The Dissertation Committee for Philip Christopher Milling
certifies that this is the approved version of the following dissertation:

Identifying Infection Processes with Incomplete Information

Committee:

Sanjay Shakkottai, Supervisor

Constantine Caramanis

David Morton

Sujay Sanghavi

Gustavo de Veciana

Identifying Infection Processes with Incomplete Information

by

Philip Christopher Milling, B.S., M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2014

Dedicated to my family.

Acknowledgments

I would first like to thank my co-authors on the papers upon which this thesis is based, Prof. Constantine Caramanis, Prof. Shie Mannor, and Prof. Sanjay Shakkottai. They were extremely helpful in writing each paper, supplying many ideas and suggestions, including some of the algorithms considered in this paper. Their help with writing/editing the introduction and problem statement was especially valuable, as was their assistance in improving the clarity and precision of the language throughout.

I am extremely grateful to my advisor, Prof. Sanjay Shakkottai, for his enthusiastic support and guidance. His help has been invaluable, from formulating new research problems to assistance in any administrative issues. He always motivated me to think about my problems in new ways, and I credit many of my accomplishments to his assistance. Whenever I had difficulty, he always provided value insight and advice to help me resolve any issues.

I am also thankful to Prof. Constantine Caramanis. His support and feedback has been helpful and insightful, and he has always encouraged me to expand my horizons. It has been a privilege to work with him on the papers underlying this thesis. I would like to thank Prof. Sujay Sanghavi for participating in my committee, as well as for his thought-provoking class I had the privilege of taking. I would like to thank Prof. Gustavo de Veciana as well for being a committee member and for being an enjoyable teacher. I would also like to thank Prof. David Morton for being a committee member.

Finally, I would like to thank my family and friends. Your encouragement and occasional nagging has helped motivate me through this long journey. I never would have made it here without your support.

Identifying Infection Processes with Incomplete Information

Publication No. ____________

Philip Christopher Milling, Ph.D.
The University of Texas at Austin, 2014

Supervisor: Sanjay Shakkottai

Infections frequently occur on both networks of devices and networks of people, and can model not only viruses, but also information, rumors, and product use. However, in many circumstances, the infection process itself is hidden, and only the effects, e.g. sickness or knowledge, can be observed. In addition, this information is likely incomplete, missing many sick nodes, as well as inaccurate, with false positives. To use this data effectively, it is often essential to identify the infection process causing the sickness, or even whether the cause is an infection. For our purposes, we consider the susceptible-infected (SI) infection model. We seek to distinguish between infections and random sickness, as well as between different infection (or infection-like) processes in a limited information setting.

We formulate this as a hypothesis testing problem, where (typically) in the null, the sickness affects nodes at random, and in the alternative, the infection is spread through the network. Similarly, we consider the case where the sickness may be caused by one of two infection (or infection-like) processes, and we wish to find which is the causative process.

We do this is a setting with very limited information, given only a single snapshot of the infection. Only a small portion of the infected population reports the sickness. In addition, there are several other limitations we consider. There may be false positives, obfuscating the infection. Similarly, there may be a random sickness and epidemic process occurring simultaneously. Knowledge of the graph topology may be incomplete, with unknown edges over which the infection may spread. The graph may also be weighted, affecting the way the infection spreads over the graph. In all these cases, we develop algorithms to identify the causative process of the infection utilizing the fact that infected nodes will be clustered. We demonstrate that under reasonable conditions, these algorithms detect an infection with asymptotically zero error probability as the graph size increases.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Research into social networks garnered increased interest in recent years. These networks can represent relationships between people, devices, and companies. In an age where online networking services such as Twitter and Facebook play an increasingly ubiquitous role is peoples' lives, understanding social networks is now more vital than ever before. These services both make social networks more important, but also makes analyzing them more feasible and useful. There is a massive amount of information available about the interactions between people, as well as about the individuals themselves, that can analyzed to provide superior service, better targeting of ads, and many other applications. Though the number of ways to approach and analyze this data is practically unlimited, this thesis focuses mainly on developing our understanding of infections over such graphs.

An infection on a network is a simple representation of a process where some state spreads from one person/entity to another. This is clearest in a standard infection: a biological or computer virus. A virus spreads from an infected person or device to another person/device they are connected to. For people, this would be from one person to another person they spend significant amount of time with. That is, it spreads between people in close social contact. Likewise, for computers and other devices, viruses can spread through physical networks, but also through social networks. For example, many computer virus exploit the trust people have in email that appears to be from acquaintances to infect additional people. These infections can be modeled as nodes (people or devices) spreading an infection over a network. This network may be real life social contacts, Internet social networks, physical networks, etc. We use the SI infection model, where the infection spreads at a constant rate across the edges of a graph, and once a node becomes infected, it never recovers.

There are two classes of approaches to understanding infections, which can be termed the forward and backward problems. In the forward problem, the goal is to understand how the infection spreads over the social network. Topics in this area include understanding the speed and size of infections and determining how the shape of the network impacts the infection's spread. Considerable effort has been devoted to such problems. On the other hand, the backward problem involves trying to infer properties of the infection when given the resulting infection. Relative to the previous class of problem, work on this topic is lacking. This is despite of the number of potential applications for this approach, especially with the amount of data readily available on many social networks.

In many cases, the key question is, is there an infection occurring and what is its causative network? Prior work on topic has focused on a high information regime, where one has knowledge of the entire infection process, and possibly even of multiple infections. The focus of thesis is on the low and unreliable information regime. Information on the infection is sparse, and even unreliable, characteristics of many practical data sets. Under these conditions, the goal is to distinguish between two candidate hypothesis for the infection process. These infection processes may be random sicknesses, or spread from node to node on a graph. We refer to the latter case an an *epidemic*.

There are many applications to this problem. In the case of an illness in a population, the ability to distinguish between a mostly random sickness (such as the common cold) and a very infectious illness (such as the flu) can be invaluable. Early detection of such infections could lead to faster and more efficient resources deployment, earlier warnings for the population and similar benefits. This is likewise true for device malfunctions. They may be due to part defects, or be caused by malware spreading over the network. Again, distinguishing between these two cases would be helpful in diagnosing and thereafter solving the problem.

A similar application can be found in the case of advertising. For instance, suppose there is a Facebook ad promoting some product. If the advertisement is effective, product usage will spread over the Facebook social network. In this case, we want to know if the the advertisement led to a significant increase in the popularity of that product. Identifying an epidemic on that network as a significant contributor to increased product use in this case would mean that the advertisement was effective.

1.1 Main Contributions

We develop algorithms to determine the causative infection process between two alternative possibilities. We do this for several low information regimes, as well as different types of infection processes. Our approach relies on utilizing the clustering of the sick nodes on the infection graph. When the sick nodes are clustered on a graph, the nodes are likely the result of an epidemic on that graph.

We evaluate our algorithm performance by the asymptotic error probability. In particular, we are interested in the range of infection sizes for which the error probability tends to 0 as the graph size increases. Note that once the entire graph is infected, it is impossible to distinguish between different infection processes since there is no topological information. Likewise, when the infection contains only a small number of sick nodes, it is likely no node reports an infection, and again solving the problem is impossible. For each of our algorithms, we demonstrate sufficient conditions on the infection size (and other problem parameters) so that the error probability vanishes asymptotically. Our conditions are generally lenient, and in some cases, they are order-wise optimal in the infection sizes for which they succeed. These are supported by simulations. The simulations also provide intuition on the behavior of the error probability as the algorithm parameters vary.

The fundamental problem we consider first is where the infection is either due to a random sickness (nodes are sick independently with identical probability), or an epidemic (the infection spreads from node to nodes across the graph). We develop two algorithms to distinguish between these infections processes. In the first, we evaluate the likelihood that the sick nodes are from an epidemic by the size of the smallest ball containing the nodes. In the second, we rate the probability of an epidemic by the size of the smallest tree containing all the sick nodes. We evaluate these on three standard graphs: grids, trees, and Erdös-Renyi graphs. We find that in most cases, the ball based algorithm is superior or equal to the tree algorithm.

The next case we consider is when the infection spreads on one of two different graphs. We require these graphs to satisfy basic topological constraints satisfied by many standard graphs. By comparing the relative clustering on each graph using the minimum containing ball's size as before, we demonstrate it is possible to determine the correct infection graph with high probability.

We extend our algorithm to be more robust by eliminating outliers. We apply this algorithm to the case when there are false positives, both random and adversarial. Another variation we consider is distinguishing between two mixed infection processes, where a random sickness and epidemic processes occur simultaneously. However, in one process, the epidemic is the dominate process. A similar robust algorithm is shown to succeed in this case as well. In addition, we examine when some edges on the graph are not known, which we show can also be solved by our algorithm.

The final case we consider is weighted graphs. In these graphs, the infection may spread at different rates between different pairs of nodes. We develop a modified algorithm that uses the size of the largest ball that contains a minimum density of sick nodes. This algorithm is shown to achieve asymptotically zero error probability and in simulations, may perform better than previous algorithms.

1.2 Thesis Outline

In Chapter 2, we consider the fundamental problem of distinguishing a random sickness from an epidemic. In Chapter 3, there are two different epidemics that must be distinguished. In Chapter 4, we analyze the case when there are false positives. The problem we consider in Chapter 5 is when the infection processes are a mixture of random sicknesses and epidemics. In Chapter 6, we examine the case when some edges of the infection graph are unknown. Finally, we analyze infections on weighted graphs in Chapter 7. Our conclusion and opportunities for future work are presented in Chapter 8.

Chapter 2

Fundamental Problem

2.1 Introduction

It is vital to understand and identify the spread of infections through networks, social and physical, in order to respond appropriately, whether through quarantines, predicting future spreads, planning for future actions, etc. In these circumstances, the key challenge is to understand the process by which the infection is spreading with limited information available.

The situation we consider is when the infection is observed at a particular time. Other time related information, particularly the time when each sick becomes infected, is not available. Likewise, it is not known which node infects which other nodes. Without the time history of the infection, it is impossible to directly determine how the infection spread, including determining the underlying network. However, the set of sick nodes can be used to evaluate whether a provided hypothesis for the infection process is likely.

We suppose that there are two candidates for the process by which the

The work in this chapter appears in the following publication:
Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. Network forensics: random infection vs spreading epidemic. *SIGMETRICS Perform. Eval. Rev.*, 40(1):223–234, June 2012.

infection is spreading, one of which represents the true infection process. The infection is either the result of a random sickness, or an epidemic. The parameters of these hypotheses are fully specified to the algorithm. In particular, the full infection network is known. The collection of sick nodes is used to evaluate both processes to determine which is the most likely cause of the infection.

In an epidemic, the sickness travels along the edges of the network from a source and results in a clustered infection. On the other hand, in a random sickness, each node is sick randomly and independently with some probability. Then there is no structural relationship between sick nodes in a random sickness. Exploiting this characteristic helps us determine whether or not a collection of sick nodes represents an epidemic. For example, there may be many people with flu-like symptoms being treated in some area. We would like to determine whether this represents an actual outbreak of the flu, or just an occurrence of several colds, using the topological information in the set of sick people.

If the full set of infected nodes were known, this problem would be relatively simple to solve in most circumstances. Simply testing the connectivity of the nodes would be sufficient. However, real data is almost never so complete. Thus, we assume the knowledge of the sick nodes is only partial. This information may be available from self-reports of the infection, which will necessarily not include all the nodes. For example, only a portion of people infected with an illness may go to the doctor, or only a portion of those infected may be correctly identified. Alternatively, if the data is from a survey, it is impossible to reach every person. The fact that the knowledge of the infected nodes is incomplete must be considered to accurately discover the spreading mechanism of viruses. This limitation can be modeled by having each node decide randomly whether to report the infection.

We phrase this as a hypothesis testing problem. Our null hypothesis is that the infection is caused by a random sickness. On the other hand, the alternative hypothesis is that the infection resulted from an epidemic. In a Type I error, a random sickness is mistaken as an infection, because for example, the randomly sick nodes were grouped like an infection. A Type II error is when an infection is incorrectly diagnosed as a random sickness, often because the infection has grown too large. Figure 2.1 provides examples of when a Type I and a Type II error might occur.

In this chapter we consider only this most basic formulation. The algorithms we develop represent a fundamental and simple solution to the problem of identifying the correct infection process. Later chapters cover extensions of this problem in additional limitations on the information available, and for other infection models. These will build on the approach developed here.

2.1.1 Contributions

We develop several algorithms to solve these problems. These algorithms use the clustering of the sick nodes to estimate the most likely causative process from the two hypotheses. We term these algorithms the *Threshold Ball Algorithm* and the *Threshold Tree Algorithm*. These algorithms use the small-

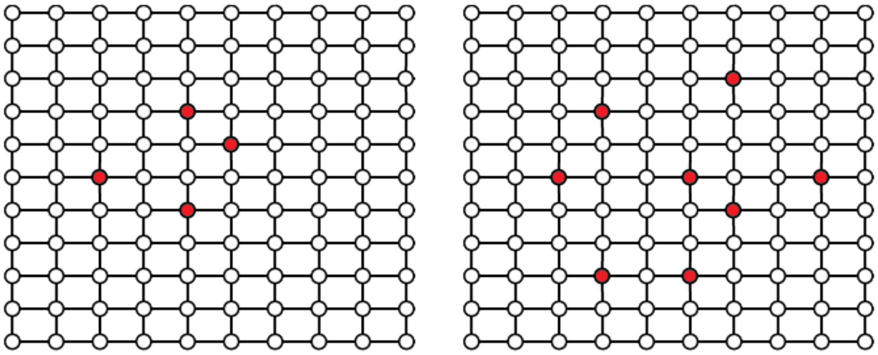


Figure 2.1: Grid graphs with infected nodes red. The left-hand graph shows a possible Type I error, with randomly sick nodes unfortunately clustered. If there are very few reporting sick nodes, such errors are impossible to rule out, hence our results impose an assumption that at least $\log n$ nodes report. The right-hand graph shows a possible Type II error, where the infection has spread out considerably, and the many false negatives make the infection appear like a random sickness. If the infection has spread too far, such errors are again difficult to rule out, hence our results provide guarantees in the presence of upper bounds on the number of infected nodes.

est ball and smallest tree, respectively, that contain all the reporting sick nodes as the key measurement to evaluate whether the sickness is random, or due to an epidemic. These algorithms compare that measurement to a calculated threshold to determine whether the sick nodes are clustered like an epidemic. We prove that for a reasonable range of infection sizes, these algorithms have a probability of error that tends to 0 as the infection size increases.

2.1.2 Related Work

Most of the work on the susceptible-infected (SI) infection model has focused on understanding the spread and speed of infection, the analytic side of infections. These infections have been analyzed has been for a variety of settings, such as for graph with both local and global spreading [4], and even where infected nodes are mobile [24]. Though our problem is not on this analytic side, we leverage many such results in our proof.

A majority of the work on the inference side of infections is on estimating various parameters of the infection. Demiris and O'Neill estimate the infection rate using time data for the complete set of infected nodes [15, 16]. One method to doing this is to use a Bayesian inference approach [15]. Another interesting approach is to use Markov chain Monte Carlo (MCMC) methods to estimate the infection rate [16, 58].

A related idea is inferring the source of the infection. Shah and Zaman develop an algorithm to determine the most likely source of an infection given the complete set of infected nodes [56]. The method can be efficiently cast as a belief-propagation algorithm to find the maximum-likelihood estimator for tree. It can be applied to general graphs by approximating them by breadth first search (BFS) trees to estimate the infection source. They show through simulations that the estimated infection source is often close to the true infection source for several types of graphs. Following this work, there have been extensive studies on this problem of determining an infection's source in various contexts [55, 37, 38, 17, 32, 36, 63].

Netrapalli and Sanghavi consider the related problem of estimating the graph structure using the infection information [51]. In this instance, no knowledge regarding the network structure is provided. However, they use the full set of infected nodes, as well as the time information regarding when each node is infected. Multiple infections are performed on this same graph, which allows their algorithm to detect patterns in infection times, and thereby infer a graph structure similar to the true network. They develop both a maximum-likelihood and greedy algorithm, and establish probability of error upper bounds based on the number of samples.

A similar problem is also considered by Gomez-Rodriguez [22] with random incubation times (the time from when a node becomes infected until it can infect its neighbors). They develop an algorithm that approximates the maximum-likelihood graph and bound the log-likelihood distance from the optimum graph. The ability to determine the full graph structure is similar to determining the true infection process considered here, except that candidate processes are not required. However, this approach requires extreme amounts of information that is not fully available in many contexts. First, the full infection set with time stamp information is required, as opposed to only a partial infection set with no time information. Second and perhaps more problematic, this data must be obtained over many infections. Then though their solution may be considered more powerful, their data requirements make it impossible to use in the minimal information regime.

An alternative interpretation of our problem is that we seek to determine if any of the likely ‘infection shapes’ (from the set of infected nodes) explain the known sick nodes. From this perspective, our work is closely related to the problem in [1], [2]. In that work, the authors consider a hypothesis testing problem where every node reports an i.i.d standard normal random variable, except (in the alternative hypothesis) for a cluster of nodes reporting a normal with positive mean, from a class of possible clusters.

2.2 Model and Algorithms

In this section, we formally specify the problem details. The models described here form the foundation of the rest of this thesis.

2.2.1 Infection Model

We consider an infection spreading on a graph $G = (V, E)$, where $n = |V|$, the number of nodes. The infection spread according to the standard susceptible-infected (SI) infection model [20]. Initially, at time 0, a random node on the graph is selected to be infected. This node is the infection source. For each edge connected from the infected node to a susceptible node, a clock is started that expires after duration which is independent and exponentially distributed with rate 1. After the clock for an edge expires, the adjacent susceptible node (if it has not been infected already along a different edge) becomes infected. Then, new clocks are started for each edge between this node and adjacent susceptible nodes in the same way as before. In this way, the infection spreads at rate 1 through the network until time t has passed.

In addition, we also consider a random infection. In this case, the time t is not used. Rather, each node independently becomes sick with a probability q'. Then the expected number of sick nodes is determined by q' rather than by time t, and is equal to $q'n$. By setting q' appropriately, the expected infection size can be normalized to a desired value.

Though we phrase the random sickness as being fixed in time, it is also possible to imagine it as an infection spreading on a complete graph. Since there is no structure distinguishing different nodes, the resulting sick nodes would appear random. In this sense, distinguishing a random sickness from an infection is the same problem as distinguishing two different infection graphs. The main difference is that, even if the expected number of sick nodes is the same, a random sickness has less variance than an infection on a complete graph.

2.2.2 Reporting Model

After the infection proceeds for time t, a subset of the infected nodes report. The reporting mechanism is simple, and identical for both processes. At this time, each sick nodes decides to report their sickness independently with probability q. We define the full set of sick nodes as S, and the set of reporting nodes as S_{rep}. Therefore, $E[|S_{\text{rep}}|] = qE[|S|]$. If q is very small, then we may only have a very small proportion of the sick nodes report, which is the most difficult setting for this problem. In our theorems, we only require that a logarithmic (in n) number of sick nodes report, even if the entire infection is much larger. Both t and q may depend on n, and we write $t^{(n)}$ and $q^{(n)}$ when it is necessary to make this dependence clear.

2.2.3 Normalization

The goal in solving this problem is to use the 'shape' of the sick nodes to determine the causative infection process. To highlight this, it is necessary to remove other factors that could be used when possible. In particular, we must remove differences in the expected number of infected nodes. If the expected infection size was significantly different in each process, the infection size itself would suffice to distinguish the two processes. For this reason, we want to match to expected infection size. For the case of a random sickness, q' (the probability of a node being sick) is set so that $q'n$ is equal to the expected size of the epidemic.

