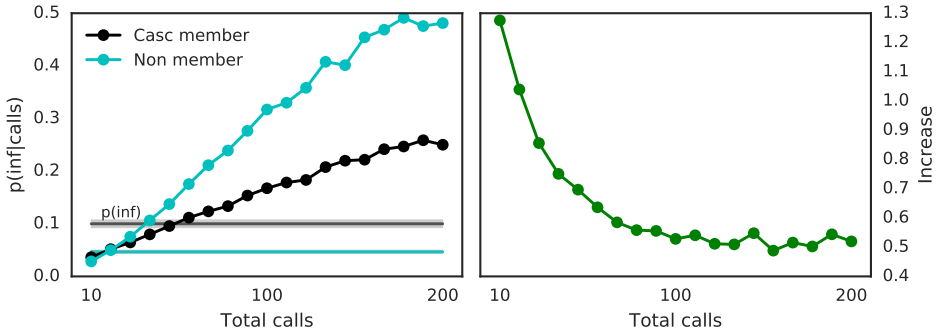
The specific values of λ are chosen to be comfortably far away from the transition point (i.e. when the spreading process tends to become population-scale) on each side. This transition point is determined empirically to be approximately $\lambda = 0.15$ for this data.

Discussion. Results of this experiment are shown in Figure 4.15. We find that our hypothesis appears to hold. First, in the case of small λ: cascade membership appears to signficantly increase the probability of infections, especially for those with an above-average number of calls. It contributes about a 1.5 to 3 times increase in probability regardless of activity level.

This is an interesting finding: those who contribute to persistent cascades are more likely to receive a piece of information seeded randomly in the network than those who are not in this club, and this property is mostly independent of their overall activity level. One may imagine an office where the director sends out a bi-weekly email with priorities for the day, which generates a persistent cascade of emails between project managers. Our finding indicates that this group of people is more likely to receive, say, a randomly seeded email virus (with low infectivity), than their coworkers — even coworkers with much higher levels of activity — simply because of the persistent nature of their communication. The intuition is that the random traffic is not efficient enough in



(a) $\lambda = 0.05$



(b) $\lambda = 0.3$

Figure 4.15: **Importance of persistent cascades in information spread.** Here we compare the probability of infection under SIR model given cascade membership or not (and controlling for different levels of call activity), for small (top) and large (bottom) values of λ. The **left** plots show the **probability of infection** given a particular range of total call activity, with bins $[10, 20]$, $[20, 30]$, etc. (Note this is a series of conditional probabilities, not a distribution.) We see that under the low infectivity regime ($\lambda = 0.05$) the cascade members are more likely to receive information due to the effectiveness and persistence of their call patterns. By contrast, under a high infectivity regime ($\lambda = 0.3$) the random mixing masks this effect. Population average across all simulations (not controlling for call activity) shown as a solid line, with two standard deviations above and below shown as a shaded rectangle. The **right** plot gives the resulting **increase due to cascade membership** and emphasizes that this effect is not simply correlated with higher call activity.

spreading the virus before the recovery period "kicks in."

In the case of large λ, the effect of cascade membership is completely surpassed by random calls, again as expected. Also, we note that this effect is still uncorrelated with overall activity. Interestingly, the cascade members due surpass the non-cascade members in the case of low activity nodes, but this appears to be minor and we do not investigate it further in this work.

Finally, note that the population average probability of infection (shown in Figure 4.15 as solid lines in the left plots) do not reveal this regime change. At a population level, the persistent cascades appear to always be more vulnerable to infection. However, it is evident from comparison to the probabilities when controlling for overall call activity that this is simply the effect of cascade members tending to have more activity in general, which we know will necessarily increase the probability for infection through sheer exposure.

Table 4.2: **Persistent cascade effect on notions of centrality.** This table illustrates the difference in what nodes are deemed "central" with or without the persistent cascade analysis. We separate individuals by whether they are ranked in the top 10% or not of the network as measured using cascade-weighted degree, or unweighted degree. The entry in **bold** corresponds to individuals who are in the top 10% of users in the network using a cascade-weighted measure of degree, but are in the bottom 90% using an unweighted, aggregated approach. In other words, this group (which constitutes 6.6% of the total population) is highly central in information spread, but is unnoticed using a standard approach.

		Weighted	
	k_i (degree) rank	*Bottom ranked*	*Top ranked*
Unweighted	*Bottom ranked*	195,248 (83.9%)	**15,357 (6.6%)**
	Top ranked	18,020 (7.7%)	10,261 (4.4%)

* Bottom rank = lower 90% of users, top rank = top 10% of users

4.5.2 Cascade-weighted network

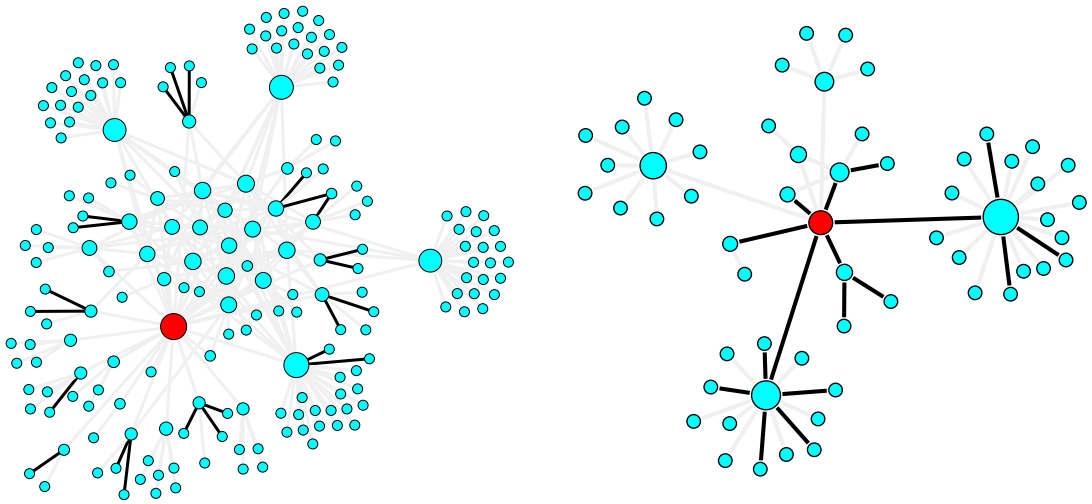
Now consider applying this knowledge of persistent structure back to a static structure, and observing the effect on, in particular, centrality. Specifically, for a network $G = (V, E)$, weight the subset of edges E_C that are present in at least one persistent cascade with $w_c = \alpha \in [0.5, 1]$ and all $e_n \in E \setminus E_C$ with $w_n = 1 - \alpha$. Now with $\alpha = 0.5$ we recover the standard aggregated network, and with $\alpha > 0.5$ we are putting extra weight on the "persistent" edges which we claim carry more meaning.

This results in a network of about 278k nodes and 505k edges, with about 45k users having at least one persistent class of 2 or more cascades (counts are for City A). Setting α in $[0.5, 1)$, we find a giant connected component (GCC) comprising 80-85% of the total network for all three datasets (cf. [57]). With $\alpha = 1$, the GCC splits into several thousand smaller subgraphs, the largest usually being about 2k nodes. This echoes previous results that show the inability of information to reach any sort of macroscopic diffusion when traveling solely through information cascades [59], and is a version of our earlier findings in this chapter on the connectedness of the k-persistent subnetwork.

We now consider the weighted degree (or node strength [54]) of a user i, defined $k_i = \sum_j A_{ij}$, where A is the adjacency matrix of G and $A_{ij} = w_c$ if $(i, j) \in E_C$, w_n if $(i, j) \in E \setminus E_C$, and 0 otherwise. We examine a 1-month time period in City A, for both the unweighted (i.e., $\alpha = 0.5$) and cascade-weighted ($\alpha > 0.5$) networks. We use the s_{TED} measure for this analysis, with $\ell = 0.8$. We observe the effects of the weighting in Table 4.2, which presents the overlap of central and non-central users for both networks as measured by degree, when $\alpha = 0.5$ against when $\alpha = 0.9$.

We note several groups that emerge: first, the large group of users (about 7% of the total users) that are only central in the cascade-weighted network. This suggests a group of users with unremarkable importance as measured in a naïve way by counting calls, but who play a pivotal role in the persistent communication patterns of their social network. Similarly, a large group of influential users in the standard unweighted network disappears when we begin weighting cascades, implying their centrality was only due to a web of edges corresponding to mostly random calls. And lastly, we note that a large portion of the network has their status essentially unchanged.

Figure 4.16: **Persistent cascade effect on notions of centrality.** Here we contrast the local network of a typical user in the bottom left group of Table 4.2 against one in the top right group of the Table. The person-of-interest is depicted as a red node, all others as light blue. Edges present in a persistent cascade depicted in black, all others as gray. Sample includes friends and friends-of-friends. On the **left**, the person of interest has a high number of different social contacts (high unweighted degree), but none of which are actually persistent. By contrast, on the **right**, the person of interest has close to the population average number of different social contacts, but is involved in a large number of persistent cascade activity. This right individual is representative of a large group of users with unremarkable importance as measured in a naïve way by counting calls, but who play a pivotal role in the persistent communication patterns of their local network.



(a) High unweighted degree

(b) High cascade-weighted degree

4.6 Case study: HRC Emails

Our methodology thus far has been designed for metadata where we are inferring information spread by extracting recurrent patterns in the data. It would be revealing to apply the method to data where we *do* have access to the content: in this case, we could do the pattern matching as before, keeping ourselves blind to the content, and then afterwards do an analysis with the content now at hand. We are also interested to test whether the methodology is generalizable to other, non-mobile phone, datasets.

To this end, we will conduct a case study of sorts, using a dataset of emails from Hillary Clinton's private email server, which were recently made public as part of a federal investigation.

4.6.1 Data

Hillary Rodham Clinton (HRC) was the Democratic nominee for President of the United States during the 2016 campaign season. She was involved in a long-running and heavily politicized controversy during this campaign regarding her use of a private email server during her previous tenure as

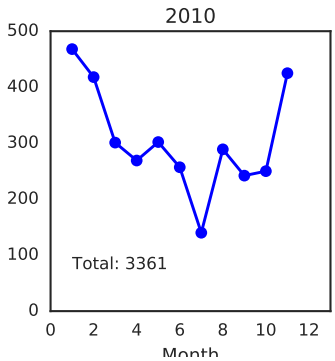


Figure 4.17: Number of emails in the HRC emails dataset, by month. This only depicts emails for which we have a timestamp (required for our analysis).

Secretary of State. There were a string of Freedom of Information lawsuits demanding a full release of the emails, which crescendoed in mid-2016 with the State Department releasing approximately 7,000 of the confiscated email records. These are available from various sources, but we use the publicly available set available at https://www.kaggle.com/kaggle/hillary-clinton-emails because it has been reasonably cleaned and vetted from the original PDFs.

After doing some baseline data preparation to remove missing timestamps and resolve aliases for the different active users in the dataset (e.g. "Huma Abedin," "abedinh@state.gov," "Abendin" [sic], etc.), we are left with 3,361 emails over a 1-year period in 2010. (See Figure 4.17.) There are 382 unique users, most with activity in the range of 100-200 emails. The full network is shown in Figure 4.18, with the four individuals with highest activity labeled. These four individuals are: Hillary Clinton (1); Jacob Sullivan (2), deputy chief of staff during this period; Cheryl Mills (3), chief of staff; and Huma Abedin (4), deputy chief of staff. There are also 280 emails with an unlabeled sender/receiver, so we represent these unknowns as a single node in the network.

4.6.2 Persistent cascade analysis

We now apply the persistent cascades algorithm to this dataset. We find that, using a similarity threshold of $\ell = 0.8$ and thresholding at $k = 3$ cascades per class (cluster), there are 11 root users generating persistent cascades with these minimum requirements. This is only about 2% of the "population," which is lower than the CDR dataset, but this is not surprising since this email network is much more centralized (around one user, HRC).

Already, there are several interesting observations about these 11 roots. First, it includes the 4 high-activity individuals, but also several users with below-average activity, such as Judith McHale (Under Secretary of State) and Richard Verma (Assistant Secretary of State for Legislative Affairs). It also excludes the node representing unknown sender/receiver, which is encouraging since this node was relatively high-activity, but should not be generating any sort of persistent events.

Distribution of class and cascade size. The distributions of both class size (i.e. how many persistent cascades there are in a class) and cascade size (i.e. how many individuals are in a persistent cascade) in Figure 4.19 looks very similar to what we saw in the CDRs, in that there is a long tail of large classes, but most are only 3-4 cascades. This is again more than expected under a random model, as we showed for the CDR data in a previous section.

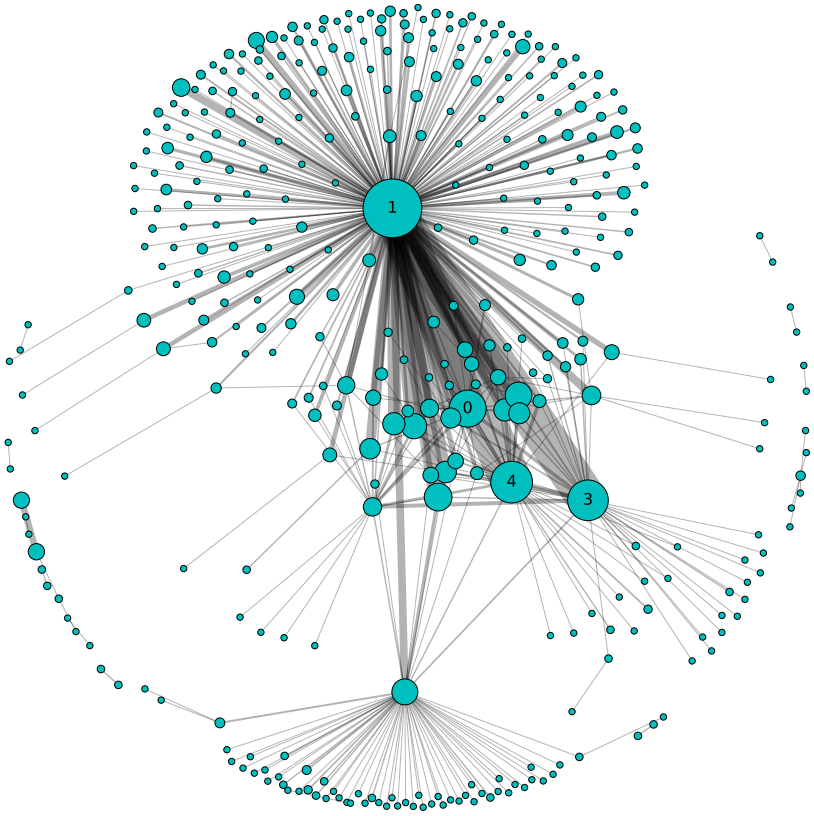


Figure 4.18: **Communication network structure in the Hillary Clinton emails dataset.** Labeled are the four individuals with highest overall email activity: Hillary Clinton (1); Jacob Sullivan (2) and Huma Abedin (4), deputy chiefs of staff; and Cheryl Mills (3), chief of staff. Edges appear wherever two individuals have exchanged at least one email, and edge thickness denotes email activity within the pair (undirected).