2.2.4 Graphs

In order to analyze the asymptotic performance of our algorithms, we consider infinite families of graphs, where the graph size has no upper limit. Formally, we denote a family of graphs as $\mathcal{G} = \{\mathcal{G}^{(n)}\}$. Each $\mathcal{G}^{(n)}$ is a collection of graphs $G^{(n)}$, each of degree n. For each of these, there is a (possibly trivial) probability space $\left(\mathcal{G}^{(n)}, \sigma(\mathcal{G}^{(n)}), P^{(n)}\right)$. A series of graphs $\{G^{(n)}\}$ is chosen from $\prod_n \mathcal{G}^{(n)}$, and an infection spreads on each graph as described as above. Examples of families are d-dimensional grids, Erdös-Renyi graphs, and trees. An mentioned previously, the infection time $t^{(n)}$ and reporting probability $q^{(n)}$ may depend on the graph size. We are interested primarily in the properties of the infection as $n \to \infty$. For additional clarity in our results, we drop the superscript (n) when the n is clear from context.

For this problem, we consider standard graph topologies representative of the typical social networks. There are two main types of topologies to consider. The first type are geographic topologies. These topologies are for social networks where social contact is primarily from geographic proximity. The distinguishing aspects of these graphs are the large number of local cycles and absence of long range edges. That is, they have a relatively large diameter. A convenient representative from these graphs are multidimensional grid graphs. These graphs are be represented as a lattice, where adjacent points of the lattice are connected by an edge. Such graphs exhibit the required properties and are simple enough to analyze. We connect the opposing sides forming a torus to avoid edge effects.

The second type of topology that must be represented is a tree-like social network. These topologies have much lower diameters (logarithmic in the size of the graph). Most social networks, especially over the Internet, fit in this category. In addition, this problem is much more difficult on these types of graphs because the infection spreads much faster. A suitable representative for this graph topology is an Erdös-Renyi graph. An Erdös-Renyi graph is formed by starting with a graph with no edges. Then an edge is randomly added between each pair of nodes independently with a fixed probability. This type of graph will exhibit the desired properties when the edge probability is set appropriately. We also consider trees, which have the same local structure as Erdös-Renyi graphs. The performance of our algorithms on each these graphs also provides an insightful contrast.

These topologies represent our reference topologies for the purpose of determining how the graph structure determines the performance of our algorithms. In addition, we perform simulations on these graphs, as well as graphs from real data, to evaluate our empirical performance and support our theorems.

2.2.5 Error Probability

We assume the prior probability of both processes are equal. We label the random sickness as Process 0 and the epidemic as Process 1. We phrase the error probability in the language of hypothesis testing. The null hypothesis H_0 is that Process 0 is the true infection process. Correspondingly, the alternative hypothesis H_1 is that Process 1 is the true infection process. The error where the infection is caused by Process 0, but we label it Process 1, is termed a Type I error. Likewise, when the infection is caused by Process 1, but we believe it is caused by Process 0, it is a Type II error. Then the overall error probability is the average of the Type I and Type II error probabilities.

Another major question to be resolved is how the algorithm's performance should be judged. The goal is to choose the correct infection process with the minimum probability of error. One possible measure is the asymptotic error rate of the algorithm. However, for many graphs, the error probability does not decrease exponentially in the graph size, and the probability is highly dependent on the expected infection size. The objective however is to establish a clear range of parameters for which this problem can be solved, so this measure is unsuitable.

For this reason, the performance of the algorithm will be measured in the range of parameters (such as infection size) for which the probability of error decays to 0 as the graph size increases without bound. Equivalently, we are interested when both the Type I and Type II error probability decays to 0. That is, the algorithm is measured by the range of parameters for which the error probability eventually is low. Though the error probability may decay slowly, this is a straight forward condition on when the algorithm succeeds, allows for clear valid parameter ranges, and is relevant for all graph topologies.

2.2.6 Algorithms

The key idea we use is that when the sickness is due to an epidemic, the sick nodes will be clustered on the graph. On the other hand, in a random sickness, the nodes will be spread out evenly over the network. However, there are multiple ways to measure clustering of sick nodes.

We use two methods to rate the clustering, 'ball clustering' and 'tree clustering'. These are the basis of the two algorithm we consider, the *Threshold Ball Algorithm* and *Threshold Tree Algorithm* respectively. The idea of these clustering is as follows. For 'ball clustering', we look at the smallest radius ball that contains all the sick nodes. The radius of this ball acts as a 'score' for the level of clustering. If the radius is small, then the sick nodes are well clustered. On the other hand, if the radius is close to the radius of the entire graph, then the sick nodes are heavily separated. For 'tree clustering', the 'score' is the number of nodes in the smallest tree containing all the sick nodes. In this case, the measure can be also be phrased as the smallest possible infection that could have resulted in the set of reporting sick nodes. Once the score is determined, it is compared against a threshold determined either by the infection time t or the number of reporting nodes.

To define our algorithms, we use the following definitions. With a graph G, a node v, and radius r, we use $\mathrm{Ball}_{v,r}(G)$ to denote the collection of all nodes on G that are at a distance of no more than r from the central node v, where graph distance is measured by hop-count. For any collection of nodes S, we now denote by $\mathrm{Ball}(G, S)$ the smallest-radius ball that contains all the nodes in S, and we let $\mathrm{BallRadius}(G, S)$ denote its corresponding radius. Finally, let $\mathrm{Tree}(G, S)$ be the smallest subtree of G containing all nodes in S, and $\mathrm{TreeSize}(G, S)$ be the number of nodes in this tree. The algorithm to determine $\mathrm{BallRadius}(G, S)$ can be specified simply as follows.

Determining the size of the smallest tree containing all the sick nodes is a more difficult problem. This tree is the minimum Steiner tree [29], and finding it is an NP-hard problem. However, there are efficient algorithms that give approximate solutions, guaranteeing no more than twice the optimum number of nodes or better [44, 26].

From these measures of clustering, we can then define the *Threshold Ball Algorithm* and *Threshold Tree Algorithm*. As mentioned, these algorithms compute a 'score' rating the clustering of the sick nodes appear. If the score is below a specified threshold (which would be set using the infection time t),

Algorithm 1 BallRadius

```
Input: Graph $G$; Set of reporting sick nodes $S_{\text{rep}}$;
Output: Radius $r$

$k \leftarrow \infty$
for all $v \in V$ do
    $d \leftarrow 0$
    for all $u \in S_{\text{rep}}$ do
        if $\text{dist}(u, v) > d$ then
            $d \leftarrow \text{dist}(u, v)$
        end if
    end for
    if $d < k$ then
        $k \leftarrow d$
    end if
end for
return $k$
```

then the sick nodes are sufficiently clustered and algorithm labels the sickness an epidemic.

Since it may not be possible to know the duration of the epidemic, we also consider adaptive versions of these algorithms. In this case, the threshold is determined by using the number of infected nodes and the graph topology to estimate the infection time. With a sufficiently accurate estimate of the infection time, the threshold can be computed as before. These algorithms are analyzed in a similar way as the basic threshold algorithms.

Algorithm 2 Threshold Ball Algorithm

Input: Graph G; Set of reporting sick nodes S_{rep}
Parameters: Threshold m
Output: EPIDEMIC or RANDOM

```
$k \leftarrow \text{BallRadius}(G, S_{\text{rep}})$
if $k \leq m$ then
  return EPIDEMIC
else
  return RANDOM
end if
```

Algorithm 3 Threshold Tree Algorithm

Input: Graph G; Set of reporting sick nodes S_{rep}
Parameters: Threshold m
Output: EPIDEMIC or RANDOM

```
$k \leftarrow \text{TreeSize}(G, S_{\text{rep}})$
if $k \leq m$ then
  return EPIDEMIC
else
  return RANDOM
end if
```

2.3 Results

We prove that the probability of error tends to 0 for a reasonable range of infection sizes. For grid graphs, the sufficient conditions to guarantee low probability of error for the Threshold Ball Algorithm are looser than those for the Threshold Tree Algorithm. That is, the Threshold Ball Algorithm seems to perform better than the Threshold Tree Algorithm on a grid, and our simulation results reflect this as well. However, we have not proven necessary conditions that confirm this result. For tree graphs, our results suggest that the Threshold Tree Algorithm is slightly superior to the Threshold Ball Algorithm. On Erdös-Renyi graphs, the conditions are similar, but empirically, the Threshold Ball Algorithm performs somewhat better. Overall, the Threshold Ball Algorithm performs better and is much more efficient.

2.3.1 Grids

First we analyze the performance of our algorithms on grid graphs. Let the graph $G = Grid(n, d)$ be such a grid network with n nodes and dimension d, so the side length is $n^{1/d}$. We avoid edge effects by adding edges that wrap from one side to the other, which makes the graph a torus. This modification allows us to avoid dealing with additional complexities resulting from the choice of the initial source of the infection.

In order to evaluate our algorithms on a grid, we need to understand the expected shape of the epidemic. Since we model the time it takes the infection to traverse an edge as an independent exponentially distributed random variable, the time a node is infected is the minimum sum of these random variables over all paths between the infection origin and that node. This simply phrases the infection process in terms of first-passage percolation on this graph. This allows us to use a result characterizing the 'shape' of an infection on this graph (see [34]). Let $I(t)$ be the set of infected nodes at time t. Identifying the nodes of the graph with points on the integer lattice embedded in $\mathbb{R}^d$ with the infection starting at the origin, let us put a small ℓ^∞-ball around each infected node. This allows us to simply state inner and outer bounds for the shape of the infection. To this end, define this expanded set as $B(t) = I(t) + [-1/2, 1/2]^d$.

Lemma 2.3.1 ([34])**.** *There exists a set B_0 and constants C_1 to C_5 such that for $x \leq \sqrt{t}$,*

$$P\{B(t)/t \subset (1 + x/\sqrt{t})B_0\} \geq 1 - C_1 t^{2d} e^{-C_2 x}$$

and

$$\begin{aligned} &P\{(1 - C_3 t^{-1/(2d+4)}(\log t)^{1/(d+2)})B_0 \subset B(t)/t\} \\ &\quad \geq 1 - C_4 t^d \exp\left(-C_5 t^{(d+1)/(2d+4)}(\log t)^{1/(d+2)}\right). \end{aligned}$$

That is, the shape of the infected set $B(t)$ can be well-approximated by the region tB_0. In addition, the variation of the edge is on the order of $\sqrt{t}$ or less.

Moreover, one can show that this set B_0 is symmetrical and convex. Define $\mu \triangleq \sup_x\{(x, 0, ..., 0) \in B_0\}$. That is, μ is effectively the rate the infection spreads along an axis. Then B_0 contains an ℓ^1-ball and is contained in an ℓ^∞ ball: $\{x : \|x\|_1 \leq \mu\} \subset B_0 \subset [-\mu, \mu]^d$. Note that μ does not depend on the *realization* of the process, only the dimension of the grid. Though this result is for infinite grids, it applies to the torus case as well. One way to see this is to label the nodes of an infinite grid '1' to 'n' so that all nodes where each coordinate is the same modulo $n^{1/d}$ have the same label, forming an infinite pattern of the size n torus. Since the non-self-intersecting paths on the torus correspond to such paths on this infinite grid, and the infection time of a node is the minimum traversal time over all such paths, the infection on the torus spreads no faster than it does on the infinite grid. In addition, we consider only infection times sufficiently small that edge effects do not come into play.

The second result we need is to show that the number of reporting sick nodes is close to the expected number. This follows from the following well-known Chernoff bound.

Lemma 2.3.2. *If at least s nodes are sick, then the number of reporting nodes will be at least $(1-\delta)qs$ with probability at least $1 - \exp(-(1-\delta)^2 qs/2)$.*

For each result, we first present sufficient conditions when the threshold is based on the time t. When the time is known, the estimated expected infection size and spread can be determined, which allows the threshold to be set more accurately. However, the infection duration or speed would often not be known. In that case, the infection time can be estimated from the number of reporting infected nodes, and the threshold can be set using this estimation. With the adaptive thresholds, the maximum infection size in our sufficient conditions is typically reduced by a factor of $\log n$.

Theorem 2.3.3. *Suppose the infection spreads on a grid, and we use the Threshold Ball Algorithm. Suppose that the expected number of reporting nodes scales at least as* $\log n$*.*

(a) Suppose t is known. Set the threshold $m = 1.1 d\mu t$. Then there exists constant C_6 such that, if the expected number of infected nodes is less than $C_6 n$,

$$P(\text{error}) \to 0.$$

(b) Next, suppose time t is unknown. Let X_{rep} be the number of nodes reporting an infection, $|S_{\text{rep}}|$. Use threshold $m = 1.1 d^2 (X_{\text{rep}} \log\log n/q)^{1/d}$. Then provided that for a constant C_7, the expected number of infected nodes is less than $C_7 n/\log\log n$,

$$P(\text{error}) \to 0.$$

Proof outline. This theorem follows using the shape theorem given in Lemma 2.3.1: the epidemic will be contained within a ball with radius scaling linearly with time. A simple counting argument is sufficient to show that the random sickness will be sufficiently spread out. The proof details are given in the appendix. □

Then for the Threshold Ball Algorithm, the infection can be identified when up to a constant fraction of the network is infected. This is clearly order-wise optimal. The reason this ball algorithm works so well is that the shape of the infection can be well approximated by ball. This fact can be shown from use percolation theory on infinite lattices [34]. We find that the Threshold Ball Algorithm heavily outperforms the Threshold Tree Algorithm in this setting.

Theorem 2.3.4. *Consider an infection spreading on a grid. Apply the Threshold Tree Algorithm and assume the expected number of reporting nodes scales at least as* $\log n$*.*

(a) Consider when t *is known. Use threshold* $m = (3\mu t)^d$*. Then there exists constant* C_8 *such that, if the expected number of infected nodes is less than* $C_8 n/(\log\log n/q)^d$*,*

$$P(\text{error}) \to 0.$$

(b) Consider unknown t*. Define* X_{rep} *as the number of nodes reporting an infection, as set the threshold to* $m = X_{\text{rep}} \log\log n/q$*. If there exists constant* C_9 *such that expected number of infected nodes is less than* $C_9 n/(\log\log n/q)^{3d}$*,*

$$P(\text{error}) \to 0.$$

Proof outline. The size of the Steiner tree for the epidemic is clearly no larger than the size of the epidemic, so the Type II error probability clearly goes to 0. We lower bound the size of the Steiner tree containing a random sickness by dividing the grid into blocks, and showing the tree must travel through a large number of these blocks. See the appendix for details. □

2.3.2 Trees

In this section, we analyze the algorithm performance on a tree. A tree is represents a simple type of social network, where there are no cycles that complicate the infection process. Thus, let G be a balanced tree with n nodes, constant branching ratio c, and a single root node a. To reduce edge effects, we force the infection to start at the root of the tree instead of being randomly placed. This makes the infection spread more evenly through the network and not be bottlenecked by the root node (as it would be if the infection started at a leaf node).

First, we provide sufficient conditions for the Threshold Ball Algorithm to succeed with probability tending to 1. As before, the thresholds are first set based on t, and an adaptive threshold is set based on the number of infected nodes. A key fact used here is that on trees with a fixed branching distribution, the infection speed (the graph distance from the root of the farther infected node divided by the time) can be upper bounded with high probability as time scales.

This speed bound follows from results in first passage percolation [7]. In particular, one can compute the fastest-sustainable transit rate. This quantity is basically the time from the root to the leaves, normalized for depth, as the size of the tree scales. Formally (again, see [7] for details), let us consider a limiting process of trees whose size grows to infinity, with Γ_n denoting the balanced tree on n nodes, and $\delta(\Gamma_n)$ denoting the set of paths from the root to the leaves, and for a node $v \in p$ for some path $p \in \delta(\Gamma_n)$, let X_v denote the time it takes the infection to reach node v. Then the *fastest-sustainable transit rate* is defined as: $\lim_n \inf_{p \in \delta(\Gamma_n)} \limsup_{v \in p} \frac{X_v}{\text{depth}(v)}$. Basic results [7] show that this quantity exists, and thus shows that the rate at which an infection travels, defined as the maximum distance of the infection from the root over time, converges to a constant b that depends on the branching ratio. The probability that an infection travels at a faster rate converges (exponentially) to 0 in the size of the tree.

Computing the speed constant may be difficult. One simple method that is applicable to all graphs with maximum degree $\bar{d}$, upper bounds the infection process by an infection on a degree $\bar{d}$ tree. See Section 3.3.3.2 for additional detail regarding this technique. Then we can use a bound in [7] to find that a degree $\bar{d}$ tree satisfies the speed condition with speed $1.1(\bar{d}+1)$. Therefore, the original graph satisfies it with the same speed. Depending on the graph structure, this bound may be weak.

Theorem 2.3.5. *Suppose G is a balanced tree with constant branching ratio and the Threshold Ball Algorithm is used. Additionally, suppose t is sufficiently large that the expected number of reporting nodes is at least $\log n$.*

(a) In the case t is known, there exist constants b, β such that if the expected number of infected nodes is less than n^β, then the algorithm with threshold

$m = 1.1bt$ succeeds:

$$P(\text{error}) \to 0.$$

(b) On the other hand, suppose t is not known. Define X_{rep} as $|S_{\text{rep}}|$. Then there exists constants b_2 and β, such that with the threshold set to $m = 1.1b_2 \log(X_{\text{rep}}(\log\log n)^2/q)$, where if the expected number of infected nodes is less than n^β,

$$P(\text{error}) \to 0.$$

The constant β is identical in both parts (a) and (b).

Proof outline. This result follows using the speed bound for trees. In addition, the random sickness (nearly) always contains a leaf node under the given conditions, and therefore can only be covered by a ball of maximum size. The details of the proof are presented in the appendix. □

That is, there is some exponent $\beta < 1$ such that, as long as the expected number of infected nodes is less than n^β, the Threshold Ball Algorithm works well on a tree. Next, we consider the Threshold Tree Algorithm.

Theorem 2.3.6. *Consider a balanced tree G with constant branching ratio and suppose that the Threshold Tree Algorithm is applied to this problem. Suppose $q = \omega(\log\log n/\log n)$, and t is sufficiently large that the expected number of reporting nodes is at least $\log n$.*

(a) Consider when t is known. Then for any constant $\alpha < 1$, if the expected number of infected nodes scales as less than n^α, with threshold $m = E[|S|] \log\log n$,

$$P(\text{error}) \to 0.$$

(b) Suppose t is not known. Set $X_{\text{rep}} = |S_{\text{rep}}|$, the number of nodes reporting an infection. Use threshold $m = X_{\text{rep}}(\log\log n)^3/q$. Then if for any constant $\alpha < 1$, the expected number of infected nodes is less than n^α,

$$P(\text{error}) \to 0.$$

Proof outline. We again can upper bound the Steiner tree size of the epidemic by the size of the epidemic itself. Lower bounds on the Steiner tree size for the random sickness are obtained by showing the tree must include most of the branches down to a certain depth due to the large number of sick leaf nodes. See the appendix for the complete details of the proof. □

Note that the Ball Algorithm succeeds until the farthest infected node reaches the edge of the graph. At this point, the ball radius can increase no further, thus there is no hope of distinguishing an epidemic from a random sickness. Since this farthest point travels at a faster rate than the bulk of the infection, the Ball Algorithm can only work up to some time $\log_c n/b$. However, the Tree Algorithm can still correctly identify an infection with high probability nearly to the point where $\Theta(n)$ nodes are sick. This includes infection times close to $\log_c n$, the time it takes for almost every node to be infected. From this, we see that the Tree Algorithm works for a wider range of times compared to the Ball Algorithm. This is demonstrated by simulations in Section 2.4.

2.3.3 Erdös-Renyi Graphs:

The final graph we consider are Erdös-Renyi graphs. These represent standard social networks, with both a small diameter and rapid epidemics. Both of these factors make this problem more challenging. Define the graph $G = G(n, p)$ to be the graph with n nodes and for each pair of nodes, there is an edge between them with probability p. In the section above, we used c to denote the branching ratio. We overload notation and use it again to measure the spread of the graph, but here as the (approximate) expected degree: let $p = c/n$ with $c > 1$. In this regime, the graph is almost surely disconnected, but there is a giant component. Since this problem would be trivial on a disconnected graph, we limit both the epidemic and random sick nodes to the giant component. Unlike the case of trees, we are unable to distinguish infection from random sickness for close to a constant fraction of nodes. Instead, we consider infections that cover only $o(n)$ nodes. As is well-known (e.g., [18]) in this connectivity regime, the graph is locally tree-like, and hence tree-like in the infected region. Then locally, the infection behaves very similar to the trees in the last section, and as might be expected, our results are similar.

As before, first we analyze the Threshold Ball Algorithm.

Theorem 2.3.7. *Suppose we use the Threshold Ball Algorithm with $G = G(n, p)$. Consider the case when the expected number of reporting nodes is no less than $\log n$.*

(a) Suppose we have knowledge of t. There are constants b_3, β_2 where, using threshold $m = 1.1b_3t$ and with expected number of infected nodes less than n^{β_2},

$$P(\text{error}) \to 0.$$

(b) Consider unknown t. We set X_{rep} to be the number of nodes reporting an infection, $|S_{\text{rep}}|$. Then there exists constants b_4 and β_2 such that for threshold $m = b_4 \log(X_{\text{rep}}(\log\log n)^2/q)$ and if the expected number of infected nodes is less n^{β_2},

$$P(\text{error}) \to 0.$$

The constant β_2 is the same for both (a) and (b).

Proof outline. The Type II error probability is shown to be low using a similar speed result as in the case for trees. Neighborhood size bounds are used to establish that a random sickness is spread out so the Type I error rate also decays. The details of the proof are presented in the appendix. □

Therefore, the form of the sufficient condition is the same as for trees: for a constant β_2, the Threshold Ball Algorithm will succeed for expected infection size up to n^{β_2}. However, this constant β_2 is not the same as β from the previous section. The condition for the Threshold Tree Algorithm is easier to compare.

Theorem 2.3.8. *Suppose* $G = G(n, p)$. *Also suppose the Threshold Tree Algorithm is applied. Assume that the expected number of reporting nodes is at least* $\log n$ *and* q *is constant.*

(a) Consider the case where t *is known. Let the threshold* $m = E[|S|] \log \log n$. *For any* $\alpha < 1/2$, *if the expected number of infected nodes scales as less than* n^{α},

$$P(\text{error}) \to 0.$$

(b) Suppose we have unknown t. *Define* X_{rep} *as* $|S_{\text{rep}}|$. *In this case, set the threshold to be* $m = X_{\text{rep}}(\log \log n)^3/q$. *Then like before, for any constant* $\alpha < 1/2$, *if the expected number of infected nodes is less than* n^{α},

$$P(\text{error}) \to 0.$$

Proof outline. Again, we use the size of the epidemic to bound the size of the Steiner tree containing the reporting nodes in that case. Bounding the size of the Steiner tree of the random sickness is much harder. We examine the value equal to the sum over all reporting nodes of the distance from that node to the nearest reporting node. It can be shown that the Steiner tree size is at least half this value. Using appropriate bounds on neighborhood sizes, we lower bound this quantity. The proof details can be found in the appendix. □

Then, the Threshold Tree Algorithm algorithm works for exponents up to 1/2, as opposed to 1 for a tree. For the Erdös-Renyi graph, the sufficient condition for the ball and tree algorithms are not directly comparable. Our simulations results are similar. The Threshold Tree Algorithm has a lower error probability at smaller infections sizes, but the Threshold Ball Algorithm works better for larger infections, when the problem is more challenging. Overall, the Threshold Ball Algorithm seems superior for this graph topology.

2.4 Simulations

In this section we provide simulation-based evidence of the theoretical results of the previous sections. The simulations aim to demonstrate, in particular, two facts. First, the thresholds specified in the previous sections do actually work empirically: as the graph size increases, the probability of both types of error decrease to zero. In addition, this provides insight into how quickly the probability of error decays. While our results include rate estimates given as part of the proof of correctness, we have not made an effort to optimize these in this work. Second, we seek to describe the relative performance of each algorithm, and show that it is as described above. Thus, we show that the Ball Algorithm outperforms the Tree Algorithm on a grid; the Tree Algorithm performs better than the Ball Algorithm on a balanced tree (for larger infections); and on an Erdös-Renyi graph, the performances are similar, with the Ball Algorithm performing slightly better. We accomplish this by determining the probability of error for a range of infection times. We call an algorithm superior if it works in a wider range of times.

2.4.1 Methodology

We executed both of our algorithms under a variety of conditions to estimate the probability of error. In order to use the Threshold Tree Algorithm in a reasonable time frame, it was necessary to use an approximate Steiner tree algorithm. Naturally, since the exact problem is NP-hard, this would be required in any practical use of this algorithm at the moment. However, as a consequence, the empirical results may differ from the true theoretical result that would be obtained by employing an exact algorithm. Nevertheless, approximation algorithms typically have reasonable performance and we do not expect significant deviation from the correct results. The approximation algorithm we use is the Mehlhorn 2-approximation algorithm provided by the Goblin library [44]. This algorithm is an efficient algorithm which produces a Steiner tree with no more than twice the optimal number of edges.

Each of the points in these results represents the average of 10000 runs. The average infection size, which is used to normalize the expected infection size in a random sickness, was determined by averaging the results of 10000 infections. For each simulation, we use a reporting probability $q = 0.25$ (unless otherwise specified), and other parameters (n, t and m) as specified in each section below. Finally, the graphs are plotted with error bars at one standard deviation.

2.4.2 Error Rate Versus Graph Size

Though our theoretical results have characterized the range for which each algorithm works, naturally we wish to see empirically the error probability for each algorithm and the rate at which the error decreases as graph size increases. Both Type I and Type II error probabilities were determined for each algorithm and graph topology. For this section, we have chosen time to keep the fraction of infected nodes at a consistent scaling. In particular, $t = 0.2\sqrt{n}$ for the grid, and $t = 0.5 \log(0.5n)$ with $p = 2/n$ for the Erdös-Renyi graph. The exact constants for these scalings were chosen empirically so that the probability of error was low and the Type I and Type II errors were as balanced as possible. The thresholds m were also chosen with the same scaling, according to our theoretical results. To be exact, for the grid, the Threshold Ball Algorithm used threshold $m = 0.75\sqrt{n}$ and the Threshold Tree Algorithm used threshold $m = 0.28n$. For the Erdös-Renyi graphs, the Threshold Ball Algorithm used threshold $m = 0.69 \log(4.33n)$ and the Threshold Tree Algorithm used threshold $m = 0.03\sqrt{n \log n} \log n$.

Figure 2.2 presents our results for grid graphs. The error probability of the Threshold Ball Algorithm on a grid is very low, while the tree algorithm performs relatively poorly. This is expected since the Threshold Ball Algorithm is closely aligned with the shape of an epidemic on this graph. The Threshold Tree Algorithm has a much higher error probability which decays slowly with n, in particular the Type II error.

Next, the results for Erdös-Renyi graphs are in Figure 2.3. Here we see again that the Threshold Ball Algorithm performs better than the Threshold Tree Algorithm, at least for larger n, and that the error probability also seems to be decreasing faster for the Threshold Ball Algorithm as well. Though a tree more closely matches the infection shape on an Erdös-Renyi graph, it is also easier for a random sickness to mimic a small tree, especially for small world graphs like Erdös-Renyi graphs. This causes the Threshold Ball Algorithm to be ultimately superior. The Threshold Tree Algorithm is superior for larger infection sizes on bottle necked graphs (such as trees) where the random sickness can be easily distinguished, as we see in Section 2.4.3.

2.4.3 Error Rate Versus Infection Size

Next, we examine empirically how the infection duration affects the probability of error for each of our algorithms. As discussed above, we compare the two algorithms by the range of infection sizes for which they work, and accordingly, we call an algorithm superior if it maintains a lower probability of error for a larger infection size (fraction of total infected nodes). We use thresholds that minimize the empirical overall probability of error. That is, the sickness was chosen to be either an infection or simply random with equal probability, and the threshold with minimum probability of error from the simulations was chosen.

These results are presented in Figure 2.4 for grids, trees, and Erdös-Renyi graphs. For each of the graph topologies, we used a graph size of $n = 1600$. The error probability is plotted against the average infection size

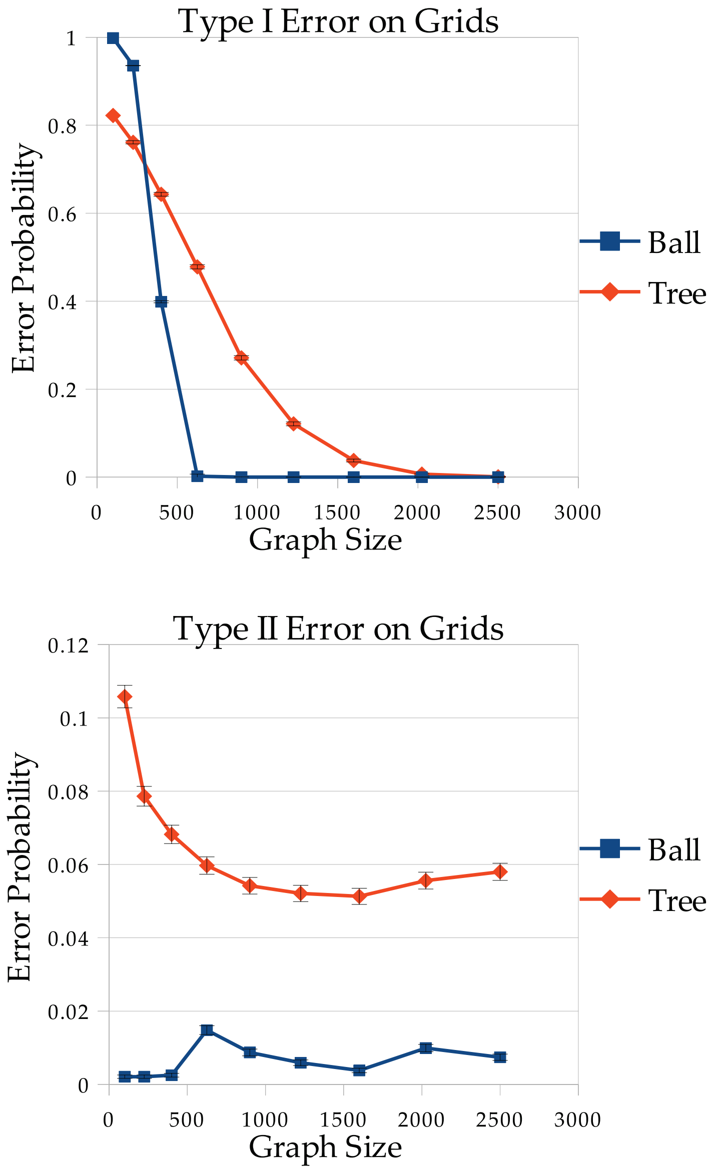
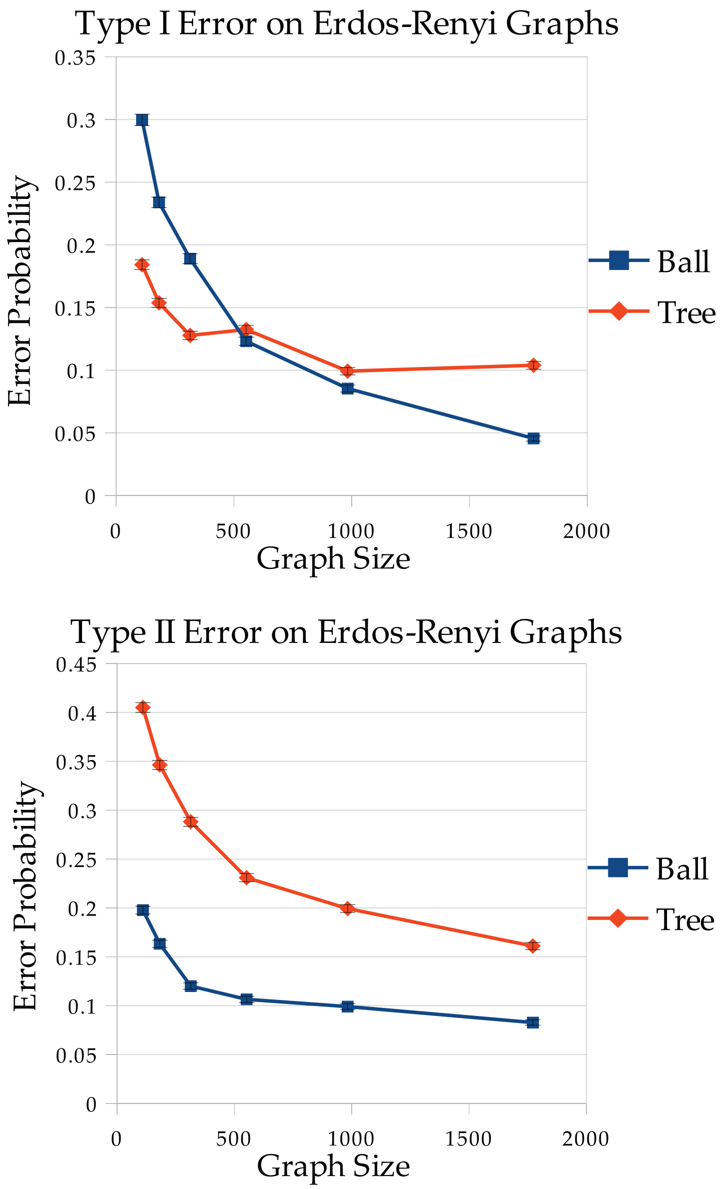


Figure 2.2: Empirical Type I and Type II error probability vs graph size for grid graphs. The sample size is 10000 and infection size scales linearly with n.


Type I Error on Erdos-Renyi Graphs

Type II Error on Erdos-Renyi Graphs

Figure 2.3: Empirical Type I and Type II error probability vs graph size for graphs $G(n, 2/n)$. The sample size is 10000 and infection size scales order-wise as $\sqrt{n}$.

from the simulation. This choice better conveys how infection size affects the error rate, which is the chief question of interest.

These charts allow us to compare the performance of the algorithms. It is clear that the error probability of the Threshold Ball Algorithm is less than that of the Threshold Tree Algorithm on both the grid and Erdös-Renyi graphs. On these graphs, the Threshold Ball Algorithm performs uniformly better across variations in fraction of nodes infected. However, the results on a tree are more complex. When the total infection is small, the Threshold Ball Algorithm has superior performance. However, as a larger fraction of the network becomes infected, the Threshold Tree Algorithm has better performance. We believe it is this right tail that is most significant. In the regime where many of the nodes are infected, the infection is likely to have reached some of the leaves by this time, thus explaining the superiority of the Threshold Tree Algorithm in this regime.

However, many practical applications of these algorithms would occur when the infection is still of limited size, in which case the Threshold Ball Algorithm would perform better. The best algorithm would depend on the circumstances.

It is particularly interesting to ask how these results extend to real-world graphs, as opposed to random (or highly regular) graphs that we have constructed. To this end, we used the call-graph from an Asian telecom network. In this graph, each node is a cell customer, and there is an edge between two users if they contacted each other over this network during a certain range

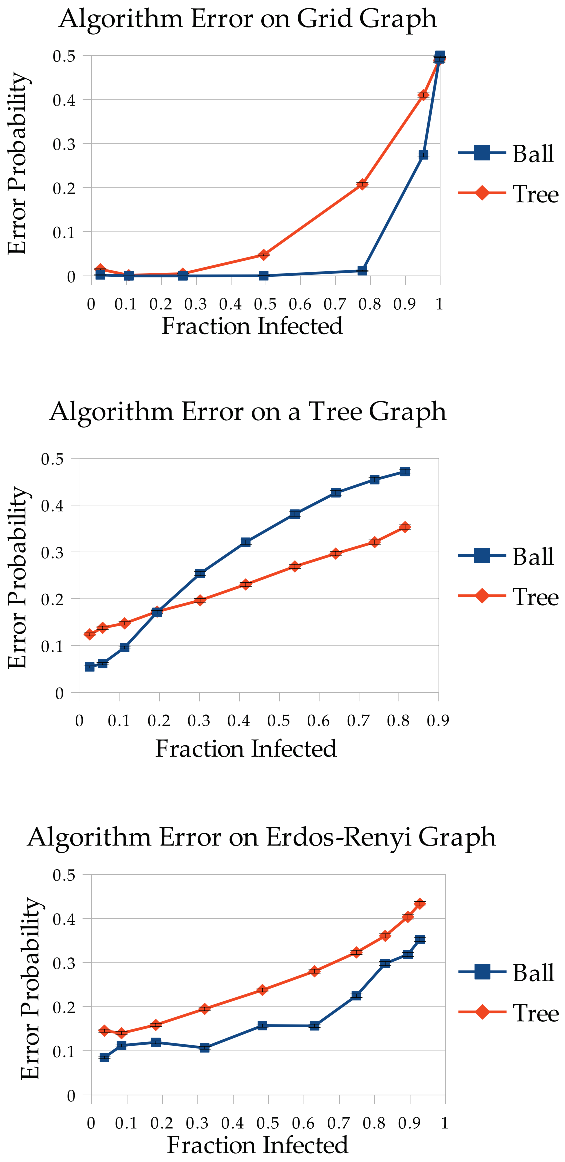


Figure 2.4: This figure shows the overall error probability for each algorithm, for each of the three topologies we consider, over a range of infection sizes.

of time. Since the original graph was too large for practical simulation times, we cut out a partial subset. We chose a random node and all nodes with a distance 9 and used the induced subgraph generated by these nodes. The resulting graph has size $n = 13189$. The probability of error for a range infection sizes are presented in Figure 2.5. We see that the results are similar to those for a Tree graph, where the Threshold Ball Algorithm performs better on small infections, but it is out performed by the Threshold Tree Algorithm in larger infections. This is to be expected, as the intuition for the Threshold Ball Algorithm stems from the geometry of spatial grid-like networks. The call-graph here is very much tree-like (with very small diameter and high degree), and infections are unlikely to propagate to the same depth across various leaves. This results in poor ball "fits," especially as the infected fraction of nodes grows. This intuition is indeed borne out in the simulations.

2.4.4 Error Rate Versus Reporting Probability

The final simulation focused on determining how varying the reporting probability affects the probability of error. Our theoretic results do not provide any intuition on the how the error probability will change as the reporting probability increases, and simply require a minimum reporting probability (sufficiently large so that at least $\log n$ nodes report) for good algorithm performance. To provide this otherwise absent information, we simulated the Threshold Ball Algorithm on a grid graph with 1600 nodes. We used epidemic durations of $t = 10$ and $t = 11$, close to the threshold where the probability of error for the algorithm begins to increase rapidly. The threshold m was set to the optimum value as determined empirically. The average probability of error, with epidemic and random sickness equally likely, are shown in Figure 2.6.

The figure shows that at very low reporting probabilities, the error probability is high. However, the probability of error decreases rapidly as q increases. Once q reaches a value where approximately 40% of infected noded report their infection, the error probability is near a minimum and increased knowledge of the reporting nodes does not substantially improve the algorithm's performance. Note that there is a slight jump in the error probability around $q = 0.6$ which is caused by the fact that the threshold must be an integer, and this jump represents when the threshold increases by one.

2.5 Conclusion

We develop the Threshold Ball Algorithm and the Threshold Tree Algorithm, and show that these algorithms can distinguish between a random sickness and an epidemic on a variety of graph topologies. A summary of the maximum infection size for which our algorithms succeed from our sufficient conditions is shown in Table 2.1 (where reporting probability is constant). From our analytic and empirical results, we conclude that the ball based algorithm is superior. It is more efficient and has a lower probability of error for most of our tests. In later chapters of this thesis, we focus on the ball

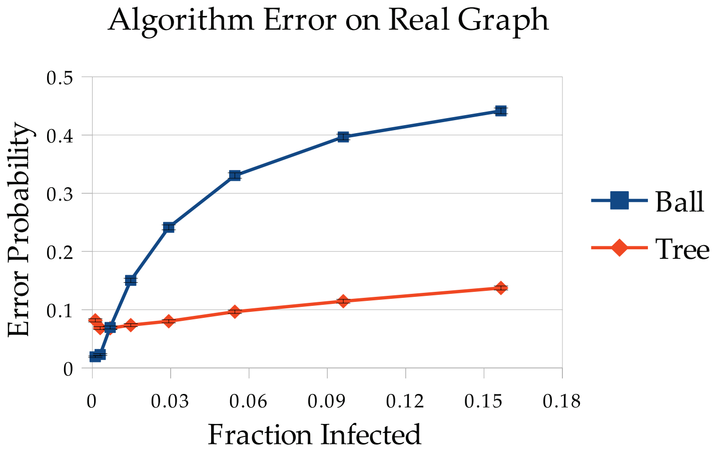


Figure 2.5: This figure shows the overall error probability for each algorithm on a real world graph.