Centrality. We now repeat the centrality analysis from the previous section. Recall that we will compare the high-centrality nodes (e.g. top 10%) in the unweighted, aggregated network against the high-centrality nodes when we place higher weight on edges involved in persistent communication. Using again $\alpha = 0.9$ as our weight, we find a similar group of "hidden spreaders" emerges (see Table 4.3). In the top 10% of individuals under both the weighted and unweighted analysis, we find such high-activity and central users as HRC, Huma Abedin, and others already mentioned.

However, we find 11 individuals who are unremarkable (i.e. lower 90%) in terms of total degree, but who are in the top 10% of individuals as weighted by membership in persistent cascade activity. As an example, one of these individuals is Doug Band, whose email correspondence is limited but about 80% of the time with HRC in a persistent cascade.

Table 4.3: **Centrality analysis in the HRC email dataset.** Contrast of top ranked users (by degree) in the standard unweighted vs. cascade-weighted network, in the HRC Email dataset. Users in **bold** (2.8% of total pop.) are highly central in information spread, but are unnoticed using a standard approach.

		Weighted	
	k_i (degree) rank	*Bottom ranked*	*Top ranked*
Unweighted	*Bottom ranked*	333 (87.2%)	**11 (2.8%)**
	Top ranked	14 (3.7%)	24 (6.3%)

* Bottom rank = lower 90% of users, top rank = top 10% of users

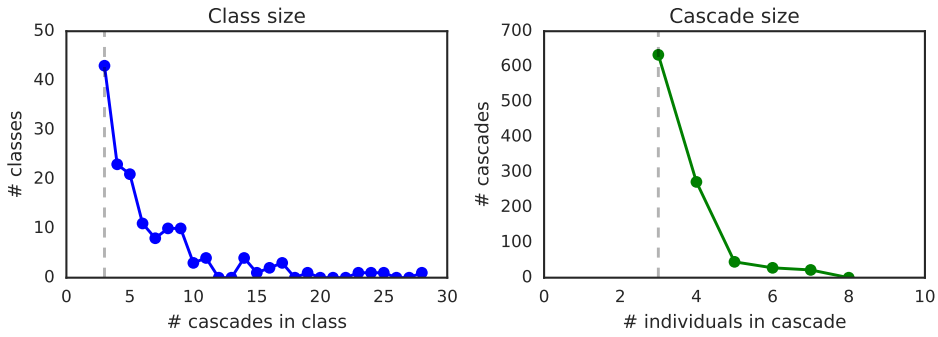


Figure 4.19: **Persistent class membership and size is similar to mobile phone data.** Distribution of **(left plot)** class size (i.e. how many persistent cascades are in a class), and **(right plot)** persistent cascade size (i.e. how many individuals are in a persistent cascade). This mirrors findings in the CDRs, in terms of the long tail of large classes and number of individuals per cascade being higher than expected under a random model.

Example content. Finally, we will discuss a few examples of content within the discovered persistent cascades, although we leave more in-depth analysis to future work (for example, bag-of-words text comparison within a cascade against a random sample).

The following three examples are summaries of the structure and content of *actual persistent cascades* in the dataset, and give a sense for the type of communication we observe:

- `sbwhoeop` (corresponding to Sid Blumenthal) emails Hillary, who in turn forwards the information to Philip Reines or Jake Sullivan. Mr. Blumenthal's emails are typically in the vein of breaking-news, usually in foreign policy issues, which Hillary then ensures her deputy chief of staff is aware of (e.g. February 5th 2010, "FWD: Northern Ireland. FYI," regarding a "historic" power-sharing agreement in Northern Ireland).

- Anne-Marie Slaughter (Director of Policy Planning) emails Hillary, who then emails Jake Sullivan for opinion or situational awareness. For example, in early April, about a British planning policy, or a project proposal read-ahead for "creating more leverage" (context unclear).

- Less high-profile, but relevant, is the multiple cascades that find a pattern of Cheryl Mills, Jake Sullivan, or one of 3-4 other high-activity staffers, emailing Hillary, who then forwards the informaiton to Lauren Jiloty (special assistant) and asks "Pls print."

We do not do a comparison of text similarity in persistent cascade emails against a random sample, for example by using a bag-of-words style analysis. This may be fruitful, and we expect that content within persistent cascades is more similar than content from email chains on randomly selected days and times. However, we wish to emphasize that the specific content of the emails is not the aim of the persistent cascades analysis, as much as the observation that communication in a persistent cascade is more meaningful than elsewhere. We expect this to be borne out by visible information spreading from time to time, but in general we do not require a "viral message" to be passed to be able to say something like "communication from Sid Blumenthal to Hillary is meaningful, persistent, and tends to produce communication with Jake Sullivan."

4.7 Conclusion

In this chapter we introduced a novel method for extracting temporal patterns of information spread from large-scale communication metadata, using methods of inexact tree matching and hierarchical clustering. We showed that analysis of these so-called *persistent cascades* reveals new properties of information spread, such as weekday-weekend roles, a habitual hierarchy of spreading, and long-term persistence on the scale of months and years. We showed that these patterns are significant by comparing them to both analytical and simulated models of the network, indicating that the temporal clustering inherent in real communication patterns is critical to producing the persistent cascading patterns we observe in the real data. We also showed that these persistent cascades play a crucial role in information spreading through simulation of diffusion processes on the temporal network — specifically, members of a persistent cascade are more likely to receive information spreading through the network under realistic conditions of spreading. Lastly, we showed that this analysis leads to new understanding of centrality and revealed a population of super-spreaders who were otherwise unremarkable under an aggregated approach.

We also indicated directions for extension of the method: first, our assumptions about the structure of information spread limited our ability to detect all relevant patterns (as revealed through an exhaustive search), and second, our pattern-mining method of identifying structure limited our ability to describe the relationships in the network in a probabilistic way. In the next chapter, we will introduce a probabilistic model for approaching this problem that addresses these concerns.

Modeling Influence Structure with Hawkes Processes

5

We now seek to answer the question: how can we *model* and *predict* the underlying influence structure of a communication network? We use a point process called a *Hawkes process* that can model the important properties we observed in the previous chapter, such as temporal clustering and information cascades, and also provides us with an interpretable, predictive influence network structure. We propose a novel methodology for parameter estimation of this model, apply it to the mobile phone datasets, and find it both extends our findings related to the persistent cascades and reveals new properties.

5.1 Introduction

5.1.1 Motivation

The graph-mining approach presented in the previous section is designed to identify recurring patterns indicative of information spread, and allows us to analyze the effect of those patterns on diffusion and individual roles in the network. However, it is not designed to model the network, or quantify the observed structure in a probabilistic sense. We may extract a recurrent temporal structure, but we have no clear way of describing relative importance (such as tie strength), or being able to predict the occurence of the structure in the future. As outlined in the background chapter, previous work has been done in this vein using general probabilistic models (such as [19, 16, 20]), and modeling the network as a point process (such as [63, 58, 43, 72, 78]).

The *Hawkes process* is a flexible point process we will implement in this chapter which allows events to exert influence on future events. This influence is additive and decaying with time, and may be extended to a multidimensional case where there are many different *streams* of events.

The Hawkes process was proposed in 1971 [24], and many of its early application was in econometric modeling, but its generalizability has since brought it wide use in modeling, for example:

- stock price fluctuation [7, 43],
- earthquake activity [72],
- gang violence [65],

- neuron impulses in the brain [44],
- social networks [79, 58],
- trend detection [60, 17],
- and product adoption [70, 18].

This gives us a rich literature of techniques to draw on and gives our methodology broad applicability.

5.1.2 Contributions

We offer three methodological contributions. First, we present a technique to adapt the Hawkes process, in the multivariate case, to model events on *edges* instead of *nodes*. We term this the *Dyadic Network Hawkes* model, and we argue it is often a more sensible paradigm for communication and influence networks where observations involve two individuals. Second, we introduce a Bayesian maximum aposteriori (MAP) expectation-maximization (EM) approach which allows us to incorporate a prior distribution on the amount of influence between individuals. This regularization is absolutely critical on networks of any size to prevent overfitting due to the large number of parameters, and our proposed methodology allows us to use the elegant framework of EM without the heavy machinery of nuclear norm regularizers. Third, we propose a simple way to translate the estimated influence matrix into an individual measure of influence and susceptibility.

We also demonstrate our proposed approach in real communication data. First, we apply the method in the 1-dimensional case to the previously identified persistent cascades, which we now reinterpret as single group conversations where each call represents an event. This analysis shows us that the group conversations split into two clusters: one that is highly "excitable" but with relatively low background intensity, and one that is moderately excitable with high background intensity. Second, we apply the model in the multidimensional case to samples of the mobile phone data without prefiltering with the persistent cascades methodology. We show that the method gives interpretable estimates of the network influence structure. We then perform the persistent cascade analysis on the same samples, allowing us to compare the distribution of estimated measures of influence and susceptibility between persistent cascade members and non-members. We show that persistent cascade members are both more *influential* and more *susceptible* than the population on average.

5.2 Methodology

5.2.1 Theoretical preliminaries

Poisson process

A *point process* is a stochastic process that generates a random and finite series of events that are governed by a probabilistic rule. For example, consider a series of points along the nonnegative real line such that the probability of k points on any interval length n is given by a Poisson distribution

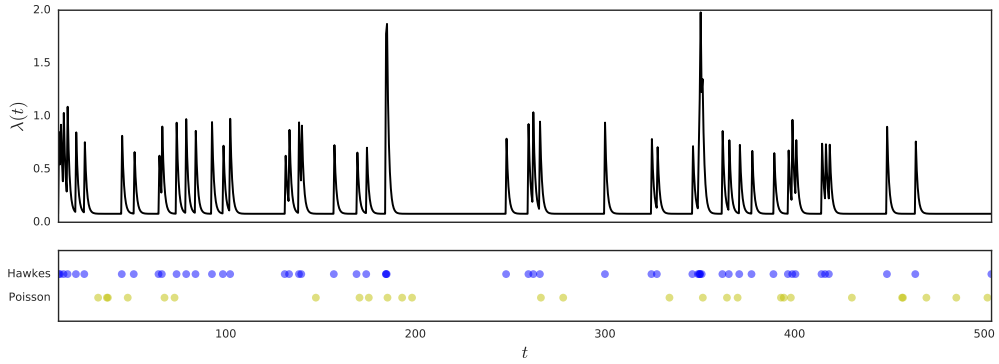


Figure 5.1: **Temporal clustering in Hawkes process is not present in Poisson process.** Bottom chart shows arrivals in a univariate Hawkes process (HP) contrasted with a Poisson process. Note the temporal clustering and "burstiness" inherent in a Hawkes process, not apparent in the memoryless Poisson process. Top chart shows the corresponding intensity function for the HP arrivals.

with parameter λn — this particular process is called the Poisson process (with rate λ). Because of our application, we will always consider the real line to represent time, such as an interval $[O, T]$.

We may even consider a U-dimensional Poisson process, with U different Poisson processes generating events in $\mathbb{R}^U$, each with a different rate λ_u. Now, the overall number of points in a particular interval (or now more appropriately, volume) is again given by a Poisson distribution with parameter λn, where the rate $\lambda = \lambda_1 + ... + \lambda_u$ (the Poisson superposition theorem). This additive property allows us to compute the probability that a particular event originated from a particular dimension u_i as λ_{u_i}/λ.

One can also show that the number of events in disjoint subsets are independent of each other. This leads to the critical observation that the Poisson process is *memoryless*. Another way to state this is that the *interarrival* time between two successive events is an exponential random variable, and therefore the probability of the next interarrival time is independent of the previous. This memoryless property makes the Poisson process an extremely tractable, and universally applied, modeling tool.

Hawkes process

However, the memoryless property of Poisson processes means that it is unable to capture a dependence on history, or in other words, interaction between events. For example, we may want the event of an arrival to increase the probability of arrivals in the next small interval of time. For this, we introduce the *Hawkes process* ([24]), which gives an additive, decaying increase to the intensity function for each new arrival. Now, the intensity function is only *conditionally* Poisson: that is, *given* the history of events $\{t_i\}$ up to t, the conditional intensity at t $\lambda(t|t_i < t)$ is Poisson. [43, 36]

Definition 5.2.1 (Hawkes process)**.** Consider a sequence of events $\{(t_i, u_i)\}_{i=1}^n$ consisting of a time t_i and dimension u_i (i.e. the i-th event occurred at time t_i in dimension u_i), for $t_i \in \mathbb{R}^+$ and $u_i \in \mathcal{U} = \{1, 2, ..., U\}$. This sequence is a *Hawkes process* if the conditional intensity function has the parameterized form

$$\lambda_u(t; \Theta) = \mu_u + \sum_{i:t_i<t} h_{uu_i}(t - t_i; \theta_{uu_i}) \tag{5.1}$$

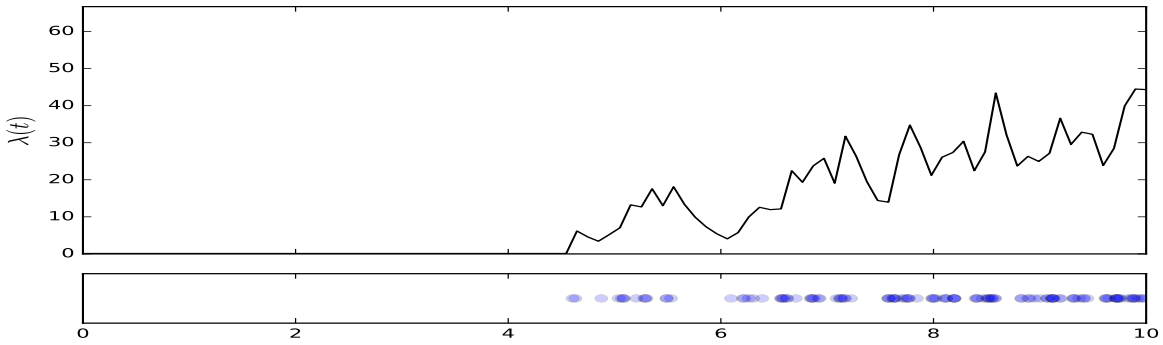


Figure 5.2: **Example of a nonstationary Hawkes process.** When the excitation parameter $\alpha > 1$, the expected change in intensity tends to infinity and the process has the tendency to "blow up." The example above is a univariate example with $\mu = 0.1, \alpha = 1.1, \omega = 3$.

where $\Theta = (\mu, \theta)$ are the model parameters and $H = [h_{ij}]$, $h_*(t) : \mathbb{R}^+ \to \mathbb{R}^+$ is the matrix of *triggering kernels* (also sometimes called the *excitation function* or *decay kernel*) which is varying with u and u_i.

Contrast with Poisson process. Note that when $h \equiv 0$, we recover the (homogeneous) Poisson process with rate μ and the intensity is independent of the history $\mathcal{H}(t) = \{t_i : t_i < t\}$. In contrast, a Hawkes process with $h > 0$ is *self-exciting*: recent arrivals increase the value of the intensity function, thereby generating more arrivals. This property results in stronger "clustering" of arrival events than observed in homogeneous Poisson processes. As an example, consider Figure 5.1 which shows a realization of a Hawkes process (top) and Poisson process (bottom). The Hawkes process displays clear temporal clustering, also evident in the "sawtooth" behavior of its intensity (top of chart).

Interpretation as superposition of Poisson process. On the other hand, we may also interpret the Hawkes process as a superposition of multiple Poisson processes. One can imagine, in a single dimension u, the base rate leading to a sequence of events (Poisson with parameter μ_u), and each summand leading to a sequence of events (with parameter $h_{uu_i}(t - t_i)$). In this way, the probability that a particular event t_j was the result of, say, the background rate, is:

$$\mathbf{P}(t_j \text{ background}) = \frac{\mu_u}{\mu_u + \sum_{i:t_i<t_j} h_{uu_i}(t_j - t_i)} \tag{5.2}$$

or in other words, the fraction of the total rate at time t_j that came from the background rate.

This property will be highly useful later when we introduce the idea of the Hawkes process as a branching process and exploit this latent (i.e. unknown) structure in an expectation-maximization scheme for parameter estimation.

Triggering functions, branching process, stationarity

The triggering function controls how much past events affect future ones, and should be defined for all pairs (u, u') over all $u \in \mathcal{U}$. A common choice is a scaled exponential function (e.g. see [70, 78, 72, 79, 24, 58, 17]), which is interpretable and computationally tractable. Other forms have

been explored (e.g. power-law), although [79] show that choice of functional form is less critical to performance than accurate parameter estimation. We therefore adopt the exponential form, defined below.

Definition 5.2.2 (Exponential triggering function)**.** Decompose the triggering kernel matrix $H = [h_{ij}]$ into an *influence matrix* $A = [\alpha_{ij}]$ and *exponential decay kernel* $G(t) = [g_{ij}(t)]$, such that $H = A \odot G$ and

$$h_{uu'}(t;\ \alpha, \omega) \stackrel{\text{def}}{=} \alpha_{uu'} g(t;\ \omega), \qquad g(t;\ \omega) = \omega e^{-\omega t}. \tag{5.3}$$

where we have set a global parameter ω, and let α_* vary between dimensions.

This has the immediate interpretation that as an event becomes more distant, it has exponentially less effect on the probability of a new event occurring. We can tune the ω parameter to adjust the rapidity of this decay, and tune the α parameter to adjust the relative weights different dimensions place on each others' activity (including α_{uu}, the self-excititation of a dimension on itself).

The practice of treating ω as a global parameter has precedent in [78, 70, 17] and allows us to avoid the addition of U^2 new parameters to the model.

The scaled exponential, as defined, also has an intuitive form if we interpret the Hawkes process as a branching process. Consider the univariate case $U = 1$. Note that when the intensity $\lambda(t) = \mu$, we can consider any arrivals as *parent* events. Now some immediately subsequent event (where now $\lambda(t) > \mu$ due to the excitation of $h(\cdot)$) is either another parent event, or (more likely) an *offspring* that was caused by a previous parent event's increase in the intensity function.

Under this interpretation, $\alpha > 0$ controls the *branching ratio*, or likelihood of an arrival causing another arrival.

Furthermore, we note that in the univariate case when $\alpha > 1$, the process N is nonstationary; i.e. $\mathbb{E}[N(t + \delta t) - N(t_0)] \to \infty$ as $t_0 \to \infty$, for any choice of δt. This nonstationarity mirrors standard results in branching processes (such as the Galton-Watson process), and is easily seen by noting that, when $\alpha > 1$, each parent event produces infinitely many offspring in expectation. See [36] for further discussion.

In the multivariate case, this has the natural extension to the matrix of $A = [\alpha_{ij}]$, and one can show that in order to ensure stationarity, the largest eigenvalue of A must be less than 1,

$$\rho(A) \stackrel{\text{def}}{=} \max_i |\lambda_i| < 1 \tag{5.4}$$

where λ_i here denotes the eigenvalues of A. This again makes sense by considering that the expected number of offspring at each generation is related to the successive powers of A, which must be constrained to have a largest eigenvalue less than one in order to be certain to converge. We refer the reader to [43, 45, 36] for further discussion.

5.2.2 Simulation method

For learning the parameters $\Theta = (\mu, A, \omega)$ of this model, it will be useful to test our methods on synthetic data generated from known parameters, where we can compare our results to some "ground truth." In this subsection we describe our method for simulating a multivariate Hawkes process. We improve the standard algorithms with two small but important modifications.

Simulation of a Hawkes process is nontrivial. A well-worn approach which we will adapt (with some improvements) is known as *Ogata's thinning method* [55], which essentially generates new events from an exponential distribution parameterized by the Hawkes intensity at that time, but then rejects some events with some probability that decreases as the time since the last event increases. In the multivariate case, this is only slightly more complicated, since we must also attribute each generated event to a particular dimension based on the proportional likelihood the new event came from that dimension.

Algorithm 2 describes the details, but let us mention two important modifications.

The algorithm as typically described [66, 36] requires $O(n^2U^2)$ operations to draw n samples over U dimensions, which is prohibitive for large graphs. Instead, we modify an approach mentioned in [70]. Namely, given the rates at the last event t_k (which note do not include effects of t_k), we can calculate $\lambda(t)$ for $t > t_k$ by

$$\lambda_u(t) = \mu_u + e^{-\omega(t-t_k)}\big(a_{uu_k}\omega + (\lambda_u(t_k) - \mu_u)\big) \tag{5.5}$$

which we can do in $O(1)$, and only requires saving the rates at the most recent event. Note also that when $0 < t - t_k < \epsilon$, this reduces to

$$\lambda_u(t) = \lambda_u(t_k) + a_{uu_k}\omega \tag{5.6}$$

or in other words, the previous rate plus the maximum contribution the event t_k can make since it has just occurred.

Secondly, we find that texts describing the algorithm typically frame the attribution/rejection test as finding an index n_0 such that a uniformly random number on $[0, 1]$ is between the normalized successive sums of intensities around that index (see e.g. [66, 36]). We would like the reader to note that this entire procedure amounts to a weighted random sample over the integers $1, 2, ..., U + 1$ where the probabilities are the normalized rates, and selecting $U + 1$ is equivalent to the "rejection" condition. This allows us to use optimized package software for weighted random samples, instead of something like a for-loop (as is present in even production-level Hawkes process software), that also slightly speeds up the procedure.

The algorithm, with these two speedups, is described in Algorithm 2. We can generate many thousands of events in 1-2 seconds in this way.

5.2.3 Dyadic Network Hawkes

Problem setup and assumptions. We would like to apply this model to data where we believe there is some network structure. Specifically, we are interested not only in whether A is in contact with B, but what the influence A and B have on each other, and perhaps what influence this interaction may have on other pairs of individuals in the network.

Let us first assume that the communications between individuals as captured in some large-scale dataset like mobile phone or email records is a good proxy for observing the interpersonal interactions between individuals. (This of course is not always true: many individuals may never communicate through the medium under study, or there may be occasions when a cell phone call generates an in-person meeting which we do not observe, etc.)

Algorithm 2 Simulation of a multivariate Hawkes process

Input: $\mu = \{\mu_u\}$, $A = [a_{ij}]$, ω, horizon
Output: $\{(t_i, u_i)\}_{i=1}^n$

First event:
$I^* \leftarrow \sum_u \mu_u$
$t_0 \sim \text{Exp}(1/I^*)$
$u_0 \leftarrow u$ w.p. μ_u / I^*
$\lambda(t_0) \leftarrow \mu$

General procedure:
$k \leftarrow 0$
Step 1
$I^* \leftarrow \sum_u \lambda_u(t_k) + \omega \sum_u a_{uu_k}$
Step 2
$t' \leftarrow t_k + s,\ s \sim \text{Exp}(1/I^*)$
if $t' >$ horizon **then**
return $\{(t_i, u_i)\}$
$\lambda(t') \leftarrow \mu + e^{-\omega(t'-t_k)}\big(A_{u_k}\omega + \lambda(t_k) - \mu\big)$
Step 3
$u' \leftarrow u$ w.p. μ_u / I^*
if u' is $u+1$ **then**
Step 2 (Reject)
else
$t_{k+1} \leftarrow t'$
$u_{k+1} \leftarrow u'$ (Attribute)
$\lambda(t_{k+1}) \leftarrow \lambda(t')$
$k \leftarrow k+1$
Step 1

Second, let us assume that when A and B communicate, this has the possibility of increasing the probability that, say, B and C will communicate shortly thereafter. We say that this increase in probability represents the *influence* that $A-B$ has on $B-C$. (Note that we are implying an undirected model, and are not analyzing the more complicated case when $A \to B$ influences $B \to C$, but not $C \to B$, for example.) Further, let us assume that this influential spike in probability decays over time — if $A-B$ last spoke two months ago, there is very low probability this event has any bearing on the present.

Applying the Hawkes process model. Consider a network $G = (V, E)$. As described already in Eq. (5.3), we first separate the triggering kernel $h_{uu'}$ into two parts, a branching ratio $\alpha_{uu'}$ and exponential decay kernel $\omega e^{-\omega t}$.

Now one can interpret $\alpha_{uu'}$ as a quantified measure of the influence of u' on u, and so $A = [\alpha_{ij}]$ becomes the weighted adjacency matrix of the network G. In this work, the nodes in the network are individual entities (such as people, gangs, neurons, stocks), and the weights $a_{uu'}$ are the influences between them. (In [43], he goes further to separate each α into a $0-1$ element and a weighted element, to setup expressing prior beliefs separately on the connectivity and influence structure of the network.)

However, since we are measuring influence through interpersonal communication (i.e. anonymous communication data), it does not make as much sense to measure the activity of a single person on each dimension — each "event" involves two individuals already. We would prefer to measure each dimension as an *edge* in the network — in other words, we would like our Hawkes process to model events on each *dyad* of individuals, and determine the effect that activity on one dyad influences activity on another. We again emphasize we are considering the undirected version of this interpretation.

We call this the *Dyadic Network Hawkes* model. As the reader may suspect, it is a straightforward process to move from one model to the other, as we will now show.

Definition 5.2.3 (Line graph)**.** For an undirected graph $G = (V, E)$, the *line graph* $G' = L(G)$ is the graph such that each edge in G maps to a node in G', and two nodes in G' are connected if the corresponding edges in G share an endpoint. We can compute the adjacency matrix $A(G')$ by employing the *incidence matrix* of G, which is the $m \times n$ matrix $B(G) = [b_{ij}]$ such that $b_{ij} = 1$ if edge i is "incident with" (connected to) node j. Now,

$$A(G') = B(G)B(G)^T - 2I \tag{5.7}$$

where I is the $m \times m$ identity matrix.

This is a simple operation and can be easily performed on graphs of hundreds of thousands of nodes. It can also be extended to the directed case, but we do not pursue this in this chapter. There is an analogy here to G being the *primal* graph, and G' being the *dual* graph, but we point out that the dual of the dual is *not* the primal in this case; that is, $L(G') \neq G$ (in general).

This formula allows us to simply apply the standard network Hawkes approaches, but with the resulting matrix A corresponding to the adjacency matrix $A(G')$.

For completeness, we wish to mention that there is a unique correspondence between the line graph G' and the original graph G, and so it is also possible to move in the inverse direction and recover the original graph given only G'. (This is true for all graphs except a special case when G is the triangle graph on 3 vertices. [37]) There are several algorithms available for this task which can be solved in linear time $O(m)$, for example [37].

However, this is not relevant for our work for two reasons. First, the primary hurdle in such an algorithm is determining which two nodes in G correspond to a node G' — in our application, this is already known, since our data provides this information. Second, our resulting estimated influence matrix $A = [\alpha_{ij}]$ will only rarely be symmetric, meaning we have a kind of directed line graph which does not have an obvious interpretation in terms of G. This is an interesting property that may merit future work, but we omit any further discussion in this thesis.

5.2.4 Parameter Estimation: Expectation-Maximization

Having defined the model, we now propose a method for estimating its parameters. Especially in the multivariate case, the problem appears daunting in the sparsity of available information and the high number of parameters. We will show that with using a straightforward application of Bayesian maximum aposteriori (MAP) expectation-maximization (EM) to the Dyadic Network Hawkes problem, we can achieve strong regularization customizable to known priors about the network.