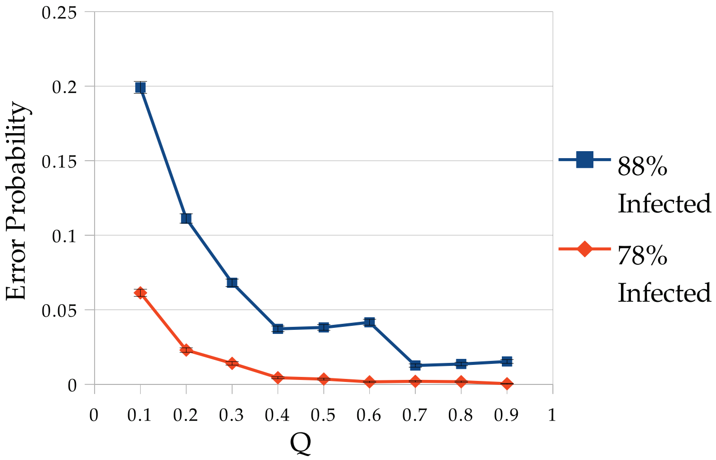


Figure 2.6: The error probability of the Threshold Ball Algorithm on a grid graph (n = 1600) for a large range of reporting probabilities, with a sample size of 10000.

algorithm and variations of it.

Table 2.1: Summary of maximum proven distinguishable infection sizes.

Graph	**Ball Algorithm**	**Tree Algorithm**
Grid	$\Theta(n)$	$\Theta(n/(\log \log n)^d)$
Tree	$\Theta(n^\beta)$	$\Theta(n^{1-\epsilon})$
Erdös-Renyi	$\Theta(n^{\beta_2})$	$\Theta(n^{1/2-\epsilon})$

Chapter 3

Distinguishing Two Infections

3.1 Introduction

People and devices routinely interact through multiple networks – contact networks – be they virtual, technological or physical, allowing the rapid exchange of ideas, fashions, rumors, but also viruses and disease. Throughout this paper we refer to anything that spreads over a contact network as an *epidemic*. In many domains, it is of critical importance to understand the *causative network* of that epidemic. Economists, sociologists and marketing departments alike have long sought to understand how ideas, memes, fads and fashions, spread through social networks. Meanwhile, epidemiology has understood the value of knowing the causative network of disease epidemics, from Influenza to HIV. Indeed, at one point, HIV was known as the "4H disease" where 4H referred to "Haitians, Homosexuals, Hemophiliacs, and Heroin users" [62, 12]. Understanding the causative network has greatly contributed to controlling the worldwide spread of the virus.

The work in this chapter appears in the following publication:
Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. On identifying the causative network of an epidemic. In *Proceedings of 50th Annual Allerton Conference on Communication, Control, and Computing*, October 2012.

While smartphone viruses have not yet supplanted computer viruses as the spreading technological threat of the hour, their potential for broad destructive impact is clear. Just as different human viruses may have different dominant spreading networks (again, compare Influenza and HIV), so may smartphone viruses spread over multiple networks, including bluetooth, SMS/MMS messaging, or e-mail. Yet the symptoms of these viruses may be deceptive, appearing to be simple hardware failure, and may disguise the true infection mechanism.

A first step towards containing epidemics, be they technological or physical, relies on properly understanding the phenomenon as an epidemic in the first place, and then, accurately understanding the causative spread, before then adopting network-specific strategies for containment, quarantining and treatment.

Many factors complicate the process of determining the causative network. First, possibly because of long latency/hibernation periods, variation in reporting/detection, or simply lack of data, in some cases it may be difficult or impossible to collect accurate longitudinal data. Equally importantly, the reporting set of those infected (be they people or devices) may be only a tiny fraction of those in fact infected. We consider the most dire information regime: we assume we have data from only a single snapshot of time, where only a (perhaps vanishing) fraction of the infected population reports.

With these data, this paper focuses on determining the causative network for the spread of an epidemic (e.g., virus, sickness, or opinion) from limited samples of the network state. We do this in the setting where we are given two possible graphs over which epidemic may spread. Provided the networks are sufficiently distinct, we use the topological differences of the infection on each graph to determine which network represents the true infection process.

3.2 Problem Statement

In this section, we detail the infection model and required graph properties. We specify our proposed algorithm, the Comparative Ball Algorithm, which we analyze throughout the rest of this chapter.

3.2.1 Infection Model

We assume that an epidemic is propagating on one of the two graphs, G_1 or G_2. The objective is to determine on which network it is spreading. We reiterate that this 'epidemic' could model many situations, including the spread of a cellphone virus, physical sickness of humans, and opinions or influence about products or ideas.

Given that the epidemic is on graph G_i, the spread occurs as follows (the standard SI dynamics [20]). A node is randomly selected to be the epidemic seed, and thus is the first infected node. At random times, the illness spreads from the sick nodes to some subset of the neighbors of the sick nodes, according to an exponential process. Specifically, associate an independent mean 1 exponential random variable with each edge incident to an infected and an uninfected (a susceptible) node. The realization of this random variable represents the transit time of the infection across that specific edge. Thus an infected node proceeds to infect its neighbors, with each non-infected neighbor becoming infected after the random transit time associated with the edge between the infected node and this neighbor. This process proceeds until eventually the entire graph G_i is infected.

In either case, the infection continues until some time $t^{(n)}$. At this time, a sub-sample of the infected nodes report their infection state independently, each with some probability $q^{(n)} < 1$. Both $t^{(n)}$ and $q^{(n)}$ may depend on the total number of nodes n. We let $S^{(n)}$ denote the set of infected nodes, and let $S^{(n)}_{\text{rep}} \subseteq S^{(n)}$ be the set of reporting infected nodes. Note that $S^{(n)}$ is a function of $t^{(n)}$ and $S^{(n)}_{\text{rep}}$ is a function of both $t^{(n)}$ and $q^{(n)}$. On the causative network of the infection, $S^{(n)}$ will be a clustered, connected set of nodes. Unless required for clarity, we suppress the dependence on n and write t, q, S and S_{rep} for the infection time, reporting probability, set of infected nodes, and set of reporting nodes respectively.

3.2.2 Graph Independence

For the statistical problem of distinguishing the causative network to be well-posed, the contact networks encoded by graphs G_1 and G_2 must be sufficiently different. Note that this does not imply that the topology of the graphs must be different (indeed, it could be identical). Rather, the neighborhoods of each graph must be distinct, i.e., the nodes that are near an infected node with respect to one graph, must be different from the nodes near the same infected node, with respect to the other graph. We note that if this is not the case, then both graphs encode approximately the same causative network, and hence solving the comparative graph problem is not that important.

In this paper, we encode this idea of graphs having sufficiently different neighborhoods via a probabilistic construction that guarantees that corresponding nodes on the two graphs have *independent neighborhoods*. This essentially means that given a node, v, its neighborhood in G_1 and its neighborhood in G_2 are *independent*. We suppose that both graphs G_1 and G_2 come from graph families $\mathcal{G}_1$ and $\mathcal{G}_2$ as defined in Section 2.2.4. For each pair of these graphs, we require them to have independent neighborhoods as defined by the following construction.

Definition 3.2.1. Graphs G_1 and G_2 have *independent neighborhoods* if their nodes are labeled as follows. Let V be the set of nodes in the population under consideration. These nodes are mapped to the nodes in G_1 and G_2 (V_1 and V_2) by uniformly random labeling functions. That is, let $\text{label}_1 : V_1 \mapsto V$ be a one-to-one function where the mapping is chosen uniformly at random. Let label_2 be likewise defined for V_2, and independently from label_1. Two nodes are identified if they receive the same label (that is, map to the same vertex in the population V), and hence are both infected or both well. Hence we can talk about a single set of common nodes, and then edges that come from G_1, and edges that come from G_2.

For a set of nodes I, define $L_1(I) = \bigcup_{i \in I}\{\text{label}_1(i)\}$ and similarly for L_2. Then when G_1 and G_2 have *independent neighborhoods* as defined above, for any pair of sets of nodes $I_1 \subset V_1$ and $I_2 \subset V_2$, $L_1(I_1)$ and $L_2(I_2)$ are independent. In particular, a set of clustered nodes on one graph may correspond to any possible set of nodes on the other graph, each equally likely.

This independent neighborhood condition is simply one way to make precise, and encode into a probabilistic framework, the natural condition that two graphs have neighborhoods that are "unrelated." For a practical example, consider the bluetooth contact graph during a commuter's subway transit to work in a busy city, compared to the e-mail contact graph. The majority of people on the subway are typically strangers and hence do not exchange e-mails; meanwhile the majority of co-workers and friends have different morning commutes, and hence are not in bluetooth range during the morning commute. That is, nodes (in this case, people) that are connected or nearby on one graph (the proximity graph) may be spread out on the other graph (the e-mail contact graph). The distances between pairs of nodes on each graph are approximately independent.

On the causative network, the epidemic will consist of a connected, clustered set of infected nodes. However, due to the above condition, the infection will appear to be a completely random sickness on the other graph. That is, the infection will only be clustered on the network over which the infection spread. This fact can be exploited to determine the correct network. Figure 3.1 shows an example of two graphs that have indepedent neighborhoods.

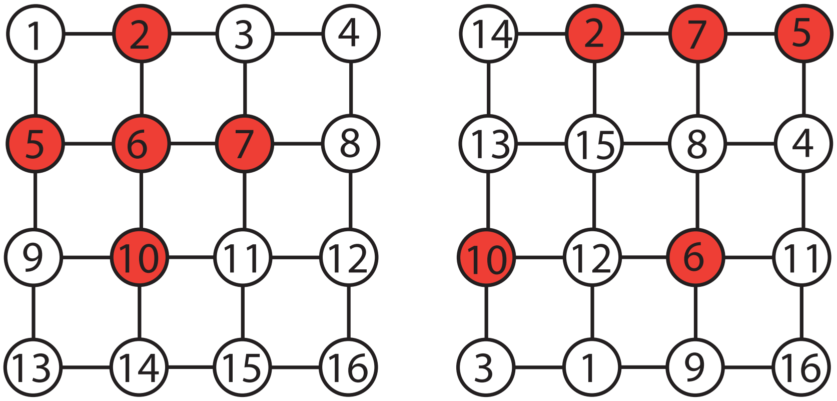


Figure 3.1: This figure shows two different graphs with the random labels from the independent neighborhood condition shown. Infected nodes are colored red. Note that nodes with the same label have the same status (e.g. both are infected).

Note that the infection on the right-hand graph is unclustered.

3.2.3 Comparative Ball Algorithm

We provide an algorithm for this problem called the *Comparative Ball Algorithm*. The algorithm is natural, given the discussion above. We find the smallest ball on that graph that contains all the reporting infected nodes. We take the ratio of the radius of this ball to that of the graph's diameter. These ratios – called the *score* of each graph – serve as a topology independent measure of clustering on each graph. The Comparative Ball Algorithm returns the graph with the smallest normalized clustering ratio. This is formally described below.

For the below algorithm, define Ball(G, S) as (possibly one of) the ball containing all the nodes in S with the minimum radius, and denote the radius of this ball as RadiusBall(G, S). As we have done above, we denote the diameter of the graph by diam(G).

Algorithm 4 Comparative Ball Algorithm

Input: Two graphs, G_1 and G_2; Set of reporting infected nodes S_{rep};
Output: G_1 or G_2

```
$a_1 \leftarrow \text{RadiusBall}(G_1, S_{\text{rep}})$
$b_1 \leftarrow \text{diam}(G_1)$
$x_1 \leftarrow a_1/b_1$
$a_2 \leftarrow \text{RadiusBall}(G_2, S_{\text{rep}})$
$b_2 \leftarrow \text{diam}(G_2)$
$x_2 \leftarrow a_2/b_2$
if $x_1 \leq x_2$ then
  return $G_1$
else
  return $G_2$
end if
```

3.3 Main Results

We analyze the performance of the Comparative Ball Algorithm for a wide variety of graph topologies. To do this, we impose two fairly mild conditions on the graphs, termed the *speed condition* and the *spread condition.* If both graphs satisfy these conditions, then we prove that the Comparative Ball Algorithm correctly identifies the causative network with probability of error tending to zero asymptotically for a wide range of infection size. In fact, the algorithm is order-wise optimal in the maximum infection size for which is succeeds. We then show that two standard graph topologies, grids and Erdös-Renyi graphs, satisfy the required conditions.

3.3.1 Graph Conditions

Distinguishing between two graphs is only meaningful when neither graph has a trivial neighborhood structure. For instance, if one graph is the complete graph, there is no topological information conveyed by knowing which nodes are sick on that graph, and the problem is roughly equivalent to the random sickness vs. infection case. The first order of business is understanding precisely what conditions we require the topology of graphs G_1 and G_2 to satisfy, making precise the notion of "non-trivial neighborhood structure" where, unlike for example the star graph, an epidemic exhibits some statistically detectable clustering. There are two key properties required: first, the infection must spread at a bounded speed; second, a random collection of nodes on the graph must, with high probability, not exhibit a strong clustering. Of course, the star graph fails with respect to the minimum spread of random nodes condition. As another example that fails the bounded speed condition, consider a tree whose nodes have degree d^{k+1} at level k.

We now state these conditions precisely, and in addition, we show, many graphs satisfy these conditions, including familiar topologies like the d-dimensional grid and the Erdös-Renyi graphs. It is also easy to see that any graph with bounded degree also satisfies these two conditions.

We first restate the following definition:

Definition 3.3.1. Given a graph $G = (V, E)$ and a subset of its nodes, $S \subseteq V$, let RadiusBall(G, S) denote the radius of the smallest ball that contains S.

Note that for any set S, $\text{RadiusBall}(G, S)$ can be easily computed in time $O(|V|^2 \cdot |S|)$.

Let $\mathcal{G} = \{\mathcal{G}^{(n)}\}$ denote a family of graphs, where $\mathcal{G}^{(n)}$ denotes the subset of the graphs of $\mathcal{G}$ that have n nodes. For each n, there is a (possibly trivial) probability space $\left(\mathcal{G}^{(n)}, \sigma(\mathcal{G}^{(n)}), P^{(n)}\right)$ from which graphs are drawn. Concrete examples include the set of d-dimensional grid graphs, Erdös-Renyi graphs with bounded expected degree, d-regular trees, etc.

Definition 3.3.2. A family $\mathcal{G}$ satisfies the *speed* and *spread* conditions, if there exist constants $s_{\mathcal{G}}$, $b_{\mathcal{G}}$ and $\beta_{\mathcal{G}}$, such that for sequences $\{G^{(n)}\}$ picked randomly from the product probability space $\prod_n \mathcal{G}^{(n)}$, the following hold with probability approaching 1 as n increases, where the probability is over the random subset of nodes in the definitions below, and, in the case of random families, $\mathcal{G}$, such as Erdös-Renyi graphs, over the selection of $G^{(n)}$ as well:

> **Speed Condition**: For infections starting at a randomly selected node, and for infection times $t^{(n)} \to \infty$, the set $S^{(n)}$ of nodes infected at time $t^{(n)}$ satisfies $\text{RadiusBall}(G^{(n)}, S^{(n)}) < s_{\mathcal{G}} t^{(n)}$ with probability tending to 1 as n increases.
>
> **Spread Condition**: First, $\text{diam}(G^{(n)}) = \Omega(\log n)$. Define $S^{(n)}$ as a set of nodes chosen uniformly at random from all nodes in $G^{(n)}$ (as in a random sickness), with $\left|S^{(n)}\right| > \beta_{\mathcal{G}} \log n$. Given such a set, we require that $\text{RadiusBall}(G^{(n)}, S^{(n)}) > b_{\mathcal{G}} \text{diam}(G^{(n)})$ with probability approaching 1 as n increases.

These two conditions essentially encode the properties required so that an infection spreading on a graph $G_1^{(n)}$ (chosen from family $\mathcal{G}_1$) exhibits clustering, and, conversely, if it is spreading on another graph $G_2^{(n)}$ (chosen from family $\mathcal{G}_2$) with independent neighborhoods (as described above) then there is no clustering with respect to $G_1^{(n)}$.

Note that to ease notation, whenever the context is clear, we drop the superscript (n) that denotes the number of nodes.

If a graph G satisfies both of these conditions, we say that the graph is 'detectable'. An infection on a detectable graph is sufficiently well behaved that it is possible is detect whether it is likely that an infection spread on that graph.

3.3.2 Main Theorem

Using the algorithm definition, we prove sufficient conditions for the probability of error of the Comparative Ball Algorithm decaying to 0.

Theorem 3.3.1. *Consider families of graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ satisfying the speed and spread conditions above and with independent neighborhoods, and let the sequence $\{(G_1^{(n)}, G_2^{(n)})\}$ denote a sequence of graphs drawn from $\mathcal{G}_1$ and $\mathcal{G}_2$. Consider infection times $t^{(n)}$ such that the number of reporting infected nodes scales at least as $\max(\beta_{\mathcal{G}_1}, \beta_{\mathcal{G}_2}) \log n$. Then when the infection spreads over G_1, if $t < b_{\mathcal{G}_2} \mathrm{diam}(G_1)/s_{\mathcal{G}_1}$, the Comparative Ball Algorithm correctly determines G_1 is the causative network with probability approaching 1. Similarly, for an* *infection on* G_2*, if* $t < b_{\mathcal{G}_1}\text{diam}(G_2)/s_{\mathcal{G}_2}$*, then the Comparative Ball Algorithm correctly identifies the infection with probability approaching* 1*.*

Proof. By symmetry, it is sufficient to prove that an infection spreading on G_1 is indeed detected as such. Suppose then, that G_1 is the causative network. For every n, let S_{rep} (again we suppress dependence on n when it is clear from the context) denote the set of reporting sick nodes, where $|S_{\text{rep}}| > \beta_{\mathcal{G}_2} \log n$. Though S_{rep} will be clustered on G_1 since it is the causative network, by the independent neighborhood assumption, this set of nodes is randomly distributed over G_2. By the speed and spread conditions, with probability approaching 1 as n scales, $\text{RadiusBall}(G_1, S_{\text{rep}}) < s_{\mathcal{G}_1}t$ and $\text{RadiusBall}(G_2, S_{\text{rep}}) > b_{\mathcal{G}_2}\text{diam}(G_2)$. Then the score for the first graph satisfies $\text{score}(G_1) < s_{\mathcal{G}_1}t/\text{diam}(G_1) < b_{\mathcal{G}_2}$ by hypothesis. Similarly, $\text{score}(G_2) > b_{\mathcal{G}_2}\text{diam}(G_2)/\text{diam}(G_2) = b_{\mathcal{G}_2}$. Therefore, the algorithm correctly identifies an infection. □

Note in particular that $s_{\mathcal{G}}$ and $b_{\mathcal{G}}$ are constants for both graph families. Therefore, the algorithm can distinguish infections for infection times order-wise the same as the diameter of the graph. Since the infection spreads at a constant rate 1, the diameter of the graph is also order-wise the same time as it would take to infect the entire network. Naturally, it would be impossible to distinguish infections at the point when infection has spread over the whole network. Hence, Theorem 3.3.1 guarantees that the algorithm distinguishes infections for infection times that are order-wise optimal.

3.3.3 Detectable Graphs

Though we show that our Comparative Ball Algorithm performs well on detectable graphs, it is as yet unclear what graphs are detectable, and hence how meaningful our result is. In fact, our speed and spread conditions are fairly mild and are satisfied by many typical graph topologies. To illustrate this fact, we prove that grids and Erdös-Renyi graphs satisfy these conditions using the similar ideas as in Chapter 2.