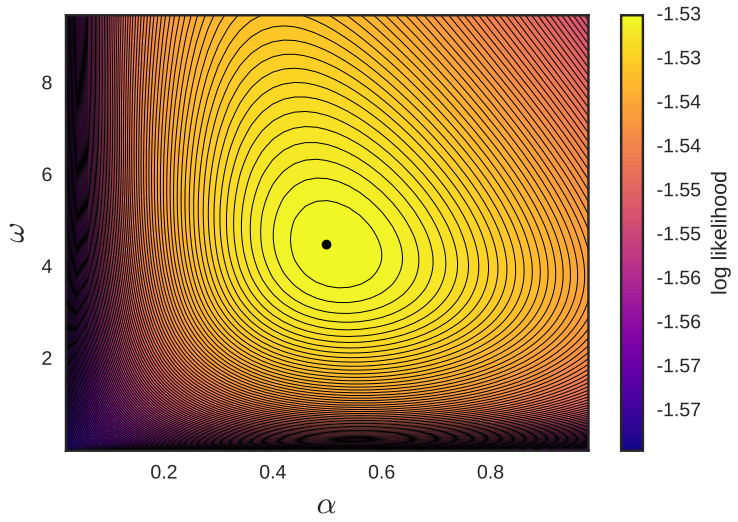


Figure 5.3: **Flat log-likelihood makes direct ML estimation difficult.** For a univariate process, we simulate synthetic events with known ground-truth parameters μ, α, ω. In this contour plot, we fix μ at the ground-truth value, and show the contours of the log-likelihood function for varying α and ω. This illustrates the shallowness of the MLE objective function near the optimum.

Challenges in Direct Maximum-Likelihood (ML) Estimation

Before introducing the EM schemes, we present the maximum-likelihood estimator (MLE) for this model and justify why we are avoiding it.

There is actually a convenient closed form of the log likelihood for a multivariate Hawkes process. While in principle this should enable standard 1st or 2nd-order optimization schemes for parameter estimation, in practice such methods pose many challenges. The main problem is the low curvature near the local optimum, as shown in [72]. This low curvature leads to vanishing gradients in 1st-order methods, and severe numerical instability associated with inverting near-degenerate Hessians for second-order methods. For completeness, we introduce the likelihood function here, visualize it, and discuss in more detail the obstacles to estimation, before motivating an EM-based approach that circumvents these difficulties in the next subsection.

Consider a sequence of events $\{\tau_i\}_{i=1}^N$ where each $\tau_i = (t_i, u_i)$ corresponding to the time t_i of the event and the stream u_i upon which it occurred. The likelihood of a given sequence $\tau = \{\tau_i\}$ is given by

$$\mathcal{L}(A, \mu) = \sum_{i=1}^{N} \log \left(\mu_{u_i} + \sum_{t_j < t} a_{u_i u_j} g(t_i - t_j) \right) - T \sum_{u=1}^{U} \mu_u - \sum_{u=1}^{U} \sum_{j=1}^{N} a_{u u_j} G(T - t_j) \tag{5.8}$$

where $G(t) = \int_0^t g(s) ds$.

Our first concern is the large number of parameters (on the order m^2) and resulting tendency for over-fitting. As a result, we need to introduce strong regularization on the parameters, such as sparsity and/or low-rank regularization on A. We could then maximize the log-likelihood or

$$\min_{A,\mu} \; -\mathcal{L}(A, \mu) + \mathcal{R}(A, \mu) \tag{5.9}$$

where $\mathcal{R}$ represents some regularization on A and/or μ to enforce sparsity, low-rank, etc.

However, in practice the function is extremely "flat" around the optimum, causing problems with slow convergence in first order methods, and near-degenerate Hessians for second-order methods. Figure 5.3 shows the situation in a one-dimensional process, with μ fixed and varying α and ω. This phenomenon is noted in many works on this process, e.g. see [72].

Research with this approach typically introduces regularization on A and/or μ (such as an L_2 norm in [70]) and highly sophisticated optimization machinery to handle the resulting non-linear and unwieldy objective function. For example, [78] uses an alternating-direction method of multipliers (ADMM) scheme with with a majorization-minimization (MM) step at each iteration, and both L_1 and L_* (nuclear) norm regularizers on A.

Regularized (Bayesian) Expectation-Maximization

Instead, we will use an Expectation-Maximization (EM) approach which, besides the advantage of in our case having concise closed-form expressions for the parameter updates at each iteration, makes beautiful use of the natural interpretation of *branching structure* in a Hawkes process. We will introduce regularization on A by applying a Bayesian, maximum aposteriori (MAP) version of the EM algorithm with a prior on the triggering function. The EM approach has much precedent as a preferred means of parameter estimation for Hawkes processes (see [72, 79, 78]), but to our knowledge, the particular MAP EM approach introduced here is novel for both the univariate and multivariate case.

In general, the EM algorithm works by introducing some latent variable Q such that it is more tractable to optimize the *complete data likelihood* — which is in terms of the data, the parameters, and Q — than to optimize just the data likelihood, which does not include Q. Of course, we do not know Q, so EM proceeds by finding an *expected value* of Q, and then maximizing the (expected) complete data likelihood using this estimate of Q.

(Sidenote: There is an analogy, therefore, between EM and the projected gradient descent method, whereby we take a step in the direction of the negative gradient and then project back into the feasible space. In EM, the maximization step is in the direction of the optimal complete data likelihood, and the expectation step projects back into the space of Q. In [40] they show a correspondence between the two methods in the case of Hawkes processes.)

In our case, the latent variable is the *branching matrix* Q describing the parent-descendent relationship of each event in the process, as described in the introductory section. Specifically, let $Q = [q_{ij}]$ represent the latent branching matrix such that $q_{ij} = 1$ if event i was caused by event j (0 otherwise), and note $q_{ii} = 1$ implies i was a background event. We will see that expressing the (complete) data likelihood using this extra information gives the problem extra structure that aids in fast and accurate convergence.

We will take a Bayesian treatment of EM, so we seek to maximize the complete data posterior, defined below.

Definition 5.2.4 (Complete data posterior of a Hawkes process.)**.** For a sequence $\tau = \{(t_i, u_i)\}_{i=1}^N$, branching matrix $Q = [q_{ij}]$, and parameters Θ, the complete data posterior is

$$p(\Theta|\tau, Q) \propto p(\tau, Q|\Theta)p(\Theta; V) \tag{5.10}$$

where V are hyperparameters of our prior on Θ. Let

$$\mathcal{L}(\tau, Q; \Theta, V) = \log p(\tau, Q; \Theta) + \log p(\Theta; V) \tag{5.11}$$

be the complete data log likelihood under the parameters Θ and hyperparameters V.

It now remains to express the complete data likelihood, that is, what is the likelihood of a particular sequence *and* branching matrix given the parameters Θ. Following [78], we can express the complete data log-likelihood as

$$\begin{aligned}\log p(\tau, Q|\Theta) = &\sum_{i=1}^{N} p_{ii} \log \frac{\mu_{u_i}}{p_{ii}} + \sum_{i=1}^{N}\sum_{j=1}^{i-1} p_{ij} \log \frac{\alpha_{u_i u_j} g(t_i - t_j)}{p_{ij}} \\ &- T\sum_{u=1}^{U} \mu_u - \sum_{u=1}^{N}\sum_{j=1}^{N} \alpha_{u u_j} G(T - t_j)\end{aligned} \tag{5.12}$$

where T is the end of the observed time interval $[0, T]$ and N is the number of events.

In the **E-step** of the EM algorithm, we compute a current distribution over the latent variable Q. Since Q is a matrix of (Bernoulli) indicator variables, the distribution is expressed by the *expected branching matrix* $P = [p_{ij}]$ based on the data τ and our current parameter estimate Θ^k. Formally, we compute

$$P^{(k+1)} = \mathbb{E}[Q|\tau, \Theta^{(k)}] \tag{5.13}$$

which will be

$$p_{ii}^{(k+1)} = \frac{\mu_{u_i}^{(k)}}{\mu_i^{(k)} + \sum_{j=1}^{i-1} a_{u_i u_j}^{(k)} g(t_i - t_j)} \tag{5.14}$$

$$p_{ij}^{(k+1)} = \frac{a_{u_i u_j}^{(k)} g(t_i - t_j)}{\mu_i^{(k)} + \sum_{j=1}^{i-1} a_{u_i u_j}^{(k)} g(t_i - t_j)} \tag{5.15}$$

regardless of decay kernel $g(t)$.

In the **M-step** of the algorithm, we use this to maximize the (expected) complete data posterior log-likelihood:

$$\begin{aligned}\Theta^{(k+1)} &= \text{argmax} \left\{ \mathbb{E}[\mathcal{L}(\tau, Q^{(k)}; \Theta, V) \mid Q^{(k)} = P^{(k+1)}] \right\} \\ &= \text{argmax} \left\{ \mathbb{E}[\log p(\tau, Q^{(k)}; \Theta) \mid Q^{(k)} = P^{(k+1)}] + \mathbb{E}[\log p(\Theta^{(k)}; V)] \right\}.\end{aligned} \tag{5.16}$$

Up to this point, we have not needed to specify our decay kernel $g(t)$. We will now incorporate the exponential decay kernel defined in Eq. (5.3) and show we can solve Eq. (5.16) in closed form by taking the gradient and setting to zero.

Exponential triggering and Gamma regularization

We will choose a convenient form for our decay kernel to be $g(t) = \omega e^{-\omega t}$, with ω fixed for the entire process, as shown in a previous section. We will also use a Gamma prior on the influence matrix A, which is conjugate with the Poisson random variables in the complete data likelihood and

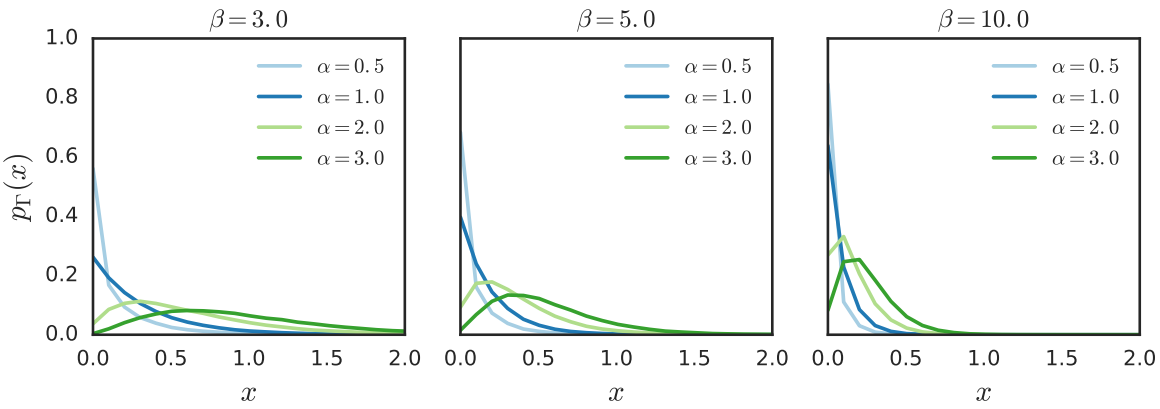


Figure 5.4: **Examples of the Gamma distribution.** The Gamma distribution Gamma(α, β) provides an interpretable prior for entries of the influence matrix $A = [\alpha_{ij}]$. We see that larger β leads to values distributed close to 0, while larger α increases the mean and variance. In the context of the $A = [\alpha_{ij}]$ entries being branching ratios, β represents a pseudocount of *parent* events, and α represents a pseudocount of *child* or descendent events.

thus tractable, and also provides an intuitive explanation of hyperparameters as "pseudocounts."

Specifically, consider the prior

$$p(A;\ V) = \prod_{i,j} p(a_{ij};\ V_{ij}) = \prod_{i,j} \text{Gamma}(a_{ij};\ s_{ij}, t_{ij}) \tag{5.17}$$

with $V = (S, T)$ and where Gamma$(\cdot)$ is the standard gamma distribution

$$\text{Gamma}(x; a, b) = \frac{a^b}{\Gamma(a)} x^{a-1} e^{-bx} \tag{5.18}$$

with mean a/b and variance a/b^2. See Fig. 5.4 for example distributions of Gamma for varying a and b (in the figure, α and β).

Now we compute the stationarity condition $\frac{\partial}{\partial \Theta} = 0$ for the expected complete data posterior log likelihood, which is sufficient for optimality due to convexity of Eq. (5.12) and (5.18), and find

$$\mu_u^{(k+1)} = \frac{\sum_{i:u_i=u} p_{ii}^{(k)}}{T} \tag{5.19}$$

$$\alpha_{uu'}^{(k+1)} = \frac{\sum_{i:u_i=u} \sum_{j:u_j=u', j<i} p_{ij}^{(k)} + s_{uu'} - 1}{\sum_{i=1}^{N} \sum_{j:\ u_j=u', j<i} G(T - t_j) + t_{uu'}} \tag{5.20}$$

We can also approximate $G(T - t_j) \approx \int_0^\infty g(s)ds = 1$ (see e.g. [79, 78]) and the denominator for $\alpha_{uu'}$ becomes simply $N_u + t_{uu'}$ where N_u denotes the number of events such that $u_i = u$.

These updates have useful interpretations that illuminate the role of the hyperparameters $V = (S, T)$. The first update sets $\mu^{(k+1)}$ equal to the expected number of background events per unit time. The second update sets $\alpha^{(k+1)}$ equal to the expected proportion of events that are descendants of a previous one, with the addition of t pseudo-observations of which $s - 1$ are descendant events.

In summary, by iterating between Eq. (5.15) and (5.20), we will converge to a parameter estimate by properties of EM. In practice we will use the log-likelihood defined in Eq. (5.8) as

our convergence criterion; for example, checking every 10 iterations whether $|\mathcal{L}(A^{(k)}, \mu^{(k)}) - \mathcal{L}(A^{(k-1)}, \mu^{(k-1)})| < \epsilon$ for some $\epsilon > 0$.

Note on hyperparameter selection. Note that under this scheme, ω is left as a hyperparameter of the model along with V, that is, we must select these parameters prior to beginning the EM routine and they are not estimated by the EM iterations themselves. We will employ the common practice of separating the data into *training* and *validation* sets. We will iterate over a "grid" of possible values for ω and V, each time fitting μ and A to the training data, then testing the predictive performance of the model on the validation data. After this grid-search validation procedure is complete, we select the ω and V which gave the highest predictive performance (as measured by Eq. (5.8)) on the validation set.

Methodology with kernel updates

When $U = 1$, we may actually treat ω as a parameter of the model and learn it along with μ and α, instead of as a hyperparameter. We will again apply a Gamma prior to ω, with hyperparameters (u, v), incorporate it into the complete data posterior likelihood, take the gradient with respect to ω, set to zero, and solve.

So in addition to the update equations in Eq. (5.20), we may incorporate the following update step for ω:

$$\omega^{(k+1)} = \frac{\sum_{j<i} P_{ij}^{(k+1)} + u - 1}{\sum_{j<i} P_{ij}^{(k+1)}(t_i - t_j) + v} \,. \tag{5.21}$$

This update sets $\omega^{(k+1)}$ equal to the expected number of descendant events divided by the expected total time between descendent events, and therefore has the expected units of a frequency. The hyperparameter u plays the same role as s, while v may be interpreted as the total time between descendant events in the pseudo-observations. When $u = 1$ and $v = 0$ (no regularization), we can view $\omega^{(k+1)}$ as the reciprocal of the expected time between descendant events.

Stability and selection of priors

Recall from Eq. (5.4) that we require the spectral radius of the influence matrix A to be less than one, to ensure stability of the process. We would like to set hyperparameters such that our prior places little mass on unstable systems. Simplifying the method outlined in [43], we will accomplish this by taking advantage of a property of stochastic matrices called the *circular law*.

A variation of the circular law states that the maximum eigenvalue of a $K \times K$ stochastic matrix with iid entries of mean $\mu > 0$ and variance σ^2 is asymptotically distributed as $\lambda_{\max} \sim \mathcal{N}(K\mu, \sigma^2)$ where $\mathcal{N}(\cdot)$ represents the Gaussian (normal) distribution. In our case, we have $\mu = \mathbb{E}[\alpha_{kk'}] = s_{kk'}/t_{kk'}$, our two hyperparameters for the Gamma prior. Thus, we can help roughly ensure that the entire matrix $A = [\alpha_{ij}]$ is (asymptotically) stable by considering the $K \times K$ stochastic matrix where each entry is iid selected from the single prior with the largest mean, and constraining the distribution of this extreme case.

Specifically, we should ensure that

$$K\mu + 2\sigma^2 = K\left(\frac{s_0}{t_0}\right) + 2\left(\frac{s_0}{t_0^2}\right) < 1 \tag{5.22}$$

With this in mind, we propose the following scheme. Consider the aggregated graph G formed from the data by placing an edge wherever there is some threshold of activity between individuals, and consider the associated line graph adjacency matrix $A' = [a'_{ij}] = A(L(G))$. Now let

$$s_{kk'} = 1 + a'_{kk'} s_0 \tag{5.23}$$

$$t_{kk'} = t_0 \tag{5.24}$$

We may select all the $a'_{kk'}$ through an exhaustive cross-validation procedure of all possible network structures; we will instead use the line graph adjacency matrix in the validation data.

It now remains to find s_0, t_0 that give rise to a stable matrix. If we fix s_0, we can compute a lower bound for t_0 using the formula from before as

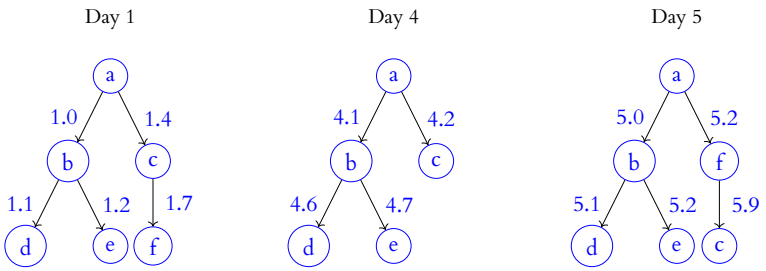
$$t_0 > \frac{1}{2}\left(Ks_0 + \sqrt{(Ks_0)^2 + 8s_0}\right). \tag{5.25}$$

This allows us to adjust a single (hyper)parameter s_0 and achieve a prior on the relationship of each (k, k') pair based on the first-order information gained from the aggregated network, and also ensure that we are placing most of the mass of this prior on a stable matrix.

5.3 Univariate case: modeling persistent cascades as self-exciting processes

Let us test the method in the univariate case on both synthetic and real data. For the real data, we will consider an interesting tie-in with the work in the previous chapter: are the events within a group conversation well-modeled by a self-exciting point process? That is, given the call events within a persistence class, using the methodology in Chapter 3, can we estimate parameters μ, α, and ω and predict future activity? Note this requires us to "collapse" the events of the cascades into a single stream, but we expect is an important exercise as it may reveal properties or categories of persistence classes that are quantifiable in a probabilistic way (which we cannot get from the graph-mining approach outlined in the previous chapter).

As a short example, consider the following persistence class consisting of three similar cascades:



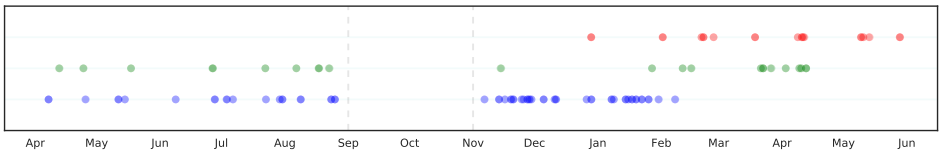


Figure 5.5: **Three example sequences from the data resulting from the persistence cascade analysis.** Dots represent call events within a persistent cascade, and so are calls between approximately the same users, in approximately the same order. There is remarkable consistency on the scale of months to a year. The dashed lines show the 2-month period of missing data that we will use to split training/validation.

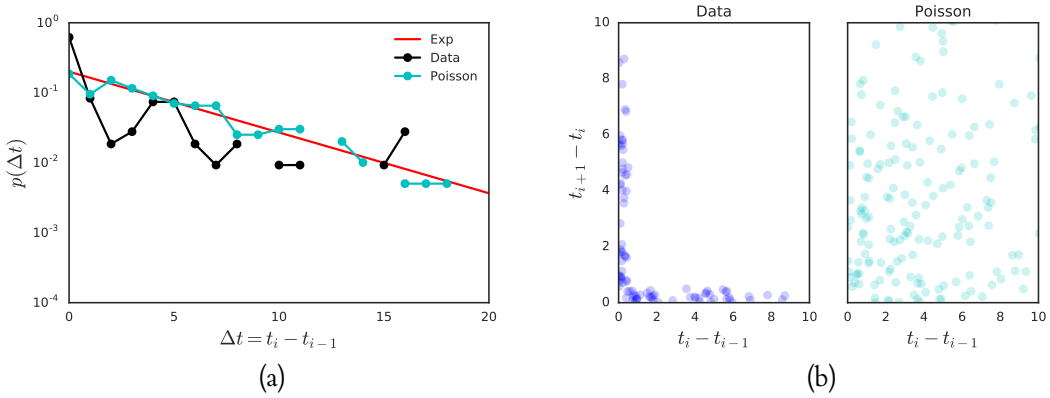


Figure 5.6: **Persistent cascades are not Poisson.** In **(a)**, we compare the distribution of interarrival times in an actual sequence from the data against a Poisson sequence generated with rate equal to the average interarrival time in the data (log-lin scale). A true exponential distribution is shown as a baseline. In **(b)** we show a "lag scatter plot" of subsequent interarrival times in the Data (left) vs. a generated Poisson process (right). It is clear that while there is no correlation in the memoryless Poisson scatter, the data exhibits a clear pattern: long pauses always precede a burst of activity.

and the corresponding sequence of events:

$$\{1.0, 1.1, 1.2, 1.3, 1.4, 1.7, 4.1, ..., 5.2, 5.9\}.$$

We begin by taking the persistent cascade structure and associated call sequence as given. That is, we use the sequences of call events within these already identified persistent group conversations as a starting point, and we focus on modeling, predicting, and analyzing them. Our (processed) data therefore consists of $\mathcal{D} = \{\tau^{(i)}\}$ where each $\tau^{(i)} = \{t_1^{(i)}, ..., t_{n_i}^{(i)}\}$ is the sequence of time stamps corresponding to the sequence of call events in the ith group conversation.

To recall from the previous chapter, some examples of sequences $\tau^{(i)}$ are shown in Figure 5.5. We note remarkable consistency on the scale of months to a year. We see interesting stories developing: in the first sequence, a new group appears to form (possibly new friends from the holidays?); in the third sequence, there is a crescendo of activity followed by the group completely vanishing (possibly planning an event?). We also note the 2-month break in the data — we do not have observations during this period, and will use this as a convenient way to separate our training and validation data.

5.3.1 Testing for fishiness: persistent cascades are not Poissonian

Before proceeding with analysis of the algorithm, let us do a few simple tests to show our data is not well-modeled by a simple 1-dimensional point process, and thus justify our self-exciting model.

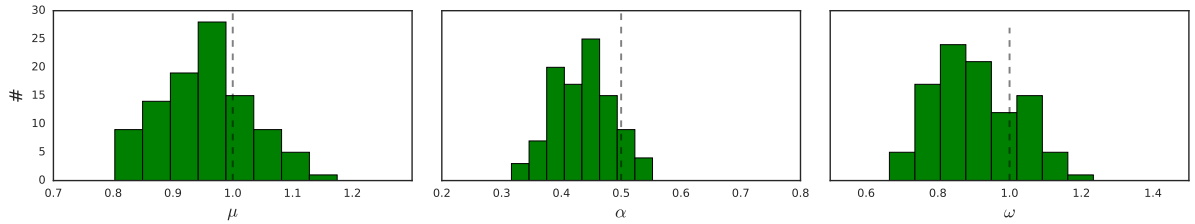


Figure 5.7: **Unregularized EM slightly skewed from ground truth values in synthetic tests.** These histogram depict the estimated parameters using unregularized EM on 100 synthetic univariate Hawkes processes generated using the same ground truth values. Ground truth values represented with a dotted black line. The EM estimates are slightly skewed, suggesting the need for regularization using validation-selected hyperparameters.

There are many ways of testing whether a series of points form a Poisson process. We will show two here, which albeit qualitative, give a convincing negative answer that the sequences in our data are Poissonian.

A first test is to check the distribution of the interarrival times, $\Delta t = t_i - t_{i-1}$. In a Poisson process, these are distributed $\Delta t \sim \text{Exp}(\lambda)$ for some rate λ. In Fig. 5.6(a) we compare the distribution of interarrival times (day scale) in an actual sequence from the data, against a generated Poisson sequence generated with the same base rate. The exponential distribution curve is shown for reference. We can see the Poisson sequence adhering to the exponential curve, while the actual data is more "bursty" — i.e. many short interarrival times, and many very long ones.

A second test is to the check the correlation in subsequent Δt, that is, the correlation between $t_i - t_{i-1}$ and $t_{i+1} - t_i$. If there is no correlation, we have reason to believe the generating process is truly "memoryless" since the Δt's appear to be independent. Fig. 5.6(b) shows the stark contrast between the real data and a sample Poisson process generated with the same base rate.

Taken together, these tests reassure us that there is temporal clustering occurring in the data which merits a more nuanced model. (This echoes the finding in the previous chapter.)

5.3.2 Synthetic tests

Let us also examine the performance of EM on some generated sequences using our methodology. In this way, we can compare the estimated parameters against what we know to be "ground truth." (This replicates experiments in [72, 79].)

We generate 100 sequences over a time interval of $T = 1000$, with ground truth parameters $\mu = 1$, $\alpha = 0.5$, and $\omega = 1$. We then run EM estimation on the resulting sequences, shown in Figure 5.7. We find generally consistent results, but a slight leftward skew in all estimates. This variance and skew decrease as we increase the sequence size (e.g. by increasing T).

Since our data has similar number of arrivals to this generated experiment, we have reason to believe the regularization procedure (with validation selected hyperparameters) will be beneficial.

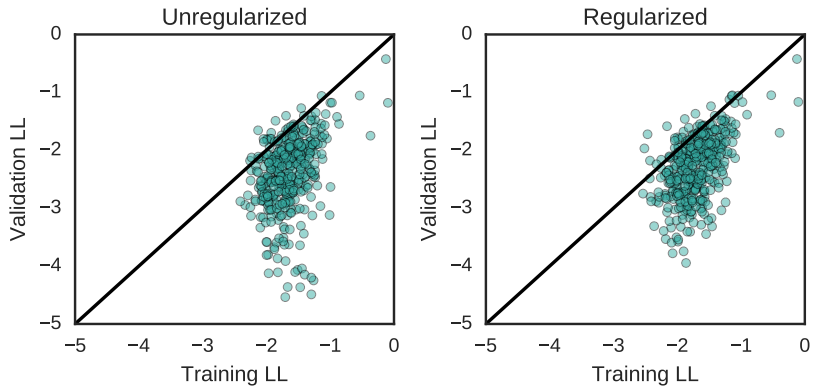


Figure 5.8: **Regularization increases out-of-sample predictive performance.** Shown are scatterplots of the training log-likelihood (horizontal axis) and validation log-likelihood (vertical axis) for unregularized (left) parameter estimates and optimal regularized (right) estimates found via grid-search. Introducing validation leads to higher validation likelihoods and stronger correlation between training and validation scores.

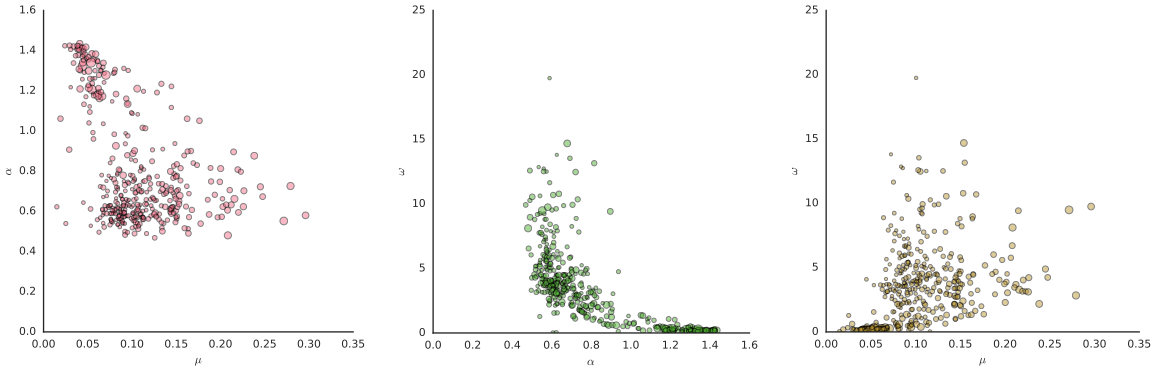


Figure 5.9: **Parameter estimates (using regularized EM) reveal two groups of persistent cascades.** Shown are scatterplots of parameter estimates for μ, α, ω under regularization. Size of dot here indicates size of the sequence. Two distinct clusters of persistent cascade type are evident in the α-μ plot: one with low background activity (μ) but high self-excitation (α), and another with high background activity and moderate self-excitation. The unstable-α group (i.e. $\alpha > 1$) also tends to have much less rapid decay of influence from each event. Taken together, this indicates the first cluster represents conversations that see dense activity for long periods of time, followed by long periods of no activity. The second cluster is conversations that see frequent, small bursts of activity.