Theorem 3.3.2. *Both d-dimensional grids, and the giant component of Erdös-Renyi graphs with constant average degree, are detectable.*

Proof. This result follows immediately from Lemmas 3.3.3, 3.3.4, 3.3.5, and 3.3.6 presented below. □

3.3.3.1 Grids

First, we consider d dimension grids of size n. A grid consist of a lattice of nodes with side length $n^{1/d}$. In order to avoid edge effects, we connect each node on the edge to its corresponding node on the other side, forming a torus. This also means the initial infected node does not effect the way the epidemic spreads. Grids serve as a useful model of geographic social networks, where nodes is close physical proximity are connected. These are characterized by a large number of small cycles and a relatively large diameter.

Lemma 3.3.3. *Let $G^{(n)} = \mathrm{Grid}(n, d)$ and let $t^{(n)}$ denote any sequence of increasing times, $t^{(n)} \to \infty$. As defined above, $S_{\mathrm{rep}}^{(n)}$ denotes the (random)* *subset of nodes infected by the epidemic, that report their infected status. Then there exists a constant μ such that*

$$\text{RadiusBall}(G^{(n)}, S^{(n)}_{\text{rep}}) < 1.1d\mu t^{(n)},$$

with probability converging to 1 *as $n \to \infty$.*

Proof. We drop the indexing w.r.t. n, since the context is clear. Let $\mu \triangleq \sup_x\{(x, 0, ..., 0) \in B_0\}$ from Lemma 2.3.1 and $m = 1.1d\mu t$. Then we must show $\text{RadiusBall}(G, S_{\text{rep}}) < m$ with probability approaching 1. Note that if the infection can be limited to the subgrid $[-m/d, m/d]^d$ (with appropriate translations), then this condition is satisfied. Define E as the event that $\text{RadiusBall}(G, S_{\text{rep}}) \geq m$. Therefore, using Lemma 2.3.1,

$$\begin{aligned} P(E) &< 1 - P\{B(t) \subset [-m/d, m/d]^d\} \\ &< C_1 t^{2d} e^{-C_2 t^{-1/2}(m/(d\mu) - t)} \qquad (3.1) \\ &= C_1 t^{2d} e^{-0.1 C_2 t^{1/2}} \\ &\to 0. \end{aligned}$$

Equation 3.1 follows from Lemma 2.3.1 with $x = t^{-1/2}(m/(d\mu) - t)$, using $[-m/d, m/d]^d \supset m/(d\mu)B_0 = (t + t^{1/2}x)B_0$. Hence, $\text{RadiusBall}(G, S_{\text{rep}})$ satisfies the required bound with high probability. □

Lemma 3.3.4. *Let $G^{(n)} = \text{Grid}(n, d)$. Let $S^{(n)}$ be a collection of nodes chosen uniformly at random from $G^{(n)}$, such that $\left|S^{(n)}\right| > \log n$ for sufficiently high*

n. Then

$$\text{RadiusBall}(G^{(n)}, S^{(n)}) > n^{1/d}/4,$$

with probability converging to 1 *as* $n \to \infty$.

Proof. Again we drop the n-index wherever context makes it clear. By assumption, we have a set S of random nodes with $|S| > \log n$. Define $X = |S|$. We show the probability all nodes in S are within some ball of radius $n^{1/d}/4$ decays to 0 with n. There are at most n of these balls, since each node is in correspondence with the ball centered on itself (though two different centers may result in the same ball). Then consider one of these balls. There are less than $l = (n^{1/d}/2)^d$ nodes in that region (the number of nodes in a 'box' of side $n^{1/d}/2$). Within this ball, there are at most $\binom{l}{X}$ arrangements of the sick nodes out of $\binom{n}{X}$ total possible arrangements. Therefore, the probability all the sick nodes are within the region is no more than

$$\begin{aligned}\binom{l}{X}\Big/\binom{n}{X} &= \frac{l!(n-X)!}{(l-X)!n!} \\ &\leq (l/n)^X.\end{aligned}$$

Using a union bound over the n balls, we find that the probability there is a ball of that size containing all nodes in S is at most $n(l/n)^X$. Then

$$\begin{aligned}n(l/n)^X &< n\left(\frac{1}{2^d}\right)^{\log n} \\ &= n^{1-d\log 2} \\ &\to 0.\end{aligned}$$

Therefore, $\text{RadiusBall}(G, S) > n^{1/d}/4$ with probability converging to 1. □

Since the diameter of a grid is (nearly) $d/2n^{1/d}$, we see that a grid satisfies both the speed condition (Lemma 3.3.3) and the spread condition (Lemma 3.3.4), and hence grids are detectable.

3.3.3.2 Erdös-Renyi graphs

Now we consider Erdös-Renyi graphs, representing infections that spread over low diameter networks (the diameter grows logarithmically with network size). An Erdös-Renyi graph is a random graph with n nodes, where there is an edge between any pair of nodes, independently with probability p. These graphs are denoted $G(n, p)$. We study the Erdös-Renyi graph in the regime where $p = c/n$, for some positive constant $c > 1$. This setting leads to a disconnected graph; however, there exists a giant connected component with $\Theta(n)$ nodes with high probability in the large n regime. In this paper, we restrict our attention to epidemics on this giant component. Thus we limit both the infection and the random set of reporting nodes (due to the labeling when the infection occurs on the alternative graph) to occur exclusively on the giant connected component. If the infection on the other graph contains too many nodes for the giant component, we simply ignore the excess, but this point is already outside the regime of interest.

In order to establish that the Erdös-Renyi graph is detectable, we show first that on these graphs, an infection spreads at a bounded speed, and second, that randomly selected nodes are spread out. In fact, the two results given in this section also hold for bounded-degree graphs. The key properties used in the proofs are a speed upper bound for trees from [7] and that the number of nodes within distance m from a given node is $O(m^3 c^m \log n)$. Both of these are true (and even simpler) for bounded-degree graphs. The remainder of the proofs immediately carries over to this class. For simplicity, and because the randomness of the Erdös-Renyi graphs presents some further complications, we state everything in terms of the Erdös-Renyi graphs.

Lemma 3.3.5. *Let $G^{(n)}$ denote the connected component of a realization of a $G(n,p)$ graph, and let the sequence $t^{(n)}$ denote increasing time instances, scaling (without bound) with n. As above, let $S_{\text{rep}}^{(n)}$ denote the random subset of nodes reached by the epidemic, that also report. Then there exists a constant C_6 such that*

$$\text{RadiusBall}(G^{(n)}, S_{\text{rep}}) < C_6 t^{(n)},$$

with probability converging to 1 as $n \to \infty$.

Proof. Since the dependence on n is clear, we drop the index of n. This theorem essentially states that there is a maximum speed at which the infection can travel on an Erdös-Renyi graph. The statement follows from a similar maximum speed result for trees [7]. Therefore, it remains to show how this result can be applied to an Erdös-Renyi graph. To do this, we upper bound an infection on an Erdös-Renyi graph by a tree that represents the routes on which an infection can travel. Since an Erdös-Renyi graph is locally tree-like [18], we expect this approximation to be fairly accurate for low times, though this is not necessary for the proof.

Consider the tree $\tilde{G}$ formed as follows. The root of the tree is the initial infected node. The next level contains copies of all nodes adjacent to the original node in the Erdös-Renyi graph. Each of these have descendants that are copies of their neighbors, and so on. Note all nodes may (and likely do) have multiple copies.

We start an infection at the root of $\tilde{G}$ and let it spread for time t. Consider the induced set of infected nodes, $\tilde{S}_{\text{rep}}$, as the set of nodes in G which have copies that are infected on $\tilde{G}$. Since the distance of a copy from the root of $\tilde{G}$ is no less than the distance from the original node to the original infection source, we see that the distance the infection has traveled on $\tilde{G}$ is no less than the distance from the infection source to the farthest node in $\tilde{S}_{\text{rep}}$ (on G). Note that the $\tilde{S}_{\text{rep}}$ stochastically dominates the true infected set S. That is, for all sets T, $P(T \subset \tilde{S}_{\text{rep}}) \geq P(T \subset S_{\text{rep}})$.

This stochastic dominance result follows from the fact that the transition rates are universally equal or higher for the induced set. Hence, we conclude $\text{RadiusBall}(G, S_{\text{rep}})$ is also stochastically dominated by $\text{RadiusBall}(G, \tilde{S}_{\text{rep}})$, and the latter is upper bounded by the depth of the infection in the tree, which using the speed result, is bounded by $C_6 t$ for some speed C_6. That is, with probability tending to 1,

$$\text{RadiusBall}(G, S_{\text{rep}}) < C_6 t.$$

□

Next, we use the neighborhood sizes on this graph to provide a lower bound to the ball size needed to cover a random infection.

Lemma 3.3.6. *Let $G^{(n)} = G(n,p)$, and let $S^{(n)}$ denote a collection nodes sampled uniformly at random from $G^{(n)}$, such that $\left|S^{(n)}\right|$ scales at least with* $\log n$. *Then*

$$\mathrm{RadiusBall}(G^{(n)}, S^{(n)}) > \frac{\log n}{3 \log c},$$

with probability converging to 1 *as $n \to \infty$.*

Proof. We suppress the index n for clarity. We proceed by bounding the probability that all the random nodes are within a ball of radius m. This is possible only if all nodes in S are within distance $2m$ from any given node in S. Now, the number of nodes within a distance $2m$ from a given node is no more than $16m^3c^{2m}\log n$ with probability $1 - o(n^{-1})$ [11]. Then the probability of all nodes fitting inside one such ball is at most

$$\left(\frac{16m^3c^{2m}\log n}{n}\right)^{|S|-1} < \left(\frac{16m^3c^{2m}\log n}{n}\right)^{\log n - 1}.$$

Then this decays to 0 at least as fast as n^{-1} if

$$\frac{16m^3c^{2m}\log n}{n} < n^{-1/\log n}.$$

Finally we set $m = \frac{\log n}{3\log c}$ as desired. Hence $c^{2m} = n^{2/3}$. Using this substitution, the above term reduces to

$$\begin{aligned}\frac{16m^3c^{2m}\log n}{n} &= \frac{16m^3n^{2/3}\log n}{n} \\ &= \frac{16(\log n)^4}{27(\log c)^3 n^{1/3}} \\ &< (\log n)^4 n^{-1/3} < n^{-1/\log n} \qquad (3.2)\end{aligned}$$

for sufficiently large n. Therefore, RadiusBall$(G, S) > \frac{\log n}{3 \log c}$ with probability converging to 1. □

The diameter of the giant component of an Erdös-Renyi graph is $\Theta(\log n)$ [18]. Thus, Lemmas 3.3.5 and 3.3.6 establish that an Erdös-Renyi graph satisfies both the speed and spread conditions respectively.

3.4 Simulations

We simulated the performance of the Comparative Ball Algorithm to evaluate the performance empirically. We determined the error rate over a range of t for several pairs of graphs. We evaluated the two different standard graph topologies considered earlier, grids and Erdös-Renyi graphs.

We simulated the infections on various pairs of the graphs over a range of times. In order to portray the results in a comparable way, we plotted the error rate versus the average infection size instead of time. This is necessary because different times result in very different infection sizes for the different graphs. That is, the infection is large even at low t on an Erdös-Renyi graph, and vice versa for a grid graph. This would introduce a misleading effect in the results.

Each node in the graphs received a random label to ensure independence. We use $n = 1,600$ for each graph with $q = 0.25$. For the Erdös-Renyi graphs, we use $p = 2/1,600$. The probability of error was computed over $10,000$ trials. There are two possible types of errors in each simulation, when

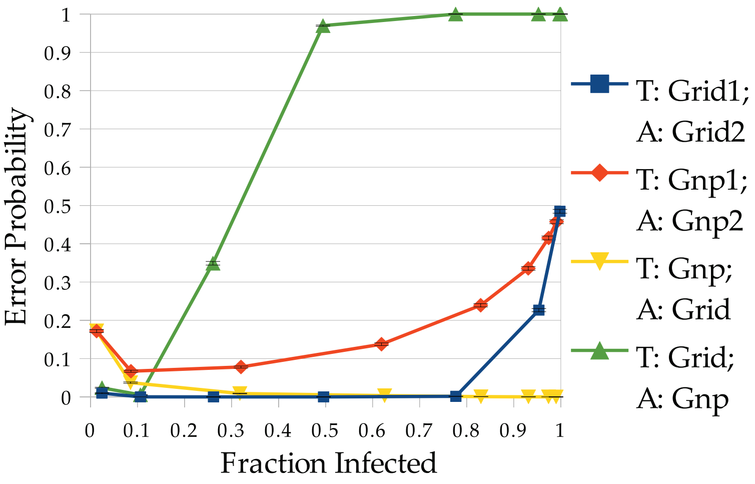


Figure 3.2: This figure shows the error probability for the algorithm on pairs of standard graphs. Various (conditional) error probabilities are illustrated – 'T:' corresponds to the true network, and 'A:' corresponds to the algorithm output.

the infection spreads on the first graph, and when it spreads on the second. We label the error event 'T:G_1; A:G_2' for the error where the infection in fact travels on graph G_1 (True event), but the algorithm incorrectly labels it as occurring on graph G_2 (Algorithm output).

The results of these simulations are shown in Figure 3.2. Note that up to about 5% of the network reporting an infection, the error rates are low in all cases. The error rates are consistently low for the 'T:Grid1;A:Grid2' comparison up to the point where the whole network is infected. When comparing a grid and an Erdös-Renyi graph, there is a bias to label it an Erdös-Renyi graph at higher times, causing the 'T:Grid;A:G(n,p)' error to be very high and conversely, the ‘T:G(n,p);A:Grid’ error to be very low. This bias results from the fact the diameter of the graph is not necessarily the optimum scaling for the Comparative Ball Algorithm. Though (as shown in our theoretical results) the two graphs can be still be distinguished at lower infection sizes, using suboptimal scaling means that overall error probability will be high for large infections, with a bias toward one of the graphs. This suggests that by simply modifying the Comparative Ball Algorithm to normalize with respect to a scaled graph diameter (where the scaling parameter would be graph dependent), we could balance these two error probabilities, and thus result in improved performance. To illustrate, by choosing a diameter scaling value of 1.6 for the Grid graph, the plot in Figure 3.3 indicates that one could distinguish between G(n,p) and Grid graphs for a significantly larger range.

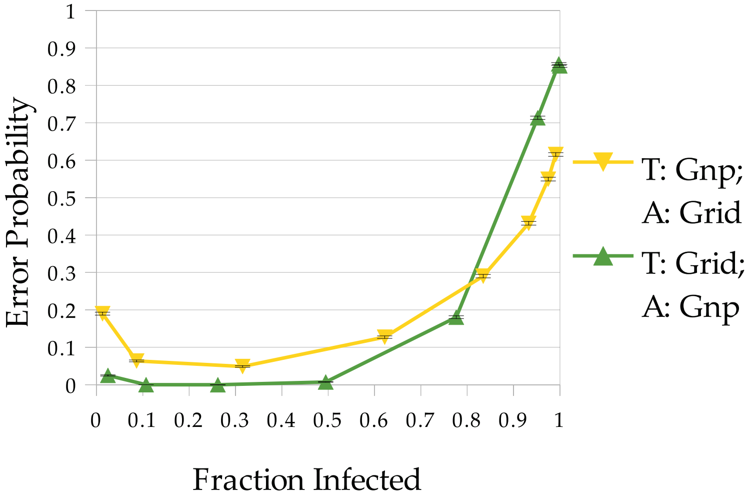


Figure 3.3: This figure shows the error probability for the G(n,p) vs. Grid graphs for the scaled diameter setting (diameter of G(n,p) graph is scaled by 1.6).

Chapter 4

False Positives

4.1 Introduction

Identifying the process causing an infection can be essential to reacting appropriately, and it is challenging when the set of sick nodes is incomplete, and especially inaccurate. In previous chapters, we demonstrated that by using the clustering of the sick nodes, it is possible to distinguish between a random sickness and an infection, and between two different infections. However, the algorithms employed were very sensitive to outliers. A single false positive, a node reporting sickness when it is not actually infected, can drastically change how clustered the algorithms rate the sick nodes. In particular, when any node at maximum distance from the infection source falsely reports sickness, then the ball algorithm will conclude that the sick nodes are maximally spread. That is, it will never be able to identify the infection.

However, real data is often inaccurate. For example, online records (flu-related keywords in social networks [13], or Internet searches such as in

The work in this chapter appears in the following publication:
Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. Detecting epidemics using highly noisy data. In *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 177–186, 2013.

Google Flu Trends [23]) provide large but noisy data sources for detecting flu epidemics, but potentially containing many false positives. In evaluating the spread of the flu, there may be many people reporting flu-like symptoms, but have a different ailment. To be useful in in practical applications, the algorithm must be modified to be robust. This requires filtering out false positives before evaluating the clustering of the sick nodes. These false positives may occur in two possible ways. In the easier case, the false positives are randomly spread out over the network. In the second case, the false positives may be placed arbitrarily, such as chosen by an adversary so that the problem is as difficult as possible. For example, if the sick nodes were created by a random sickness, the adversary may place the false positives in a cluster so that the sickness appears closer to an infection.

We consider the task of distinguishing a random sickness from an infection in the presence of false positives. After the infection proceeds for some time, a fraction of the sick nodes report the sickness. Then, we add a number of false positives proportional to the number of reporting nodes. In other terms, a fixed fraction of the complete set of reporting nodes are false positives. These false positives are either arranged randomly or are chosen by an adversary. Under these conditions, the problem is to determine whether the original sick nodes are due to a random sickness or infection.

We develop a robust algorithm based on the ball clustering to solve these problems, called the *Quantile Ball Algorithm*. We prove that this algorithm can distinguish the processes for a wide range of infection sizes in the presence of false positives. When the false positives are located randomly, the algorithm succeeds with high probability when the fraction of reporting nodes that are false positives is any value less than one, though the largest infection that can be successfully detected decreases when there are larger numbers of false positives. For adversarial placement of the false positives, we show that the algorithm succeeds as long as the faction of nodes that are false positives is less than one half. This is the best theoretically possible.

4.2 Problem Statement

The fundamental aspects of this problem are the same as in basic problem from Chapter 2. The random sickness versus infection problem is setup as before. The infection spreads according to the SI model for time t. For the random sickness, each node is randomly and independently infected with probability q'. This probability is set so that the expected size of the random sickness is the same as that of the infection. Each of the sick nodes reports with a fixed probability q. See Section 2.2 for additional details. Next we formally describe the necessary graph properties, as well as the false positives and mixed infection.

4.2.1 Graph Conditions

We assume the graphs are sufficiently well behaved that it is possible to distinguish an infection process. These conditions are similar to the conditions for a graph to be *detectable* as described in Chapter 3, but slightly more detailed. These conditions guarantee that first, the infection spreads only up to a fixed maximum speed, and second, that the random nodes are spread out. However, we require somewhat more complex conditions for our results. The conditions are labeled *limited epidemic speed* and *limited neighborhood size*.

Definition 4.2.1. A graph family $\mathcal{G}$ has *limited epidemic speed* if there exist finite, positive constants $s_{\mathcal{G}}$, $\lambda_{\mathcal{G}}$ such that for sufficiently large n, a graph $G^{(n)}$ chosen randomly from $\mathcal{G}^(n)$ and an epidemic starting at any node a with duration $t^{(n)}$, with $S^{(n)}$ defined as the set of nodes infected at time $t^{(n)}$,

$$P(\text{RadiusBall}(S^{(n)}) > s_{\mathcal{G}} t^{(n)}) < e^{-\lambda_{\mathcal{G}} t^{(n)}}.$$

The speed $s_{\mathcal{G}}$ in the above definition is in fact an upper bound on the speed, in that we require no matching lower bound. Nevertheless, we refer to it as the speed for brevity. In addition, we also need a constraint on the neighborhood size.

Definition 4.2.2. A graph family $\mathcal{G}$ has *limited neighborhood size* if, for graph $G^{(n)}$ chosen randomly from $\mathcal{G}^{(n)}$, $\text{diam}(G^{(n)})$ scales as $\Omega(\log n)$ and there exists a increasing concave function $b_{\mathcal{G}}^{(n)}(x)$ such that for all $1 \leq x$, $b_{\mathcal{G}}^{(n)}(x) > 0$ and all balls on $G^{(n)}$ of radius no more than $b_{\mathcal{G}}^{(n)}(x)$ contains less than x nodes with probability tending to 1.