5.3.3 Parameter estimation and analysis

Parameter estimation

We now investigate the results of parameter estimation using the Gamma-prior regularized MAP EM scheme.

To review, we will fit the parameters $\Theta = (\mu, \alpha, \omega)$ using the training data consisting of all sequence data before the 2-month break, and select hyperparameters $V = (s, t, v, u)$ using the validation data consisting of all sequence data after the 2-month break. We will focus on the predictive performance of the method out-of-sample, and provide some interpretation of *types* of persistent cascades we gain by clustering in the parameter space.

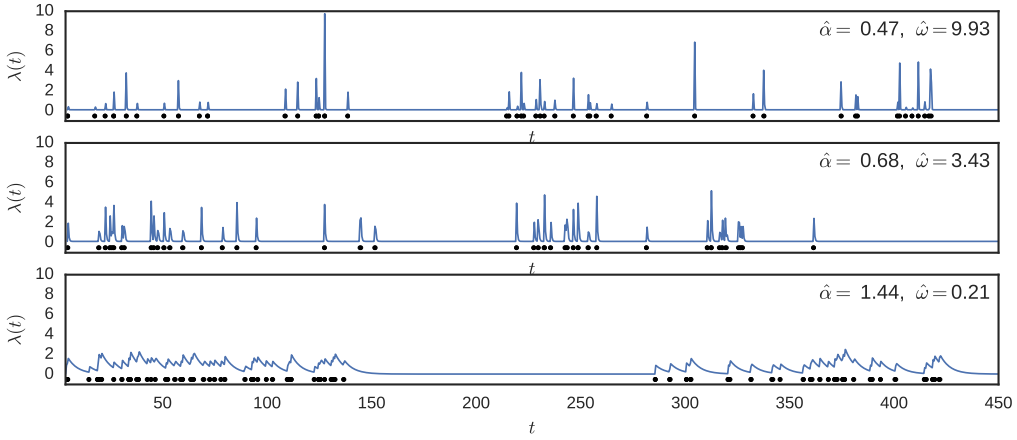


Figure 5.10: **Examples of the two types of persistent cascade.** Depicted are the process events (black dots) and estimated intensities using MAP parameters for sequences with low (top), median (middle) and high (bottom) estimated branching ratios $\hat{\alpha}$. The bottom sequence corresponds to the non-stationary category of persistent cascades, its nonstationarity reflected in the fact that the intensity is almost never at its baseline value. We see, as expected, the top two sequences are characterized by frequent, small bursts of activity, while the bottom sequence is characterized by long periods of dense activity.

Effect of regularization on validation performance. Figure 5.8 illustrates the effect of the Gamma prior regularization on performance in the validation set. In particular, we note that using optimal hyperparameters in regularization (obtained through grid-search) corrects overfitting on a large group of sequences and creates stronger correlation between training and validation scores.

Estimate comparison and non-stationary sequences. Figure 5.9 shows a comparison between all three pairs of parameter estimates, which reveals some of the dynamics at play. Note that in these plots, the dot size indicates the *size* of the sequence, $|\tau^{(i)}|$.

We first note the general trend of positive correlation in the last ω vs. μ plot, which indicates that as the base rate leads to more and more expected arrivals, the effect of each arrival tends to decrease. We also note that this is not limited to longer sequences, where we might expect the effect to be necessary to prevent the sequence blowing up, but even in short sequences.

We now consider the first plot, of α against μ, that the cluster of sequences with non-stationary α also has a much lower μ than the rest of the data. This indicates that the sequences simply have a large number of events, and instead of capturing this with a high base intensity μ, the optimization is using a non-stationary α. This is interesting, since it indicates that a highly temporally clustered process (that is, higher α) is still a better predictor in this case than a simple process with high intensity.

The second plot also shows this non-stationary group behaving with different dynamics as relates to ω — the non-stationary group has very low values of trigger function decay, which is surprising as we might expect the ω parameter to "compensate" for the high branching ratio by being even *higher*.

Categories of persistent cascade by parameter cluster. The analysis just described gives an indication there are two clear categories of persistent cascades: one with low background activity (μ) but high self-excitation (α), and another with high background activity and moderate self-

excitation. The unstable-α group (i.e. $\alpha > 1$) also tends to have much less rapid decay of influence from each event. Taken together, this indicates the first cluster represents conversations that see dense activity for long periods of time, followed by long periods of no activity. The second cluster is conversations that see frequent, small bursts of activity.

Figure 5.10 shows three example sequences from the data, with respectively low, median, and high estimated values of α. The non-stationarity of the third sequence ($\alpha = 1.44$) is reflected in the fact that the intensity is almost never at its baseline value. We also see the slow decay exhibited in this process observed in the previous plot. These examples bear out our expectation, that the top two sequences are characterized by frequent, small bursts of activity, while the bottom sequence is characterized by long periods of dense activity.

5.3.4 Discussion

We have shown that the persistent group conversations between individuals in a communication network, introduced in the previous chapter, are by nature temporally clustered and therefore not well-modeled by a homogeneous point process (i.e. a Poisson process). We introduced a regularized MAP EM scheme for estimating parameters under such a model (in the univariate case), using a Gamma prior and validation-selected hyperparameters. We demonstrated that this scheme works well and produces interpretable results, despite relatively small and somewhat noisy datasets. We also find that many real sequences in the data generate what appear to be non-stationary processes, violating a necessary model assumption.

This leads us to our next steps. The non-stationarity found may be due to the construction of the cascades, which requires that all events fall within a pre-defined time interval. This creates perhaps unnecessarily dense temporal clustering effects — there are "follow-on" events outside the time interval that are not captured, and may contribute to relaxed values of α and ω. As a result, we will next apply the model in a more general case, to the entire dataset, which will overcome this concern and allow us to capture all mutually exciting relationships in the data.

5.4 Multivariate case: Dyadic Network Hawkes

We now return to the more general model stated in the preliminary sections: a multidimensional Hawkes process such that each dimension (or stream) corresponds to a pairwise relationship in the network. Recall that under this model, each dimension is described by the variable intensity defined in Eq. (5.1), and we are learning the parameters $\Theta = (\mu, A)$, where $A = [\alpha_{ij}]$ represents the *influence matrix* of the network and μ is the vector of background rates. We will let ω be a global parameter controlling the rate of decay, following [70, 78, 77], and use the exponential triggering function defined in Eq. (5.3).

First we will illustrate the model with some small synthetic examples, then apply it to the mobile phone datasets from the previous chapter. Finally, we will analyze the resulting parameter estimates. We will show how the resulting influence structure from this approach differs from that in the previous chapter, show how it quantifies ideas such as the "strength of weak ties," and test its predictive power on unseen data.

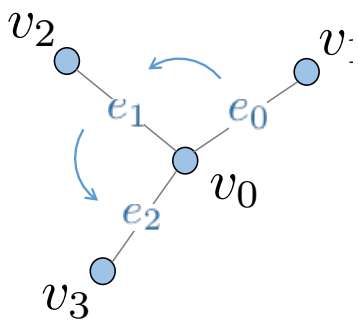


Figure 5.11: **Example network depiction.** Consider a 4-node star network where interaction between $v_0 - v_1$ increases the probability of interaction between $v_0 - v_2$, which in turn increases the probability of interaction between $v_0 - v_3$. This would create cascading series of events like we observed in the persistent cascade analysis. We can model this type of network relationship with a multidimensional Hawkes process, where each dimension corresponds to an edge in the network.

5.4.1 Example from a small network

As a small example, consider a 4-node star network with nodes $v_0, ..., v_3$, and v_0 at the center. Imagine that in this group, $v_0 - v_1$ tend to interact in bursts, and when $v_0 - v_1$ interact (edge e_0), this triggers action between $v_0 - v_2$ (edge e_1), which in turn triggers action between $v_0 - v_3$ (edge e_2). (Depicted in Figure 5.11.) This would create cascading patterns through the small network that should be evident. We can engineer such a relationship by creating our influence matrix A such that

$$\alpha_{0,0} > 0, \qquad \alpha_{1,0} > 0, \qquad \alpha_{2,1} > 0,$$

and zero elsewhere. We can make the relationship crystal clear by setting $\mu_0 > 0$ and $\mu_1 = \mu_2 = 0$, so that any events we see occur on edges e_1 or e_2 we know are due to e_0.

This is all borne out in simulation, using parameters with this setup, in Figure 5.12. We see that events occur according to some background rate in e_0, but also can "self-excite" in little bursts. We also observe that events on e_0 lead directly to spikes in the intensity on e_1, which increase the likelihood of events. This in turn creates spikes on e_2. This naturally cascading pattern is apparent in Fig 5.13 which shows only the events, without rates.

5.4.2 Findings in the mobile phone data

Network sampling method

We encounter computational limitations to apply this method directly to large-scale data: the expected branching matrix $P = [p_{ij}]$ requires $O(N^2)$ entries, where N is the number of events, which may be in the millions for even small time periods of a city-scale mobile phone dataset, and not feasible to hold in memory. We can start to minimize this limitation by noting that the only useful entries of this matrix are close to the diagonal (for example, it is highly unlikely that an event at the beginning of the month affects one at the end, for realistic triggering kernels) — this reduces the memory requirement to the order $O(N)$, using sparse matrices. We do not pursue these modifications, however, and leave this to future work.

Instead, we will use samples of the network using a *snowball sampling* approach. Specifically, we

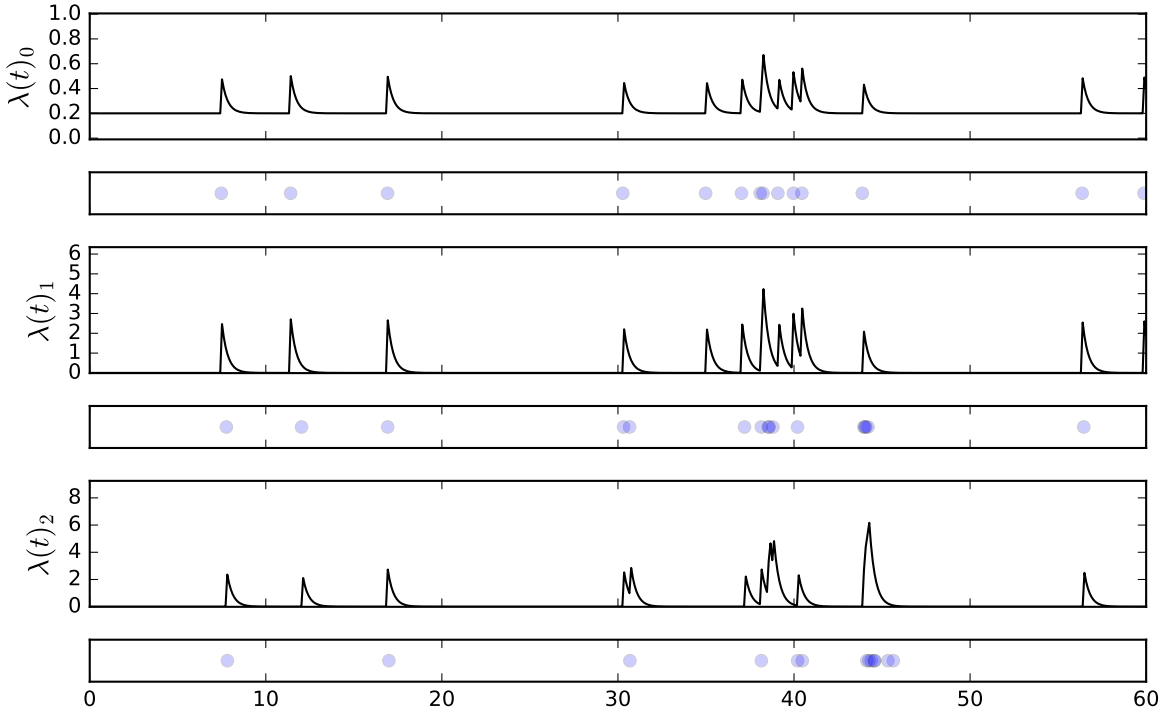


Figure 5.12: **Example network: intensities and arrivals.** Shown are events (dots) and corresponding rates (black line plots) for each edge in a toy 4-node (3-edge) star network.

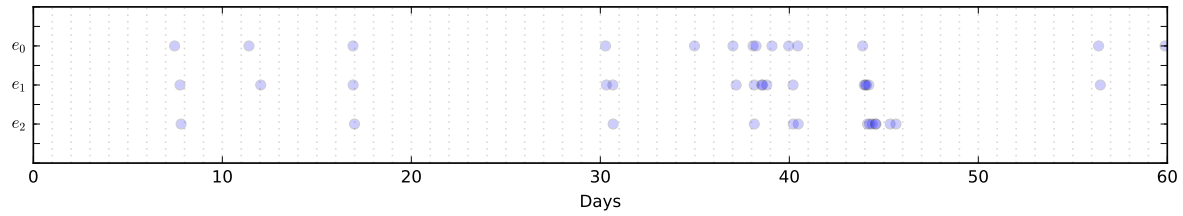


Figure 5.13: **Example network: arrivals-only.** We now plot only arrivals for the same simulation, to draw attention to the cascading tendencies in the network, beginning with e_0.

will select some node c_0, and collect the set of all individuals $\{c_1^{(i)}\}$ who communicated with c_0, then all individuals $\{c_2^{(i)}\}$ who communicated with any of the $c_1^{(i)}$, etc., to a final set $\{c_k^{(i)}\}$. This creates k "layers" around c_0, and is sometimes referred to as the *ego-k network*. (For example, the ego-1 network of a node c_0 is simply c_0 and those he contacts.)

Parameter estimation

We first select hyperparameters ω and s_0 as described in the introductory section, using a cross-validation scheme. Using 100 snowball samples of the network with the center node chosen uniformly at random from the network, we fit a multivariate Hawkes process (MHP) on 5 months of data and test its performance on an unseen validation set of 2 months of data. We use a standard grid-search approach with $\omega = 1, 2, 3, 4$ and $s_0 = 5, 10, 50, 100$. We find validation-optimal values among this set at the pair $(\omega = 4, s_0 = 50)$. Note that since we are using times on a day scale, $\omega = 4$ roughly corresponds to a decay with mean $1/4$ or about 6 hours. This corresponds to the

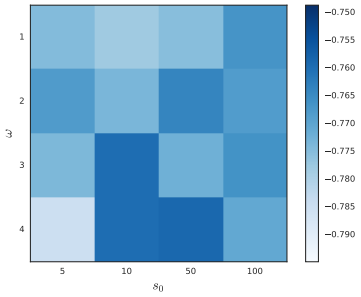


Figure 5.14: **Hyperparameter selection.** Validation log-likelihood for various combinations of hyperparameters ω and s_0.

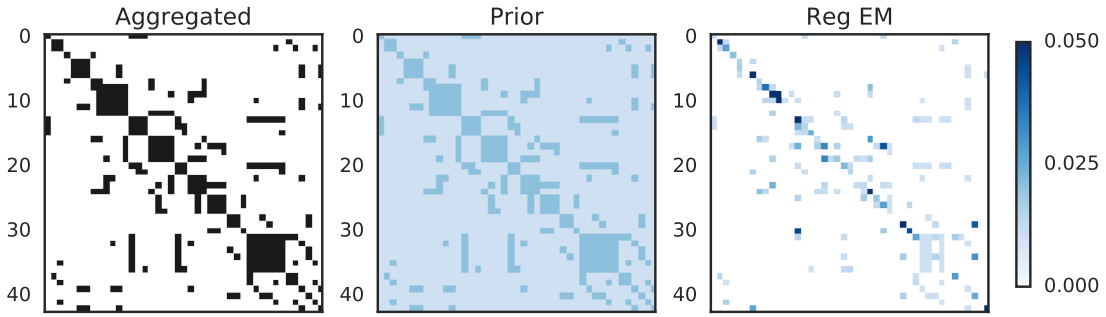
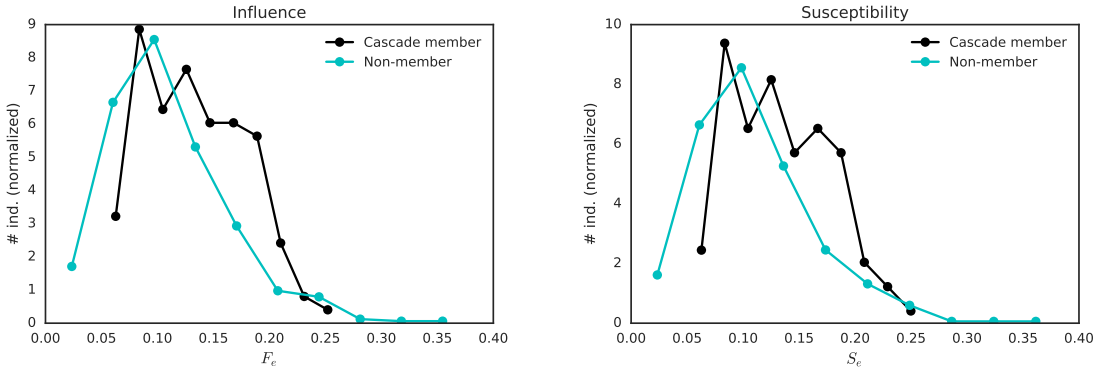


Figure 5.15: **Parameter fitting in mobile phone data.** At **left** is the adjacency matrix of the line graph of the aggregated network based on the training data; **center** is the prior on the influence matrix A; **right** is the estimated parameter values for A. Color scale is the same for the middle and right plots. Note that the prior is fairly weak, and only places a small amount of weight on the aggregated network. Note that despite this weak prior, the estimated nonzero parameter values for A largely reflect the aggregated network, although we get critical differences where the model is detecting influential relationships over edges that do not even exist in the aggregated network. This shows that the estimated A is not simply a weak copy of the prior, and is actually arising from the interactions in the data.

distribution of persistent cascades length, in a qualitative sense, that we saw in the previous chapter. We also find $s_0 = 50$, which indicates the need for a moderate prior using the aggregated network information.

We now use these hyperparameters to fit Dyadic Network Hawkes models to snowball samples of the network. Figure 5.15 shows an example of the fitted parameter values for A along with the prior and aggregated network adjacency matrix for comparison. We note that the prior is fairly weak, and only places a small amount of weight on the aggregated network. However we also note that despite this weak prior, the estimated nonzero parameter values for A largely reflect the aggregated network, although we get critical differences where the model is detecting influential relationships over edges that do not even exist in the aggregated network.



(a) (b)

Figure 5.16: **Persistent cascade members exhibit high *influence* and *susceptibility*.** Comparing distributions of (left) total *influence* and (right) total *susceptibility* as observed in the fitted influence matrix using a multivariate Hawkes process. We separate the distributions by whether the individuals were in a persistent cascade or not using the analysis from the previous chapter. We see that individuals in persistent cascades not only have higher average influence on others but also susceptibility in this model than the population in general. This is interesting because (1) it reinforces the findings in the previous chapter that these are distinct groups with quantifiably different interaction patterns, and (2) it emphasizes that cascade membership is not a forceful or one-way relationship, but an indicator that the person is more involved in his/her communication network.

Quantifying influence

We would like to focus our attention on the influence matrix A, and leave analysis of the other outputs of this model (such as the background rates μ or expected branching matrix P) for future work.

With an estimate of A in hand, what can we say about *individuals* in the network, and what can we say about *influential relationships* in the network? The MHP model provides us a quantitative measure for these questions.

Specifically, consider the value $\alpha_{uu'}$. This gives the influence that edge u' exerts on u. So, one way to measure the *influence* of a particular pair $k = (i, j)$ (or in other words how much they influence others) and how *susceptible* a particular pair $k = (i, j)$ is (or in other words how much they are influenced by others), would be to look at the average of the column and row sums of A at k, which we define:

$$F_e(k) \stackrel{\text{def}}{=} \frac{1}{U} \sum_{i=1}^{U} \alpha_{ik}, \qquad S_e(k) \stackrel{\text{def}}{=} \frac{1}{U} \sum_{j=1}^{U} \alpha_{kj} \tag{5.26}$$

where the e-subscript stands for *edge*.

Now, to examine similar qualities for a particular individual i, we simply look at the average of F_k and S_k over all $k \in \{e \in E : i \in e\}$, which we define for clarity as

$$F_n(i) \stackrel{\text{def}}{=} \frac{1}{|N(i)|} \sum_{j \in N(i)} F_e((ij)), \qquad S_n(i) \stackrel{\text{def}}{=} \frac{1}{|N(i)|} \sum_{j \in N(j)} S_e((ij)) \tag{5.27}$$

where the n-subscript stands for *node* and $N(i)$ is the set of all i's neighbors.

Comparison to Persistent Cascades

Let us compare the distributions of F_n and S_n among nodes in two classes of nodes we have already claimed have different roles: namely, between the group of individuals in a persistent cascade and those not.

For this we take repeated samples of the network, fit both the MHP model and the Persistent Cascades algorithm on the same call data, and record both the F_n and S_n values and the root nodes in the persistent cascade analysis. The resulting distributions of F_n and S_n for the two populations and on average are shown in Figure 5.16. We see that individuals in persistent cascades not only have higher average influence on others but also susceptibility in this model than the population in general. This is interesting because (1) it reinforces the findings in the previous chapter that these are distinct groups with quantifiably different interaction patterns, and (2) it emphasizes that cascade membership is not a forceful or one-way relationship, but an indicator that the person is more involved in his/her communication network.

5.5 Conclusion

In this chapter we introduced a method for modeling the interpersonal communication patterns of individuals using a multidimensional stochastic process called a Hawkes process, which is widely used in diverse modeling applications such as seismology, neuronal impulses in the brain, crime activity, and stock fluctuations. We showed that the estimated parameters governing the effect of one dimension on another can be interpreted as a matrix representation of the *influence structure* of the communication network. We extended existing work by applying the process to a *dyadic* version of the network where each dimension represents the communication between two individuals. We also propose and derive a novel method for parameter estimation using a Bayesian MAP expectation-maximization (EM) approach with a Gamma prior.

We then applied our method in the univariate and multivariate case on the mobile phone data. First, in the univariate case, we reimagined calls within a persistent cascades (as introduced in the previous chapter) as a single stream of events. We showed that clustering in the parameter space of the resulting estimated parameters reveals two broad categories of persistent cascades: one with low background rates but extremely high (and nonstationary) temporal clustering, and another with high background rates but only moderate temporal clustering.

Second, in the multivariate case, we applied the Hawkes process to an entire network. We showed that even a weak prior on the network structure is enough to reveal properties of the influence structure not apparent in a naïve, aggregated approach. We gave several examples of parameter estimation on snowball samples of the city-scale network (on the order of 30-50 individuals over the course of 4-6 months), and discussed ways to extend the method's implementation to larger samples. We introduced a simple metric for translating the estimated influence matrix into an individual metric of *influence* and *susceptibility*. We found that members of persistent cascades, as identified using the analysis in the previous chapter, tend to have both higher influence and susceptibility than non-members in the network. This is interesting because it shows that persistent cascade membership is not a forceful or one-way relationship, but an indicator that the individual is more involved in his/her communication network.

Conclusion and Future Work

6

We have proposed several novel methodologies for identifying, modeling, and predicting the structure of influence in a communication network. Our methods are applicable to a wide range of data, but we have focused on the case when content is unknown, and in particular cellular phone data, because of its pervasive and unfiltered view of a near-global sample of the population. In this chapter, we will summarize our contributions and outline avenues for future research.

6.1 Summary

6.1.1 Identifying influence structure with persistent cascades

In Chapter 4, we described a novel method for identifying and extracting temporal patterns of information spread from large-scale communication metadata, using methods of inexact tree matching and hierarchical clustering. We termed these recurring patterns *persistent cascades*, and showed that they reveal new properties of information spread and individual influence roles.

Specifically, we found the persistent cascades are present on long time scales of months to a year, and found examples of surprisingly large, recurrent structure on the scale of months. We found the patterns tend to be short in duration (over 70% last less than 3 hours), which indicates a short attention span in spreading information and echoes previous research in the "burstiness" of human communication. The individuals in a persistent cascade exhibit a habitual hierarchy, in the sense that when the same individuals communicate, they do so in the same order. We also found that our analysis reveals two new groups of individuals who have exclusive roles of information spreading on either weekends or weekdays. Individuals tend to generate more and more instances of the same pattern, and do not create new patterns, indicating predictability of communication. Lastly, we justified several of our simplifying assumptions by comparing our results against those obtained through an exhaustive search, finding that only 2% of the data is affected by our assumptions.

We demonstrated that the discovered patterns are significantly different than what is found under a random model, in both the size of the cascades and their recurrence. We accomplished this through both simulation and analytical methods. We represented the network with a so-called configuration model that matches the real degree distribution of the observed data, and approximated pairwise communication activity by sampling from a Gamma distribution fit to the actual average pairwise rates of communication. Then, extending techniques from percolation theory and epidemic modeling, we showed that these inputs (matching degree distribution and average activity

rates) are not enough to explain the persistence we observe in the data.

We then showed that these persistent cascades play a crucial role in information spreading through simulation of diffusion processes on the temporal network — specifically, members of a persistent cascade are more likely to receive information spreading through the network under realistic conditions of spreading. We gave a mathematical argument for why this is so (due to [50]), which in essence shows that when the possibility of information spread between individuals is low (low infectivity), the recurring, temporally clustered patterns we find in persistent cascades are a necessity for repeated, rapid exposure to the idea in order to achieve spreading. We also showed the effect of the analysis on our understanding of centrality, and revealed a population of super-spreaders who were otherwise unremarkable under an aggregated approach.

Lastly, we demonstrated that the method is applicable to a wide range of data by applying it to an email dataset. In this case study, we used the publicly available emails released during the government's investigation into Hillary Clinton's use of a private email server, which gives us the sender-receiver-timestamp data we need as input to the algorithm, but also a sense of "ground-truth" in terms of the email content, which we used to compare our results after the fact. We found that the persistent cascades method correctly identified key staff members, ignored known "noise" in the dataset (such as unlabeled emails), and identified several interesting persistent email chains.

6.1.2 Modeling influence structure with Hawkes processes

In Chapter 5, we introduced a probabilistic method for modeling and predicting the communication patterns of individuals, using a multidimensional stochastic process called a Hawkes process. The Hawkes process is widely used in diverse modeling applications such as seismology, neuronal impulses in the brain, crime activity, and stock fluctuations. We adapated this work by showing that the estimated parameters governing the effect of one dimension on another can be interpreted as a matrix representation of the *influence structure* of the communication network. We then extended existing work by applying the process to a *dyadic* version of the network where each dimension represents the communication between two individuals. We also proposed and derived a novel method for parameter estimation using a Bayesian maximum aposteriori (MAP) expectation-maximization (EM) approach with a Gamma prior.

We then applied our method in the univariate and multivariate case on the mobile phone data. First, in the univariate case, we reimagined calls within a persistent cascades (as introduced in the previous chapter) as a single stream of events. We showed that clustering in the parameter space of the resulting estimated parameters reveals two broad categories of persistent cascades: one with low background rates but extremely high (and nonstationary) temporal clustering, and another with high background rates but only moderate temporal clustering.

Second, in the multivariate case, we applied the Hawkes process to an entire network. We showed that even a weak prior on the network structure is enough to reveal properties of the influence structure not apparent in a naïve, aggregated approach. We gave several examples of parameter estimation on snowball samples of the city-scale network (on the order of 30-50 individuals at a time over the course of 4-6 months), and discussed ways to extend the method's implementation to larger samples. We introduced a simple metric for translating the estimated influence matrix into an individual metric of *influence* and *susceptibility*. We found that members of persistent cascades,

as identified using the analysis in the previous chapter, tend to have both higher influence and susceptibility than non-members in the network. This is interesting because it shows that persistent cascade membership is not a forceful or one-way relationship, but an indicator that the individual is more involved in his/her communication network.