These conditions hold for typical graph topologies such as grids and Erdös-Renyi graphs, as can be seen from the proofs for the results in Section 3.3.3. In fact, both of these previous conditions follow for any graph with a bounded degree distribution, as stated formally below. For these graphs, the neighborhood size function does not vary with n. We note that there are multiple choices for this function, but by using tighter functions (accounting for the exact graph topology), the sufficient conditions given in our results are improved.

Theorem 4.2.1. *Let $\mathcal{G}$ be a graph family whose graphs have maximum degree $\bar{d}$. Then $\mathcal{G}$ has both limited epidemic speed and limited neighborhood size.*

Proof. First, the spread of the epidemic on any graph $G^{(n)}$ from $\mathcal{G}$ can be upper bounded by a tree of degree $\bar{d}$ where nodes are repeated for each path to them. See [47] for details on this bound. Then using a speed upper bound for trees, we find that $\mathcal{G}$ has *limited epidemic speed*, where the exponential probability of error follows from a Chernoff bound [7]. Next, using the maximum degree condition, the number of nodes within distance r from an arbitrary node u of any graph $G^{(n)}$ is at most $\bar{d}^{r+1}$. Therefore, for any x, $1 \leq x$, no ball of radius $\log_{\bar{d}} x - 1$ contains more than x nodes. From this, we see that $\text{diam}(G^{(n)}) \geq \log_{\bar{d}} n - 1$. Letting $b_{\mathcal{G}}(x) = \log_{\bar{d}} x - 1$, we see this satisfies the desired condition for *limited neighborhood size.* This completes the proof. □

We suppress the index (n) on the graph G and the infection parameters when it is clear from context. Likewise, we omit the index (n) and subscript $\mathcal{G}$ from $s_{\mathcal{G}}$, $\lambda_{\mathcal{G}}$, and $b_{\mathcal{G}}^{(n)}(x)$ for clarity. When it is clear from context, we reference the family $\mathcal{G}$ by a representative graph G from that family. That is, we say a G has limited epidemic speed and limited neighborhood size if its family $\mathcal{G}$ does.

We assume that the speed $s_{\mathcal{G}}$ and spread function $b_{\mathcal{G}}^{(n)}(x)$ are known. Next, the following simple lemma (using a balls-in-bins argument) proves useful in the sequel, so we give it here.

Lemma 4.2.2. *Consider graph G. Let $0 < x < 1$ and $\delta > 0$. Then there exists ϵ depending on δ with $0 < \epsilon < 1$ such that the following is true. Let S be a collection of nodes chosen uniformly at random with $|S| = \Omega(\log n)$. Let B be a collection of nodes with $|B| < (1 - \epsilon)xn$. Then the probability that B contains at least x fraction of the random nodes in S decays to 0 as n increases. In particular,*

$$P(|B \cap S| \geq x\,|S|) < e^{-\delta|S|}.$$

The main way we use this lemma is to show that the probability that a large fraction of randomly selected nodes fall in a ball around a given node, goes to zero.

4.2.2 False Positives

In this problem, we add false positives to the set of reporting sick nodes. Then, only a fixed fraction of the reporting nodes available to the algorithm reflect nodes that are actually sick. Define S_{rep} to be the set of reporting sick nodes. Let f be a fixed constant with $0 < f < 1$, representing the relative fraction of false positives compared to truly sick nodes. We will then add false positives to get $\bar{S}_{\text{rep}}$, the set of both reporting sick nodes and false positives, which is then made available to the algorithm. Set the number of false positives to be $f\,|S_{\text{rep}}|$. Note that the fraction of all the reporting nodes that are false positives is $\frac{f}{1+f}$.

The false positives are then added either randomly or by an adversary. In the random setting, choose $f\,|S_{\text{rep}}|$ nodes uniformly over the entire graph. If $f\,|S_{\text{rep}}| > n$, then only n nodes are chosen, though distinguishing infections is impossible at this point. We allow nodes that are already in S_{rep} to be chosen. Let A_{rep} be these false positive nodes. Then $\bar{S}_{\text{rep}} = S_{\text{rep}} \bigcup A_{\text{rep}}$. Note then that there may be less than $f\,|S_{\text{rep}}|$ false positives. However, this effect will be small for small infections.

In the adversarial regime, the false positives are placed arbitrarily. In particular, the adversary places the false positives in whatever way would lead to the highest probability of error for our algorithm. Then, we require the algorithm to be able to handle any arrangement of the false positives. As before, defining A_{rep} as the set of false positives, set the complete set of reporting nodes $\bar{S}_{\text{rep}} = S_{\text{rep}} \bigcup A_{\text{rep}}$. We allow the adversary to choose repeats as in the random arrangement case, though generally this makes the problem easier.

4.2.3 Algorithm

To solve this problem, we use a modification of the Threshold Ball Algorithm (from Section 2.3) called the *Quantile Ball Algorithm*. It is defined in terms of a parameter α, with $0 < \alpha \leq 1$. In this algorithm, we find the smallest radius ball that contains a fraction α of the reporting nodes. That is, the algorithm is given the set of all reporting nodes $\bar{S}_{\text{rep}}$. Then it finds the ball of minimum radius containing at least $\alpha \left|\bar{S}_{\text{rep}}\right|$ of the reporting nodes in $\bar{S}_{\text{rep}}$. This lets the algorithm ignore the worst fraction of the reporting nodes. The algorithm uses a threshold r on this infection size as the maximum radius such a ball can have to be labeled an EPIDEMIC. However, it can reduce the number of true infected nodes that are evaluated, which reduces the accuracy of the algorithm. The algorithm is specified formally as follows.

Algorithm 5 Quantile Ball Algorithm

```
Input: Graph $G$; Set of reporting infected nodes $S_{\text{rep}}$;
Parameters: Quantile $\alpha$, Threshold $m$
Output: EPIDEMIC or RANDOM

  $c \leftarrow \alpha\,[|S_{\text{rep}}|]$
  for all $(u \in V$ do
    $B \leftarrow \text{Ball}_G(u, m)$
    if $|B \cap S_{\text{rep}}| \geq c$ then
      return EPIDEMIC
    end if
  end for
  return RANDOM
```

4.3 Main Results

We will establish several sufficient conditions for when the Quantile Ball Algorithm can successfully determine the causative process of the infection. In particular, we show that the probability of error decreases to 0 for reasonable ranges of infection sizes for any value of f (the ratio of false positives to true reporting nodes) if the false positives are arranged randomly. For adversarial false positives, it is possible for $f < 1$ (over half the reporting nodes are actually sick). The first case we consider is when the nodes are located randomly.

4.3.1 Randomly Located

When the false positives are spread randomly through the graph, then the behavior of a random sickness does not change: it is still an unclustered set of sick nodes. That is, if we find any large set of clustered nodes, the sickness is very likely to be an infection. By filtering out a sufficient number of false positives, this case becomes roughly equivalent to the basic random sickness vs. infection problem. Then we expect that this case is substantially easier than the adversarial case. We show that is in fact the case, and that we can distinguish a random sickness from an infection even with an arbitrarily high faction of false positives.

Theorem 4.3.1. *Let $f > 0$. Assume the number of reporting nodes is $\omega(\log n)$. Then there exists a constant C_0 such that if the infection time satisfies $t < b\left(\frac{n}{C_0(1+f)}\right)/s$, using the Quantile Ball Algorithm, setting the parameters $\alpha = 1/(1+f)$ and $m = st$, the infection type can be correctly distinguished with probability approaching to* 1.

Proof. The proof proceeds in a very similar way to Theorem 4.3.2. First suppose the infection is an epidemic. We can cover all true reporting nodes with probability scaling to 1 using the speed definition. Since at least an α fraction of the reporting nodes are truly infected, our algorithm correctly reports the infection is an epidemic. Therefore the Type II error probability decays to 0.

Now suppose the infection is a random sickness. Since the false positives are also random, the reporting nodes with the false positives are simply are larger set of random nodes. Define C_0 as the same constant as in the proof of Theorem 4.3.2. Assume $m < b\left(\frac{n}{C_0(1+f)}\right)$. Using Lemma 4.2.2 in the same way as previously, we see that no ball of radius m contains over a $\alpha = 1/(1+f)$ fraction of the random nodes with probability approaching 1. In this case, our algorithm returns random sickness. Thus the Type I error probability also tends to 0. □

Roughly speaking, this means that for infection times less than an upper bound order-wise the same as the time to infect a constant fraction of the network, the Quantile Ball Algorithm successfully distinguishes a random sickness from an infection with high probability and for any fraction of false positives. This is possible by choosing α to eliminate the false positives.

4.3.2 Adversarial

Next, consider the adversarial regime, where false positives are placed by an adversary seeking to maximize our probability of error. The Quantile Ball Algorithm succeeds in this case as well.

Theorem 4.3.2. *Suppose G is as described. Suppose further that $f < 1$ and set $f' = (1-f)/(1+f) > 0$. Suppose t scales such that the number* *of reporting nodes is $\Omega(\log n)$. Then there exists a constant C_0 such that if $t < b(f'n/C_0)/s$, the Quantile Ball Algorithm with $\alpha = 1/(1+f)$, and $m = st$ correctly determines the type of infection with probability tending to 1 with the number of nodes, n.*

Proof. First we show that the Type II error probability decays to 0. To this end, suppose the infection is in fact an epidemic. Consider only the true reporting nodes S_{rep}, and recall $\bar{S}_{\text{rep}}$ is the set of all reporting nodes, including the false positives. Note that $|S_{\text{rep}}| \geq \alpha \left|\bar{S}_{\text{rep}}\right|$. By the definition of speed s, the probability the epidemic spreads outside a ball of radius $m = st$ decays to 0, so this ball covers S_{rep} and hence at least α fraction of the reporting nodes. Therefore it is correctly labeled an epidemic.

Now we show that the Type I error probability also decays to 0. We need to show no ball of radius m can cover $\alpha = 1/(1+f)$ fraction of the nodes. Since only $f/(1+f)$ of the nodes are false positives, the ball must contain at least $(1-f)/(1+f) = f' > 0$ true reporting nodes. Then it is sufficient that the probability there exists a ball of radius m covering $f' |S_{\text{rep}}|$ true reporting nodes (which are located randomly) decays to 0.

By assumption, for some constant C', $|S_{\text{rep}}| > C' \log n$ for sufficiently large graphs. Let $\delta = 3/C'$ and $\epsilon > 0$ as guaranteed by Lemma 4.2.2. Set $C_0 = 1/(1-\epsilon)$ and assume $m < b(f'n/C_0)$. Therefore, no ball of radius m contains over $f'n/C_0$ nodes. Consider one of the n balls of radius m (one ball for each possible center node), call it B. Then by Lemma 4.2.2,

$$P(|B \cap S_{\text{rep}}| \geq f' \, |S_{\text{rep}}|) < e^{-\delta |S_{\text{rep}}|}.$$

Then for sufficiently large n, $e^{-\delta |S_{\text{rep}}|} = o(1/n^2)$. Therefore, from a union bound, there is some ball of radius m containing over f' fraction of the true reporting nodes with probability at most $o(1/n)$. Hence, no such ball covers α fraction of the nodes in $\bar{S}_{\text{rep}}$ with probability tending to 1 so the Type I error probability goes to 0. □

That is, for a similar bound on the infection duration as before, the Quantile Ball Algorithm can succeed for any $f < 1$. This means that as long as the true infected nodes are in the majority, it is possible to distinguish a random sickness from an infection. With some thought, it is clear that this is the best possible for any algorithm (in terms of size of f). If the number of false positives were the same as (or more than) the number of true reporting nodes, then it is possible for the adversary to completely imitate the incorrect infection process. This fact is given in the following theorem.

Theorem 4.3.3. *Suppose $f = 1$ and the random sickness is normalized so that the infection size distribution is equal for both infection processes. Then the probability of error for any algorithm is at least* 0.5.

Proof. There is a simple adversarial algorithm that guarantees a probability of error of 0.5. Recall the *a priori* probability for each infection process is equal. When the infection is from an epidemic, the adversary chooses nodes randomly exactly as in the random sickness. When the infection is from a random sickness, the adversary chooses nodes exactly as in an epidemic. Therefore, in all cases, exactly half the nodes are due to an epidemic, and half are due to a random sickness. Since the infection size is normalized, each collection of infected nodes is equally likely to be an epidemic as a random sickness. Then the probability of error for every set $\bar{S}_{\text{rep}}$ is 0.5 (no matter the algorithm), and hence the overall probability is 0.5. □

4.4 Simulations

We evaluate the Quantile Ball Algorithm by the empirical error probability, the average error probability for both Type I and Type II errors, weighting both equally. We used a grid graph with $n = 4900$, and infection time $t = 10$. The reporting probability was fixed at $q = 0.25$. The infection was simulated for 1000 trials for each infection processes (a random sickness and an epidemic), running the Quantile Ball Algorithm for each set of reporting nodes. The expected size of the random sickness was normalized to the empirical average size of the epidemic. We set the ball size parameter m to the optimal value as determined empirically. The probability of error is plotted against the empirical expected fraction of infected nodes. That is, for each set of parameters, we estimated the expected number of infected nodes from the simulations, which was divided by n to determine the fraction infected. This expected fraction of infected nodes conveys the size of the infection, and hence the difficulty of the problem (since the task is more difficult the larger

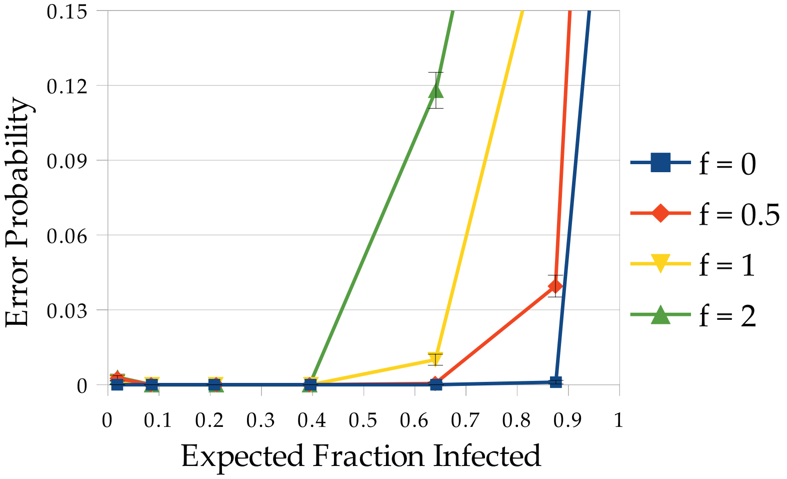


Figure 4.1: This figure shows the overall error probability, the sum of equally weighted Type I and Type II error rates, for a grid graph. The false positives were located randomly on the graph. The x-axis measures the expected fraction of nodes truly infected. As in our results, $\alpha = 1/(1+f)$. The ball radius m was set to the optimal value empirically.

the infection is). Note that since $q = 0.25$, the expected fraction of reporting nodes is approximately 0.25 times as large.

We present our simulation results on the probability of error for grid graphs for a variety of false positive frequencies. As in our analytical setting, the random sickness infection size was normalized to the same distribution as the epidemic as determined empirically. The results are shown in Figure 4.1.

The error probability is very low up to a very large number of truly infected nodes. It climbs fairly slowly as the number of false positives increases. Even when two-thirds of the reporting nodes are false positives, the error probability is low even up to an expected 40% of the network infected. Therefore our algorithm works very well in this setting.

Chapter 5

Mixed Infections

5.1 Introduction

The study of epidemic spread over social, communication, and human contact networks, be it a contagion of a human or computer virus, or a rumor, opinion or trend, begins with two basic questions: do we indeed have a spreading epidemic, and if so, what is the causative network spreading it? Numerous famous examples from the history of epidemiology ([57, 12]) have illustrated the importance and difficulty of determining the causative network. With accurate data collected over time, for example, from high accuracy medical diagnoses of a known illness, the causative network essentially reveals itself. Yet such data are rarely available. More to the point, highly incomplete and noisy data *often are available*. Indeed, the challenge arises in particular, when time lapse data of "true" illness is not available, and when the data we do have is highly noisy.

The key idea in this work, is that different spreading mechanisms have

The work in this chapter appears in the following publication:
Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. Detecting epidemics using highly noisy data. In *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 177–186, 2013.

different statistical signatures, in terms of the subset of people infected. This is certainly the case when the causative graphs are very different, and the subset of nodes (people, machines, etc.) the epidemic has reached ("infected nodes") are completely and accurately revealed. As discussed, however, the data available are typically noisy. Moreover, the larger the fraction of the network the contagion has reached, the more this "network signature" is washed out. This paper explores these tradeoffs. We consider a broad class of graphs: graphs with bounded degree. The degree controls the infection's speed. We consider the case most relevant in spread of rumors, technology and ideas: the superposition of two spreading mechanisms. Indeed, in the age of mass advertising and mass media, trends spread friend-to-friend, but also through television, Internet ads, and similar advertising efforts that exhibit a "star-like" contagion network [27].

We consider two such mixed processes. In each process, the infection and random sicknesses occur at different rates, but the network is the same for both. One mixed process however is more infectious than the other. That is, in one process, the infection rate is much higher than the random sickness, and in the other, the random sickness dominates. Note that this is a generalization of the problem of distinguishing a random sickness from an infection. However, now there will be outliers that makes the problem more challenging. We provide sufficient conditions for determining which is the dominant effect, when only a vanishing fraction of infected nodes report, and when no time-lapse data are available.

5.1.1 Related Work

Analyzing the spread of epidemics under the susceptible-infected (SI) model [20] has been considered in depth for a variety of graphs and circumstances [4, 24]. Myers *et al.* consider a problem similar to the mixed infection regime [50]. That is, nodes infect each other through the network as usual, but in addition, nodes are become sick randomly as well. In their model, nodes become 'exposed' to information, and may decide become infected, exposing their neighbors to the information, with a probability dependent on the number of times they were exposed. An external source exposes nodes randomly at a time-varying rate, and nodes may expose their neighbors after they are infected. Their goal is to estimate the external infection rate and other algorithm parameters. To accomplish this, the full sequence of infected nodes and the times they were infected is required. The difference between this result and the proposed problem, beyond the model differences, is first, here only a partial set of the infected nodes is known. This is a nontrivial restriction, heavily impacting the algorithm used. On the other hand, our algorithm must only distinguish between two distinct processes, as opposed to estimating the random sickness rate. As in the case of estimating the graph structure (Section 2.1.2), the key distinguishing factor of our work is the minimal amount of information available to the algorithm.

5.2 Problem Statement

We consider two infection processes spreading on the same graph G. At a single instant in time, some portion of the infected nodes report being infected, and we must use this information to evaluate which infection process most likely caused the infection. We use the same reporting process as in Chapter 2, where each sick node reports with probability q. In this case however, instead having a pure random sickness and a pure epidemic, both infection processes are a mixture of both a random sickness and an epidemic. Equivalently, the processes are mixtures of an epidemic on a star graph and an epidemic on a well-structured graph. We refer to these as mixed infections. However, one process is dominated by the random sickness process and takes the role of the random sickness. Phrasing this as a hypothesis testing problem, our null hypothesis is that the mostly random infection process is the cause of the infection. The alternative is that the other epidemic dominated mixed infection is the causative process.

5.2.1 The Infection Process

Let $G = (V, E)$ denote the graph along which the infection spreads. As discussed above, in the case of an epidemic spreading node-to-node, G is a structured graph (e.g., d-dimensional grid). The initial node of the infection is selected uniformly at random. We let $n = |V|$, the size of the graph. The diameter of the graph is denoted $\mathrm{diam}(G)$.

Given a graph G, the contagion spreads as follows. At time zero, an initial node is selected and called “infected.” For the structured graph case, we assume this initial infected node is selected uniformly at random. For the star graph, it is the external central node. The infection spreads from that node to its neighbors, across the edges of the graph. The spreading occurs according to a standard susceptible-infected (SI) model [20, 18, 34] for an epidemic. The spreading rate is parameterized by a single number, or rate. To make clear the distinction between the rate for a structured graph or for a star graph, we use η to represent the rate of the structured graph, and γ/n the rate of the star graph. We divide by n in the case of the star graph so that new infections appear at rate γ (ignoring the shrinking number of susceptible nodes). This means the following: for each infected node and for each edge incident to that node, we start an exponential clock, i.e., a clock that expires after an exponentially distributed length of time, of expectation $1/\eta$, i.e., of rate η for a structured graph, and n/γ, i.e., of rate γ/n, for the star graph. The expiration of a clock indicates that the adjacent node becomes infected (if it is not already infected) and new clocks are started for each edge from this newly infected node. In this way, the infection spreads along the edges of the graph in a node-to-node fashion.