6.2 Future work

The methodology presented for extracting persistent cascades makes several strong assumptions about the structure of the cascading patterns which merit further attention. For example, it is possible that a cascade does not take on a tree structure, and that it requires two individuals to initiate a cascade. It is also possible that the relevant object is not a *cascade* at all, but simply a recurring pattern of communication among social contacts. In this interpretation, it may be better to adopt the temporal graphlet mining techniques as outlined in [33, 34], but adapted to user-specific forms and not just motifs. We also note that the methods in these papers (and others on temporal motifs such as [76]) require *graph isomorphism*, and so may benefit from the relaxed, inexact matching techniques used in our approach.

We demonstrated that the persistent cascades were significantly different, in their size and recurrence, from what is expected under a random network model. In our random network, we assumed an *average rate* on each edge, or essentially modeled interpersonal interactions with a Poisson process. We showed that this was insufficient to capture the persistent patterns we observe in the data, and then proceeded in the next chapter to investigate a more rigorous model (the Hawkes process) which can capture temporal clustering through self-excitation. However, it may be possible to capture the recurrent patterns we see in the data, or close to it, using only a simple *non-homogeneous* extension of a Poisson process, similar to work in [46]. This would allow us to more faithfully represent the vast differences in interaction rates between individuals in the middle of the afternoon vs the middle of the night, for example. How much of the recurring patterns in the data can be attributed to these circadian fluctations? Can such a condition be incorporated into the analytical model presented?

We also expect there is potential in coupling these insights of communication structure with the knowledge of *mobility* that we get with many mobile phone datasets; for example, do we find high similarity of mobility patterns [67] of users within most classes? Do information spreaders exert observable influence on their social contacts' movement habits?

Regarding the Hawkes process, it merits attention to compare the predictive ability of our simplified method to the more robust framework in works like [43, 78]. It is also critical that we extend our methods to be applicable to larger datasets. This appears to be primarily a coding challenge, using the approximation outlined of using only elements of the expected branching matrix $P = [p_{ij}]$ that are along the diagonal, but it remains to be seen if this claim of a reasonable approximation can be made rigorous, or a bound put on the resulting parameter estimate. This bound should depend on the size of the sequence and the speed of the decay kernel, ω.

It would be interesting to extend the observations in this work to other models of diffusion — e.g. *opinion modeling* like voter models, linear threshold models — where there are explicit considerations for things like peer influence and thresholds of change. The classical work in this field assumes a homogeneous Poissonian sequence of interactions (see [32, 31, 2, 1]). Does the burstiness we observe in the data affect the conclusions of this work? For example, does the submodularity of the influence

maximization problem vanish when interactions are no longer homogeneous, and do we lose the bound on optimality outlined in [31]? Does burstiness affect consensus [1] even without "forceful" actors?

Our work in identifying and modeling temporal dynamics and influence structure in large-scale communication networks is a growing field. It continues to maintain importance because it touches on such a wide variety of other research topics that build on this underlying question of how individuals interact, and what that tells us about their relationships and influence on each other. We expect this interest will hold, and we hope to continue to explore these questions.

Bibliography

[1] D. Acemoglu, G. Como, F. Fagnani, and A. Ozdaglar. Persistence of disagreement in social networks. *19th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2010)*, pages 1–3, 2010.

[2] D. Acemoglu, A. Ozdaglar, and A. ParandehGheibi. Spread of (mis) information in social networks. *Games and Economic Behavior*, 70:194–227, 2010.

[3] S. Aral. Identifying Social Influence: A Comment on Opinion Leadership and Social Contagion in New Product Diffusion. *Marketing Science*, 30(2):217–223, 2011.

[4] S. Aral. The Future of Weak Ties. *American Journal of Sociology*, 121(6):1931–1939, 2016.

[5] S. Aral, L. Muchnik, and A. Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *PNAS*, 106(51):21544–21549, 2009.

[6] S. Aral and M. Van Alstyne. The Diversity-Bandwidth Tradeoff. *American Journal of Sociology*, 117(1):90–171, 2011.

[7] E. Bacry, I. Mastromatteo, and J.-F. Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(1):48, 2015.

[8] J. P. Bagrow, D. Wang, A.-L. Szló, B. Si, and Y. Moreno. Collective Response of Human Populations to Large- Scale Emergencies. *PLoS ONE*, 6(3), 2011.

[9] A. L. Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435, 2005.

[10] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(October):509–512, 1999.

[11] V. D. Blondel, A. Decuyper, and G. Krings. A survey of results on mobile phone datasets analysis. *EPJ Data Science*, 4(1):10, 2015.

[12] P. Bogdanov, M. Mongiov, and A. K. Singh. Mining heavy subgraphs in time-evolving networks. In *IEEE International Conference on Data Mining (ICDM)*, volume 1, pages 81–90, 2011.

[13] R. S. Burt. Structural holes: The social structure of competition. *Harvard University Press, Cambridge Massachussetts*, pages 38–40, 1992.

[14] R. S. Burt, W. Barnett, J. Baron, J.-A. Bendor, J. Birner, M. Bothner, F. Dobbin, C. Heath, R. Kranton, R. Khurana, J. Pfeffer, J. Podolny, H. Raider, J. Rauch, and R. Burt. Structural Holes and Good Ideas. *American Journal of Sociology*, 110(2):349–99, 2004.

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

[16] P. Domingos and M. Richardson. Mining the Network Value of Customers. In *7th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*, pages 57–66, 2001.

[17] N. Du, A. Ahmed, and A. J. Smola. Dirichlet-Hawkes Processes with Applications to Clustering Continuous-Time Document Streams Categories and Subject Descriptors. In *Proc. 21st International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*, pages 219–228, 2015.

[18] N. Du, Y. Wang, N. He, and L. Song. Time-Sensitive Recommendation From Recurrent User Activities. *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 1:1–11, 2015.

[19] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM Trans. Knowl. Discov. Data Article*, 5(4), 2012.

[20] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proc. of WSDM*, page 241, 2010.

[21] M. Granovetter. Threshold Models of Collective Behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[22] M. S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.

[23] M. T. Hansen. The search-transfer problem: The role of weak ties in sharing knowledge across organization subunits. *Administrative Science Quarterly*, 44(1):82–111, 1999.

[24] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[25] S. Huang, A. W.-C. Fu, and R. Liu. Minimum Spanning Trees in Temporal Graphs. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, number 4, pages 419–430, 2015.

[26] Y. Hulovatyy, H. Chen, and T. Milenković. Exploring the structure and function of temporal networks with dynamic graphlets. In *Bioinformatics*, 2015.

[27] J. L. Iribarren and E. Moro. Impact of human activity patterns on the dynamics of information diffusion. *Physical Review Letters*, 2009.

[28] J. L. Iribarren and E. Moro. Branching dynamics of viral information spreading. *Physical Review E*, 84(4):1–13, 2011.

[29] M. O. Jackson and L. Yariv. Diffusion of behavior and equilibrium properties in network games. *American Economic Review*, 97(2):92–98, 2007.

[30] M. J. Keeling and K. T. D. Eames. Networks and epidemic models. *Journal of the Royal Society, Interface*, 2(4):295–307, 2005.

[31] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proc. of the 9th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*, page 137, 2003.

[32] D. Kempe, J. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. *Automata, languages and programming*, pages 1127–1138, 2005.

[33] L. Kovanen, M. Karsai, K. Kaski, J. Kertész, and J. Saramäki. Temporal motifs in time-dependent networks. *J. Stat. Mech*, 2011.

[34] L. Kovanen, K. Kaski, J. Kertész, and J. Saramäki. Temporal motifs reveal homophily, gender-specific patterns, and group talk in call sequences. *PNAS*, 110(45), 2014.

[35] G. Krings, M. Karsai, S. Bernhardsson, V. D. Blondel, and J. Saramäki. Effects of time window size and placement on the structure of an aggregated communication network. *EPJ Data Science*, 1(4), 2012.

[36] P. J. Laub, T. Taimre, and P. K. Pollett. Hawkes Processes. Technical report, 2015.

[37] P. G. H. Lehot. An optimal algorithm to detect a line graph and output its root graph. *Journal of the ACM*, 21(4):569–575, 1974.

[38] K. Lerman and R. Ghosh. Information Contagion: an Empirical Study of the Spread of News on Digg and Twitter Social Networks. In *Proc. of ICSWM*, pages 90–97, 2010.

[39] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Patterns of Cascading Behavior in Large Blog Graphs. In *Proc. SIAM Intl Conf. on Data Mining*, 2007.

[40] E. Lewis and G. Mohler. A Nonparametric EM algorithm for Multiscale Hawkes Processes. *Journal of nonparametric statistics*, (1):1–20, 2011.

[41] C.-T. Li, Y.-J. Lin, and M.-Y. Yeh. The Roles of Network Communities in Social Information Diffusion. In *IEEE International Conference on Big Data*, pages 391–400, 2015.

[42] Y. Li and C. Zhang. A metric normalization of tree edit distance. *Front. Computing Sci.*, 5(1):119–125, 2011.

[43] S. W. Linderman and R. P. Adams. Discovering Latent Network Structure in Point Process Data. In *Proc. of the 31st International Conference on Machine Learning*, volume 32, 2014.

[44] S. W. Linderman and R. P. Adams. Scalable Bayesian Inference for Excitatory Point Process Networks. 2015.

[45] T. Liniger. Multivariate Hawkes Processes. *Ph.D Thesis*, pages 1–279, 2009.

[46] R. D. Malmgren, D. B. Stouffer, A. E. Motter, and L. A. N. Amaral. A Poissonian explanation for heavy tails in e-mail communication. *PNAS*, 105(47):18153–18158, 2008.

[47] M. Marder. Dynamics of epidemics on random networks. *Physical Review E*, 75(6), 2007.

[48] G. Miritello. *Temporal Patterns of Communication in Social Networks*. PhD thesis, 2013.

[49] G. Miritello, R. Lara, M. Cebrian, and E. Moro. Limited communication capacity unveils strategies for human interaction. *Nature, Scientific Reports*, 3, 2013.

[50] G. Miritello, E. Moro, and R. Lara. Dynamical strength of social ties in information spreading. *Physical Review E*, 83, 2011.

[51] C. Moore and M. Newman. Exact solution of site and bond percolation on small-world networks. *Physical Review E*, 62(5):7059–7064, 2000.

[52] M. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64, 2001.

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

[54] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, 2003.

[55] Y. Ogata. On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, 1981.

[56] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, M. De Menezes, K. Kaski, A. L. Barabási, and J. Kertész. Analysis of a large-scale weighted network of one-to-one human communication. *New Journal of Physics*, 2007.

[57] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabasi. Structure and tie strengths in mobile communication networks. *PNAS*, 104(18), 2007.

[58] P. O. Perry and P. J. Wolfe. Point process modelling for directed interaction networks. *Journal of the Royal Statistical Society B*, 75:821–849, 2013.

[59] F. Peruani and L. Tabourier. Directedness of Information Flow in Mobile Phone Communication Networks. *PLoS ONE*, 6(12), 2011.

[60] J. C. L. Pinto, T. Chahed, and E. Altman. Trend detection in social networks using Hawkes processes. *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15*, pages 1441–1448, 2015.

[61] R. Reagans and B. McEvily. Network Structure and Knowledge Transfer: The Effects of Cohesion and Range. *Administrative Science Quarterly*, 48(2):240–267, 2003.

[62] T. Shelling. Sorting and Mixing. In *Micromotives and Macrobehavior*, chapter 4, pages 137–152. 1978.

[63] A. Simma and M. I. Jordan. Modeling events with cascades of Poisson processes. In *Proc. of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 546–555, 2010.

[64] C. Steglich, T. A. B. Snijders, and M. Pearson. Dynamic Networks And Behavior: Seperating Selection From Influence. *Sociological Methodology*, 8:329–393, 2010.

[65] A. Stomakhin, M. B. Short, and A. L. Bertozzi. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems*, 27(11):115013, 2011.

[66] I. M. Toke. An Introduction to Hawkes Processes with Applications to Finance. Technical report, 2011.

[67] J. L. Toole, S. Colak, B. Sturt, L. P. Alexander, A. Evsukoff, and M. C. González. The path most traveled: Travel demand estimation using big data resources. *Transportation Research Part C: Emerging Technologies*, 58:162–177, 2015.

[68] J. L. Toole, C. Herrera-Yaque, C. M. Schneider, and M. C. González. Coupling human mobility and social ties. *Journal of the Royal Society, Interface*, 12:1–14, 2015.

[69] B. Uzzi. Social Structure and Competition in Interfirm Networks: The Paradox of Embeddedness. *Administrative Science Quarterly*, 42(1):35–67, 1997.

[70] I. Valera and M. Gomez-Rodriguez. Modeling Adoption and Usage of Competing Products. In *IEEE International Conference on Data Mining (ICDM)*, 2015.

[71] A. Vazquez, B. Rácz, A. Lukács, and A. L. Barabási. Impact of non-poissonian activity patterns on spreading processes. *Physical Review Letters*, 98, 2007.

[72] A. Veen and F. P. Schoenberg. Estimation of Space–Time Branching Process Models in Seismology Using an EM-Type Algorithm. *Journal of the American Statistical Association*, 103(482):614–624, 2008.

[73] D. J. Watts. Networks, Dynamics, and the Small-World Phenomenon. *American Journal of Sociology*, 105(2):493–527, 1999.

[74] D. J. Watts. A simple model of global cascades on random networks. *PNAS*, 99(9):5766–5771, 2002.

[75] K. Zhang and D. Shasha. Simple Fast Algorithm for the Editing Distance between Two Trees and Related Problems. *SIAM J. Computing*, 18(6):1245–1262, 1989.

[76] Q. Zhao, Y. Tian, Q. He, N. Oliver, R. Jin, and W.-C. Lee. Communication Motifs: A Tool to Characterize Social Communications. In *Proc. of CIKM*, 2010.

[77] K. Zhou, H. Zha, and L. Song. Learning Social Infectivity in Sparse Low-rank Networks Using Multi-dimensional Hawkes Processes. In *Proc. of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 641–649, 2013.

[78] K. Zhou, H. Zha, and L. Song. Learning Triggering Kernels for Multi-dimensional Hawkes Processes. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1301–1309, 2013.

[79] J. R. Zipkin, F. Schoenberg, K. Coronges, and A. Bertozzi. Point-process models of social network interactions: parameter estimation and missing data recovery. *Euro Journal of Applied Mathematics*, 1, 2014.