The star graph infects nodes at rate γ/n, and then these infected nodes infect their neighbors on the structured graph (e.g., the grid) at rate η. Thus, in this superposed process, nodes become infected at random at some rate γ, which we term ‘seeds’. The infection then spreads from these seeds as an epidemic on graph G at the (different) rate η. With the combination of these processes, the infection will appear as multiple ‘balls’ of decreasing size. The first infection will be much larger, followed by smaller balls and then (possibly) individual infected nodes.

In this setting, we consider two different processes: one where the dominant factor is the random infection (the spread from the star graph) and the other where it is the spread along the structured graph that dominates. Thus, in the first setting we have $\gamma \gg \eta$, and the random infection dominates the epidemic, and in the second setting, $\eta \gg \gamma$, and the epidemic spread dominates the infection process. We define S as the set of infected nodes at a given time t, and let S_{rep} be the set of reporting infected nodes.

5.2.2 Graphs

We consider on graphs G chosen from family $\mathcal{G}$ with constant bounded degree. That is, suppose that for a constant $\bar{d}$, every vertex in the graph has degree no more than $\bar{d}$ for each graph in that family. This condition suffices to limit the speed at which the epidemic can spread through the network, and otherwise makes the epidemic well behaved. In particular, from Theorem 4.2.1, the graph has *limited epidemic speed* and *limited neighborhood size*. See Section 4.2.1 for details on these conditions. We restate these conditions for the reader’s convenience.

Definition 5.2.1. A graph has *limited epidemic speed* if there exist finite, positive constants s, λ_1 such that for sufficiently large n and time t, and an epidemic starting at any node a,

$$P(v(a,t) > st) < e^{-\lambda_1 t}.$$

Definition 5.2.2. A graph G has *limited neighborhood size* if $\mathrm{diam}(G)$ scales as $\Omega(\log n)$ and there exists a increasing concave function $b(x)$ such that for all $b(x) > 0$ and all balls of radius no more than $b(x)$ contain less than x nodes for sufficiently large n with probability tending to 1.

5.2.3 Algorithm

We use an extension of the Quantile Ball Algorithm from Section 4.2.3. Like in that algorithm, we use a parameter α satisfying $0 < \alpha \leq 1$, and only look at the α fraction most clustered nodes. However, in this case, we also use β balls to contain the infected nodes, with $\beta \geq 1$, since there may be multiple clusters from the epidemic.

We term our algorithm the Multiple Ball Algorithm. The Multiple Ball Algorithm is simple to describe: it searches for the smallest ball/collection of balls that covers a minimum fraction of the reporting infected nodes. Of course, it has no way to tell if a reporting sick node is truly infected or a false positive, and as emphasized above, this is not the goal of this paper. If the resulting radius of this ball is small enough, it declares that there is an epidemic; otherwise, it concludes that the infection process is in fact a random illness. This algorithm is efficient, as even the brute-force implementation runs in time at most $O(|V|^2 \cdot |E|)$ when there is a single ball, and in general, order-wise polynomial in $|V| \cdot |E|$.

The algorithm takes three parameters α, β and m. These parameters are tailored to the problem at hand, including, in the case of m, the size of the graph. As input, it takes a graph G and a set of reporting infected nodes S_{rep}. If the algorithm can cover an α-fraction of the infected nodes with β balls, each of radius at most m, it declares the infection to be an epidemic; otherwise, it labels the infection a random illness. In most cases, it is sufficient to use a single ball (that is, $\beta = 1$).

Algorithm 6 Multiple Ball Algorithm

Input: Graph G; Set of reporting infected nodes S_{rep};
Parameters: Quantile α, Number of Balls β, Threshold m
Output: EPIDEMIC or RANDOM

```
$c \leftarrow \alpha\,[|S_{\text{rep}}|]$
for all $(u_1, u_2, \ldots, u_\beta) \in V^K$ do
  $B \leftarrow \bigcup_{1 \le i \le \beta} \text{Ball}_G(u_i, r)$
  if $|B \cap S_{\text{rep}}| \ge c$ then
    return EPIDEMIC
  end if
end for
return RANDOM
```

In the basic Multiple Ball Algorithm, we considered the case when all balls have the same radius. However, in many cases, when multiple balls are used, it makes sense to have some balls smaller than others. For example, the epidemic won't spread as far from a node that became randomly sick late into the infection, compared to the node that initially sick. To account for this, we also consider a modification of the previous algorithm called the Scaling Multiple Ball Algorithm. In this algorithm, the radius of the balls scales linearly up to the radius of the largest ball, m. When there is only one ball ($\beta = 1$), this algorithm is identical to the previous one.

Algorithm 7 Scaling Multiple Ball Algorithm

Input: Graph G; Set of reporting infected nodes S_{rep};
Parameters: Quantile α, Number of Balls β, Threshold m
Output: EPIDEMIC or RANDOM

```
$c \leftarrow \alpha\,[|S_{\text{rep}}|]$
**for all** $(u_1, u_2, \ldots, u_\beta) \in V^K$ **do**
    $B \leftarrow \bigcup_{1 \le i \le \beta} \text{Ball}_G(u_i, ri/\beta)$
    **if** $|B \cap S_{\text{rep}}| \ge c$ **then**
        **return** EPIDEMIC
    **end if**
**end for**
**return** RANDOM
```

Both forms of the Multiple Ball Algorithm take computation time exponential in β. This time can be substantially reduced by modifying the algorithms to be greedy. More precisely, instead of optimizing over all possible collections of balls, the greedy algorithm first tries to cover as many reporting nodes as possible with the largest ball. Then, it covers as many of the remaining reporting nodes as possible with the next largest ball, and so on. The resulting algorithm is much more efficient when there are a large number of balls, but may return an incorrect result. We analyze only the exact forms of the Multiple Ball Algorithm unless otherwise stated.

5.3 Main Results

Mixed processes, with both an infection component and random sickness component, can be distinguished in a similar way as in the case of false positives from Chapter 4. This is because, if the infection component dominates, the initial infection will be much larger than the others, so the secondary infections can be treated as outliers. Likewise, if the random component dominates, the infections from the many random seeds will be spread over the graph, so no small ball can contain many of the infected nodes.

We consider two distinct infection processes. In Process 0, the infection spreads mostly randomly. Let γ_0, η_0 be the infection rates for the random sickness and epidemic respectively and t_0 be the infection time for Process 0. For clarity, we also call Process 0 "Process SR-WE" (Strong random, weak epidemic). In Process 1, the infection is dominated by the epidemic, and let γ_1, η_1, and t_1 be the corresponding parameters as before. We label Process 1 "Process WR-SE" (Weak random, strong epidemic). Note that the infection is the same if the rates are scaled up by the same factor that time is scaled down. Then we can say that the epidemic dominates in Process 1 relative to Process 0 if $\eta_1/\gamma_1 \gg \eta_0/\gamma_0$. Unlike in the previous chapters, we apply no explicit normalization. Rather, we provide sufficient conditions on the range of the parameters for which the Multiple Ball Algorithm succeeds.

Theorem 5.3.1. *Consider an infection spreading as in Process* 0*. Suppose* $q\gamma_0 t_0 = \omega(\log n)$*. Suppose there exists a constant integer* $C_3 \geq 1$ *where* $\eta_0 t_0 =$

$o\left((\gamma_0 t_0)^{-1/(1+C_3)}\right)$ *and for some* $\epsilon > 0$*, suppose that* $m + C_3 < b\left(\frac{\alpha n}{\beta \bar{d}^{C_3+1}(1+\epsilon)}\right)$*. Then the Type I error probability for both the Multiple Ball Algorithm and Scaling Multiple Ball Algorithm decays to* 0 *as* n *increases.*

Proof outline. The conditions in the theorem are sufficient to show that the epidemic will not spread more than a constant distance from any seed. Due to this, it is sufficient to show that an α fraction of the seeds cannot be contained by the balls. Standard bounds on the spread of a random sickness is sufficient to complete the proof. See the appendix for the details of the proof. □

Next consider the infection spreading by Process WR-SE [Process 1]. Then we can characterize the range for which the Type II error goes to 0 as follows.

Theorem 5.3.2. *Consider an infection from Process* 1*. Suppose* $m > s\eta_1 t_1$*, where* s *is the speed of the infection when it spreads at rate* 1*,* β *is a constant, and* $\eta_1 t_1$ *scales to infinity. Suppose* $\alpha = o(\beta(1+\gamma_1 t_1)^{-1})$*, and* $\log(\beta/\alpha) = o(\eta_1 t_1)$*. Then for both forms of the Multiple Ball Algorithm, the Type II error probability tends to* 0*.*

Proof outline. Using the assumptions, we show that additional seeds are unlikely. In fact, β/α is larger than the number of seeds. Therefore, the largest β epidemics contain at least an α fraction of the infected nodes. Using the speed bound, the threshold is sufficiently large enough to contain each epidemic. Therefore, the Multiple Ball Algorithm succeeds. A similar approach works for the Scaling Multiple Ball Algorithm. The proof details are in the appendix. □

Finally, recall we can choose the algorithm parameters α, β and m. Then the question is, when can we choose appropriate algorithm parameters so that the probability of error goes to 0? This is answered by the following theorem.

Theorem 5.3.3. *Suppose there exists C_3 such that $\eta_0 t_0 = o\left((\gamma_0 t_0)^{-1/(C_3+1)}\right)$ and $q\gamma_0 t = \omega(\log n)$. Suppose $\eta_1 t_1 = \omega(\log(\gamma_1 t_1))$, $\gamma_1 t_1 = \omega(1)$, and $s\eta_1 t_1 = o\left(b(\frac{n}{\gamma_1 t_1})\right)$. Then the algorithm parameters can be chosen so that the probability of error for the Multiple Ball Algorithm approaches to* 0.

Proof. We must choose m, α and β so that $s\eta_1 t_1 < m < b\left(\frac{\alpha n}{C_4 \beta(1+\epsilon)}\right) - C_3$ and $\alpha = o(\beta(\gamma_1 t_1)^{-1})$, $\log(\beta/\alpha) = o(\eta_1 t_1)$, where $C_4 = \bar{d}^{C_3+1}$. We set $\beta = 1$, though note that we can use $\beta > 1$ by inversely scaling α with β. First we consider the conditions on α. Define an arbitrary slowly increasing function $g(n) = \theta(1)$, $g(n) = o(\gamma_1 t_1)$. This is possible since $\eta_1 t_1 = \omega(1)$. Choose $\alpha = (\gamma_1 t_1 g(n))^{-1}$. Then we have

$$\begin{aligned}
\log(1/\alpha) &= \log(\gamma_1 t_1 g(n)) \\
&< 2\log(\gamma_1 t_1) \\
&= o(\eta_1 t_1).
\end{aligned}$$

Thus α satisfies the desired conditions. Now we show it is possible to choose an appropriate m. By hypothesis, $s\eta_1 t_1 = o(b(\frac{n}{\gamma_1 t_1}))$. From our choice of α, for sufficiently large n, $\frac{\alpha}{C_4(1+\epsilon)} < \frac{1}{\gamma_1 t_1}$. Using the concavity of $b(x)$,

$$\begin{aligned} b\left(\frac{n}{\gamma_1 t_1}\right) &< \frac{\gamma_1 t_1}{\alpha/(C_4(1+\epsilon))} b\left(\frac{\alpha n}{C_4(1+\epsilon)}\right) \\ &= o\left(b\left(\frac{\alpha n}{C_4(1+\epsilon)}\right)\right). \end{aligned} \tag{5.1}$$

Therefore, $s\eta_1 t_1 = o(b(\frac{\alpha n}{C_4(1+\epsilon)}))$, with $s\eta_1 t_1 = \omega(1)$ by hypothesis. Thus it is clear m can be chosen with $s\eta_1 t_1 < m < b\left(\frac{\alpha n}{C_4(1+\epsilon)}\right) - C_3$, for example by averaging each side. With this choice of parameters, the conditions of Theorem 5.3.1 and Theorem 5.3.2 are satisfied. Hence, both the Type I and Type II error probabilities tend to 0. □

The above conditions are fairly opaque however. These conditions can be described roughly as follows:

- The total number of nodes that can be covered by a β balls of radius $2m$ (where m increases with n) must scale a constant factor less than the total number of nodes times α.
- In Process SR-WE [Process 0], the expected number of reporting seeds must be order-wise more than $\log n$.
- In Process SR-WE, the infection spreads no more than a constant distance.
- For Process WR-SE [Process 1], the threshold m must be set large enough that a ball of radius m covers the largest infection (using the epidemic speed).

- For Process WR-SE, the expected number of seeds must be order-wise less than $\beta\alpha^{-1}$.
- For Process WR-SE, $\beta\alpha^{-1}$ must be order-wise less than exponentials in $\eta_1 t_1$.

One interesting observation is that even when there are multiple clusters, it is still possible to use only a single ball in our algorithm, as long as α is reduced appropriately. This fact is borne out in our simulations.

5.4 Simulations

To support our analytic results, we performed a variety of simulations of the performance of our algorithms. Each of our simulations was performed on a grid with $n = 4900$, and with the opposing edges connected to form a torus. We use an infection time of $t = 10$ (unless otherwise stated) and a reporting probability of $q = 0.25$. The average probability of our algorithm was determined over 10000 trials, where each infection process was equally likely to be the causative infection.

To normalize the infection sizes, we adjusted the rates so that the infection sizes for both infection processes would be similar. This was done by first choosing the epidemic rate for each process, and then empirically finding the random rate to three significant digits so that expected number of infected nodes hit a target value, typically with a specified fraction of the network being infected. This was done so that all the infections (for the various parameters) would be fairly comparable. Process SR-WE [Process 0] used an epidemic rate of 0.2, and we varied the epidemic rate for Process WR-SE [Process 1].

Figure 5.1 shows the probability of error using the Multiple Ball Algorithm with a single ball ($\beta = 1$) for various infection sizes. The infection rate for Process WR-SE [Process 1] is given on the x-axis. As expected, the larger the infection, the more difficult it is to use clustering to determine whether an infection is mostly random or mostly an epidemic. When an expected 60% of the nodes in the network are infected, then the probability of error stays high, even for much larger infection rates. Note that there is a maximum infection rate before the target infection size is exceeded regardless of the random sickness rate. We used Process WR-SE infection rates close to that maximum.

Next we determine the effect of α on the probability of error. Again, $\beta = 1$. These results are shown in Figure 5.2. Surprisingly, changing α has a relatively small effect on the probability of error. The largest effect seen is using too large a value for larger Process WR-SE infection rates (when the probability of error is low). However, that is still relatively small. Then our algorithm seems fairly insensitive to the value of α.

Finally, we use the Scaling Multiple Ball Algorithm with multiple balls, implemented as a greedy algorithm. The probability of error is plotted in Figure 5.3. This simulation used a larger grid graph with $n = 10000$ with 1000 trials. Process SR-WE [Process 0] had an infection rate of 0.1 and random rate so that the expected infection size was 20% of the graph. We set the parameter $\alpha = 1$ for $\beta = 1$ and $\alpha = 0.75$ otherwise, which was empirically the

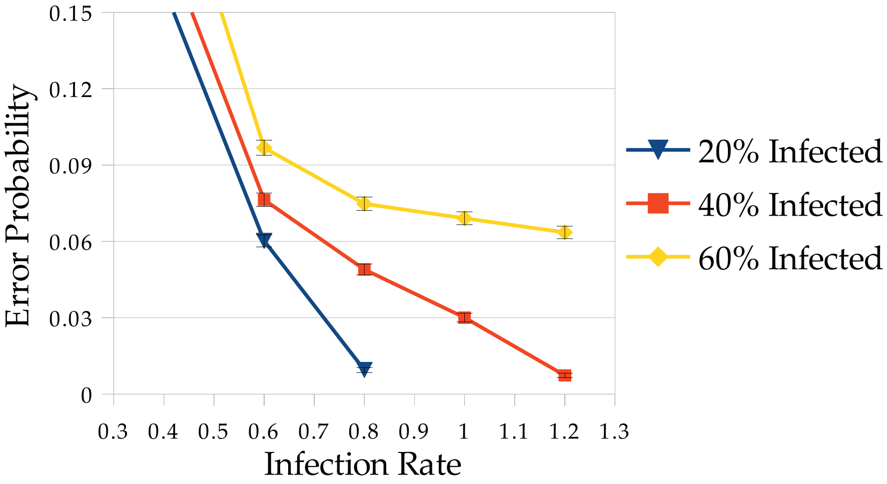


Figure 5.1: This figure shows a chart of the overall error probability for various expected fraction infected against the Process WR-SE infection rates. The Process SR-WE infection rate is 0.2. The parameter $\alpha = 0.5$. The ball radius m was set to the optimal value empirically.

optimum value of α for each β from several tested values. The x-axis shows the infection rate for Process WR-SE [Process 1]. From the simulation, we find using multiple balls achieves a reduction in error probability for $\beta \geq 5$, especially at lower infection rates of Process WR-SE, when the problem is more difficult. However, this reduction does come at a cost of computation time.

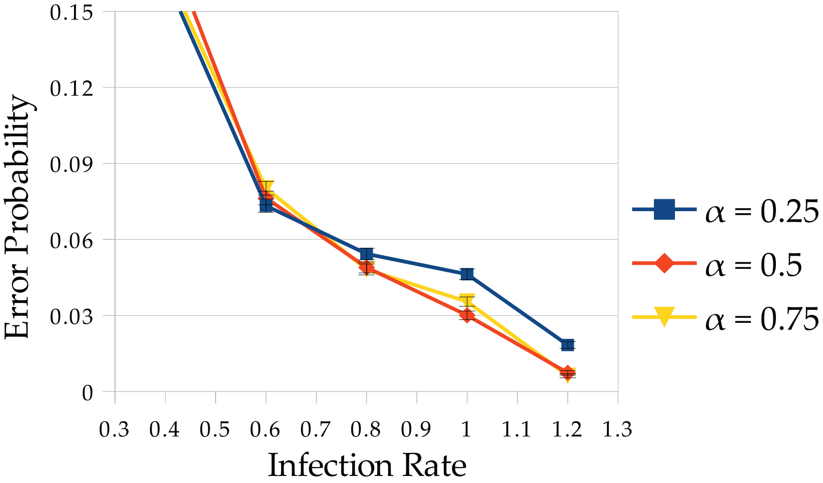


Figure 5.2: This figure shows the overall error probability for multiple values of α and Process WR-SE infection rates. The Process SR-WE infection rate is 0.2 and the expected fraction infected was 40%. The parameter m was set to the optimum value.

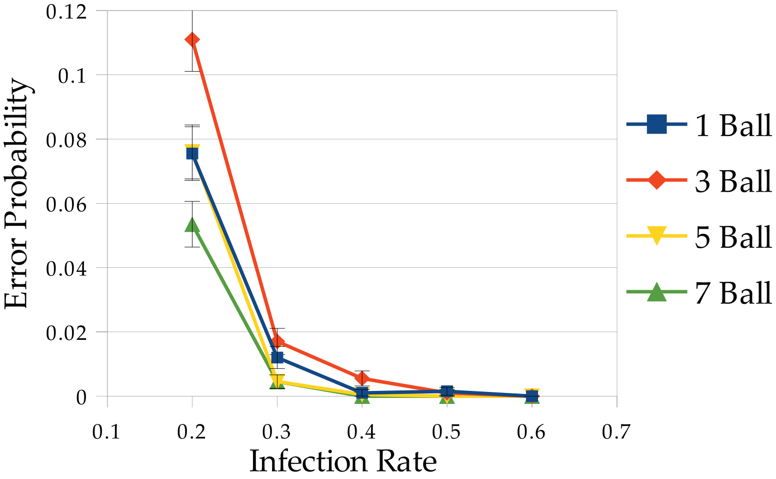


Figure 5.3: This figure shows the error probability for the Scaling Multiple Ball Algorithm for a range of β (ball count). The simulation used a grid graph with $n = 10000$, 1000 trials, a Process SR-WE infection rate of 0.1 with $t = 10$. The random sickness rate was set so the expected fraction infected was 20%. The parameters α and m were set to the optimum value from a set of predetermined values.

Chapter 6

Unknown Edges

6.1 Introduction

Modern life is dominated by communicated across networks, from over the Internet, phones, or through traditional contact networks. Virus, information, and rumors travel on these networks as well, and identifying when these infectious processes spread is valuable in many cases. Using data about a subset of infected people/devices (attained using infection reports, polling, etc.) and the associated network, it is possible to distinguish an infection from simple random noise, modeled as a random sickness [46]. Though sparse knowledge about infected users is often fairly straightforward to obtain, it may be different to know the entire social network. We pose the question: if the network on only known inexactly, is it still possible to determine when an infection occurs?

For many online networks, the information on the entire social network (formed by friends, followers, and equivalent relationship) is available. Yet even in these cases, this network may not entirely represent the network over which an infection spreads. For instance, there may be friends who do not use that social networking service. In the case of the spread of information, communicating to these offline friends causes the appearance of that information 'jumping' across the network. In the worst case, the infection may spread to someone at a far distance on the social network, causing the infection to not appear clustered as expected. Algorithms to distinguish between random sicknesses and epidemics must be robust against these jumps.

To solve this problem, we use the Multiple Ball Algorithm. We consider this problem in both the cases of unknown short and unknown long edges. For unknown short edges, the distances on the graph do not change substantially by their removal. We demonstrate that our algorithm can tolerate an arbitrarily large number of unknown short edges and still perform well. In the case of long edges, which may substantially change the network topology, we show that the algorithm succeeds when there are up to a constant number of these. These analytic results are supported by simulations.

6.2 Model

We base our model on that presented in Chapter 2. Let G from graph family $\mathcal{G}$ be the complete true social network on which an epidemic may spread as a SI infection process. We are presented with a set of sick nodes. These are either from a random sickness, or from the aforementioned epidemic. Only a small fraction of these infected nodes report their infection, each with probability q. The sizes of each infection process are normalized to be equal.

We require constraints on G so that there is sufficient topological information to distinguish the two processes. In particular, we require the graph to have *limited epidemic speed* and *limited neighborhood size*, as defined Section 4.2.1. As shown in Theorem 4.2.1, these conditions are satisfied by all bounded degree graphs.

6.2.1 Missing Edges

In this case however, some edges in G are unknown. These unknown edges may be chosen arbitrarily, or under constraints detailed in the following sections. Define $\bar{G}$ as the subgraph of G with these edges removed. That is, $\bar{G}$ is the known social network. Note equivalently we can start with $\bar{G}$ and add these unknown edges to form the social network G, which may be useful if the known edges should have some structure. The algorithm has knowledge of only the subgraph $\bar{G}$ (and the associated speed and spread functions). With only this limited knowledge, the task is to distinguish an epidemic spreading on G from a random sickness.

We say a set of unknown edges $\bar{E}$ is has maximum length ℓ if, for each $e = (u, v) \in \bar{E}$, $\text{dist}_{\bar{G}}(u, v) \leq \ell$. That is, removing the edges increases the distance between any two previously connected edges to at most ℓ.

6.3 Results

To solve this problem, we use the Multiple Ball Algorithm as given in Section 5.2.3. In this algorithm, we attempt to contain an α fraction of the infected nodes with β balls of radius m, where α, β, and m are algorithm parameters.

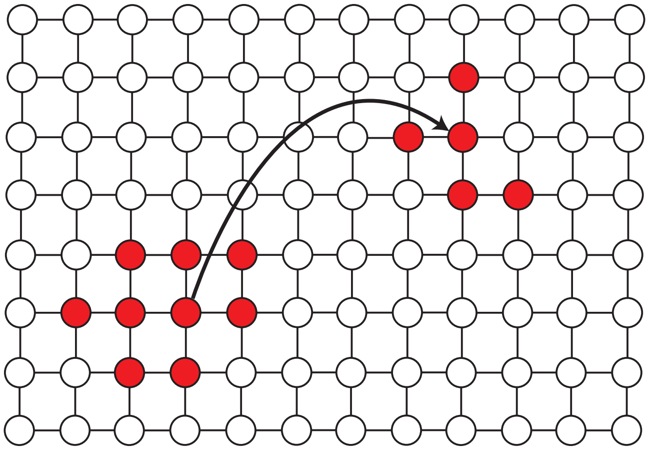


Figure 6.1: A grid with one long unknown edge, represented by the thick arrow. The infected nodes are colored red. The infection appears to jump across the graph when the long edge is not known.

We divide the problem into two cases. In the first case, all the unknown edges are 'short'. The known graph then closely resembles the true social network, just with some errors. The epidemic will still result in a clustering of reporting infected nodes, and therefore can still be identified. In fact, we show that for reasonable infection sizes, we can tolerate an arbitrarily large number of such missing edges. The second cases is when some edges are 'long'. These edges allow the epidemic to jump large distances, and may cause the set of infected nodes to appear as many different clusters. Figure 6.1 shows an example. We show our algorithm can tolerate a constant number of such edges.

6.3.1 Short Edges

Suppose the set of unknown edges has maximum length ℓ, where ℓ is a constant. Because of this, the distance between any two nodes cannot increase by more than a factor of ℓ. Note that for all nodes u and radius r, $\mathrm{Ball}_G(u, r/\ell) \subseteq \mathrm{Ball}_{\bar{G}}(u, r) \subseteq \mathrm{Ball}_G(u, r)$. Due to this fact, the known graph $\bar{G}$ satisfies the spread constraint for $b_{\bar{G}}(x) \triangleq b_G(x)$. In addition, if $\bar{G}$ satisfies the spread constraint for some function $\tilde{b}_{\bar{G}}(x)$, the original graph G satisfies that constraint with function $\tilde{b}_G(x) \triangleq \tilde{b}_{\bar{G}}(x)/\ell$. Likewise, since the epidemic would travel slower on $\bar{G}$, $\bar{G}$ satisfies the speed constraint. In order to set the threshold, we also require that the speed of the infection does not change substantially. A large speed change is possible even with short edges if they are in sufficient quantity. Formally, we suppose that for a known constant $\kappa \geq 1$, if speed condition applies to $\bar{G}$ with speed $\bar{s}$, it applies to G with speed $\kappa\bar{s}$. If the speed of an epidemic on the original graph is known (for example, from prior epidemics), then this condition is not necessary for the threshold to be calculable.

Theorem 6.3.1. *Suppose G is the true social network and satisfies the speed and spread constraints. Let $\bar{G}$ be the known subgraph of G with edges with length at most c removed, and suppose it has speed $\bar{s}$ and spread function $b_{\bar{G}}(x)$. Suppose t increases with n sufficiently that the number of reporting nodes is $\Omega(\log n)$, and that for some constant $\epsilon > 0$, $t < b_{\bar{G}}(n/(1+\epsilon))/(\kappa s_{\bar{G}}\ell)$. Under these conditions, the Multiple Ball Algorithm using $m = \kappa s_{\bar{G}}\ell t$ (and $\alpha = 1$, $\beta = 1$) identifies the type of infection correctly with probability tending* *to* 1 *with the number of nodes, n.*

Proof. Suppose first the sickness was caused by an epidemic. Note that by the speed condition on G, the infection can be contained with high probability inside a ball of radius $s_G t$ (where the ball in on G), and $s_G \leq \kappa s_{\bar{G}}$. Since the distance between any two nodes can increase by a factor of at most ℓ, the infection is contained in a ball on $\bar{G}$ with radius $\kappa s_{\bar{G}} \ell t$. Therefore, the Type II error probability decays to 0.

Now consider a random sickness. Any ball on $\bar{G}$ with radius m can contain no more nodes than the same ball (with the same center and radius) on G since removing edges only increases the distance between nodes. By hypothesis, $m < b(n/(1+\epsilon))$ for some ϵ. Hence from the spread condition, each ball contains no more than a $1/(1+\epsilon)$ fraction of the network with high probability. From standard arguments as in our previous results, the probability that a random set of at least $\log n$ nodes is entirely contained in only a fraction of the network decays to 0 exponentially. From a union bound over the n possible balls, we find that the Type I error probability also decays to 0. □

6.3.2 Long Edges

As before, let G be the correct graph and suppose it satisfies the speed and spread constraints. A constant number, K, of these edges are unknown by the algorithm. These unknown edges are chosen arbitrarily and may be any length. Then $\bar{G}$ is the subgraph of G known by the algorithm, differing only by these K edges. Note that as in the case of short unknown edges, $\bar{G}$ satisfies the speed and spread conditions.

To handle the possible multiple clusters, we use the Multiple Ball Algorithm with $\beta > 1$. This algorithm can successfully determine whether the causative process is an epidemic or not in this problem for a simple reason. Since there are at most K jumps (across the K missing edges), there are at most $K + 1$ separate infections. Therefore, the infection can be contained in balls around each of these separate infections (for sufficiently small infections). This intuition is proven in the following theorem.

Theorem 6.3.2. *Let $\bar{G}$ is the known subgraph of G, that is, with the unknown edges removed. Suppose that the number of unknown edges is at most K. Define constants $C_1 = [3(K+2)]^{-1/2}$ and $C_2 = C_1\alpha/(K+1)$. Suppose t scales such that the number of reporting nodes is $\Omega(\log n)$ and $t < b_{\bar{G}}(C_2 n)/s_{\bar{G}}$. Using the Multiple Ball Algorithm with α be an constant, $0 < \alpha \leq 1$, $\beta = K+1$, and $m = s_{\bar{G}}t$, the infection type can be determined with probability tending to 1 as the graph size increases.*

Proof. First we show that the Type II error probability decays to 0 as $n \to \infty$. To do this, we define a set of nodes a_0, a_1, $\ldots$, a_K as follows. Set a_0 to the initial infected node. The first time the infection traverses one of the unknown edges, let a_1 be newly infected node. Likewise, let a_2 be the infected node from the second time the infection traverses an unknown edge, and so on. Since there are only K edges, and an infection can spread across an edge at most once, there are at most K such infected nodes from 'jumps', which we call 'seeds'. Any remaining undefined nodes are set arbitrarily. This is well defined since two nodes cannot be infected at the same time almost surely.

Now consider the epidemic process where each of these seeds is infected at time 0 and the infection spreads over $\bar{G}$. Note that this behaves the same as the original process except the 'seeds' are infected at an earlier time. The removal of the unknown edges does not reduce the spread of the infection since the end points are already infected. Since the spread of the infection is monotonic in time, the infection is only larger on this new process.

As mentioned, since removing edges can only reduce the speed of the infection, we know $\bar{G}$ satisfies the speed condition as well. Therefore, the spread of the infection (ignoring missing edges) around node a_i for any $0 \leq i \leq K$ is at most $m = s_{\bar{G}}t$ with probability tending to 1. Hence by a union bound, the entire set of infected nodes is contained by $K+1$ balls, one around each seed of radius m, with probability tending to 1. The spread on this new process is only larger, so the same property applies to the actual epidemic. Therefore, the Multiple Ball Algorithm correctly labels it an epidemic with probability tending to 1.

Now, consider the Type I error probability. From the spread condition and since $m < b_{\bar{G}}(C_2 n)$, each ball on $\bar{G}$ can contain at most $C_2 n$ nodes, and hence, all collections contain less than $\beta C_2 n = C_1 \alpha n$ nodes. Consider any set of such balls, and let B be the nodes their union contains. Recall S_{rep} is the set of reporting sick nodes (located randomly). From standard balls-in-bins arguments as in Lemma 4.2.2,

$$\begin{aligned} Pr(|B \cap S_{\text{rep}}| \geq \alpha \, |S_{\text{rep}})| &\leq e^{\frac{-|S_{\text{rep}}|}{3C_1^2}} \\ &\leq n^{-(1/C_1)^2/3} \\ &= n^{-(K+2)} \end{aligned}$$

using the fact that $|B| \leq C_1 \alpha n$ and $|S_{\text{rep}}| \geq \log n$. There are no more than n^β such collections of balls. From a union bound over all collections B,

$$\begin{aligned} Pr(\exists B : |B \cap S_{\text{rep}}| \geq \alpha \, |S_{\text{rep}}|) &\leq n^{\beta-(K+2)} \\ &= n^{-1}. \end{aligned}$$

Therefore, the probability that an α fraction of the sick nodes can be contained by β balls of radius m, that is, the Type I error probability, decays to 0. □

6.4 Simulation

For our simulations, we assumed there were a small number of unknown, arbitrarily long edges. We started with a grid graph with $n = 4900$, and added a fixed number, K, of edges. Each edge connected two nodes, both chosen independently uniformly at random. Each of these additional edges were unknown by the algorithm (so only the grid graph was known). The reporting probability is $q = 0.25$. The Scaling Multiple Ball Algorithm was applied to this problem with $\alpha = 1$, $\beta = K + 1$, and the threshold m set to the optimum value. This algorithm was implemented in the more efficient, but inaccurate, greedy form as described in Section 5.2.3. We expect that the

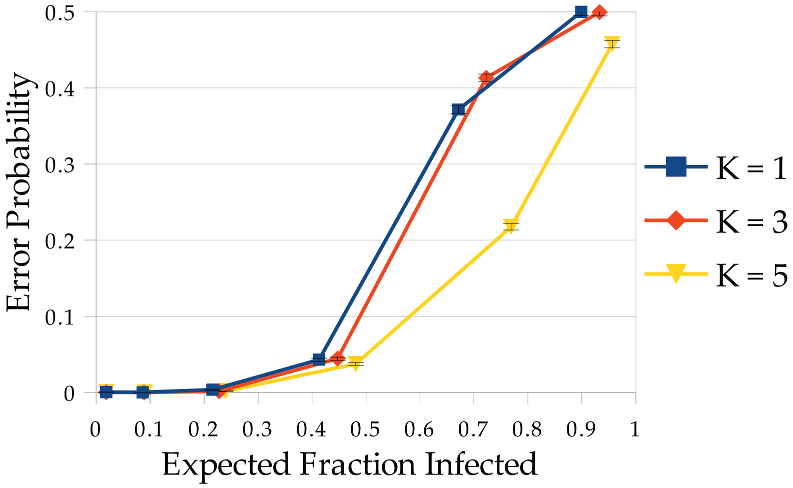


Figure 6.2: This figure shows the overall error probability for a grid graph with K additional randomly located unknown edges. The empirical expected fraction of nodes that are infected is shown on the x-axis. The parameters are $\alpha = 1$, $\beta = K + 1$, and m set optimally.

inaccuracies have no significant impact on the probability of error. The random sickness was normalized to have the same expected size as the epidemic. The overall probability of error was determined using 10000 trials, with equal number of epidemics and random sicknesses. This error probability is plotted against a range of expected infection sizes (as determined empirically) in Figure 6.2.

In all cases, the error probability is low until most of the graph is infected. Then, we see that empirically, the Scaling Quantile Ball Algorithm performs well in the presence of unknown edges.

Chapter 7

Weighted Graphs

7.1 Introduction

Detecting failures and infections spreading over a network requires being able to distinguish a phenomenon that is spreading from node to node through a contact process, from a collection of random failures occurring by chance, or perhaps driven by an external source or event. The importance of correct diagnosis of a spreading phenomenon – i.e., understanding that that there is indeed a spreading epidemic, and properly detecting the contact network over which it spreads – has been well documented in the history of human virus epidemiology and computer networks alike [12, 42, 57].

A key assumption in prior work is homogeneity of the spreading network; that is, the epidemic is assumed to spread at a constant (probabilistic) rate. In real world networks, both these key assumptions typically do not hold. For starters, close relations transmit infection more readily than distant connections. More troubling is the assumption that the contact network is

The work in this chapter is to appear in the following publication:
Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. Local detection of infections in heterogeneous networks. In *Proceedings of INFOCOM, IEEE*, 2015. (To appear).

known. While some network connections may be known (e.g., nuclear family), others can only be estimated and should be best modeled by probabilistic connections of different strength, especially from publicly available data. For example, publicly (or relatively easily) available data may include a list of coworkers, but typically would not include statistics on pairwise daily interaction times among employees. While a model assuming a known uniform weight among all coworkers equaling the edges among family members may well be inaccurate, one that assigns weighted edges that capture whatever partial knowledge may be available, can be significantly more accurate and representative.

Any realistic modeling of real-world epidemics must, therefore, be able to accommodate heterogeneous edges. This is precisely the topic of the present paper. Given a snapshot of an epidemic on a non-homogeneous graph, our objective is to correctly diagnose the existence of the epidemic, especially when parts of the network are not known, and when the data themselves are highly noisy, corrupted via high levels of false positives and false negatives.

7.2 Problem Statement

We use a similar model as in previous chapters. The infection is caused either by a random sickness or an epidemic. At a single time t, each infected node reports with probability q. We suppose that the infection processes are normalized so the expected infection size is the same for both processes. Using the set of reporting nodes and knowledge of the graph structure, we seek to determine whether the infection was caused by an epidemic or simply a random sickness. The set S denotes the complete set of infected nodes at time t, and S_{rep} denotes the set of reporting nodes.

7.2.1 Weighted Infection Model

The variant we consider is when the graph for the infection is weighted. Let $G = (V, E)$ be the infection graph, with nodes V and edges E. For each edge e_{ij} between nodes i and j, there is a weight $w_{ij} > 0$. The infection spreads over this graph in a manner similar to a standard SI infection. Initially, a single randomly chosen node is the infection source, say node i. For each edge connected from this node to an adjacent susceptible node, say node j, a clock is started with exponentially distributed duration with mean $1/w_{ij}$. When a clock expires, the susceptible node on that edge becomes infected (if it is not already infected by a different source). When a node becomes infected, clocks are started for each edge connected to that node in that same way, with the expected duration of the clock determined by the edge weight. The infection spreads between connected nodes in this way until time t has passed. Therefore, the higher the weight is between two nodes, the faster the infection will travel between them. At this time, the infected nodes report independently with probability q. In this case, we constrain $q = \omega(1/\log n)$.

These weights can substantially change how the infection spreads across the graph. Edges with very low weights can almost be ignored. The infection will spread mostly on edges with higher weights. Then the effective topology of the graph may be closer to the topology due to the higher weight edges. A challenge with weighted graphs is that it can be difficult to evaluate the infection structure, such as determining the expected time to infect a particular node, without excessive computation.

7.2.2 Graphs

For graph G and arbitrary nodes i and j, define $\operatorname{len}(i, j)$ as the length, in hop count, between i and j. Similarly, define $\operatorname{dist}(i, j)$ as the minimum *weighted distance* between i and j. Our algorithm considers "balls" on these graphs to be all nodes within a certain distance (this distance is weighted) from a central node. For graph G, node i and radius r, define $\operatorname{Ball}(G, i, r) = \{j \in V : \operatorname{dist}(i, j) < r\}$.

We suppose the graphs satisfy two conditions, and call such graphs *acceptable graphs.* These conditions are similar to the speed and spread conditions of previous chapters, though we include lower bounds. As before, graphs with bounded maximum degree satisfy these conditions. The bounded speed condition states roughly that the infection spreads at a bounded speed.

Definition 7.2.1. Consider graph family $\mathcal{G}$. This family satisfies the *bounded speed condition* for minimum speed $s_{(-)}$ and maximum speed $s_{(+)}$ (both constants) if, for infection time t increases with n without bound, for graph G, infection S and infection source i,

$$P\left(\operatorname{Ball}(G, i, s_{(-)}t) \subseteq S \subseteq \operatorname{Ball}(G, i, s_{(+)}t)\right) \to 1.$$

That is, the infection spreads at least a distance $s_{(-)}t$ and at most a distance $s_{(+)}t$.

The bounded spread condition requires that the neighborhood sizes are well behaved. The spreading functions may scale with n, but we are most interested in graphs of bounded degree, in which case these functions vary only with x. Most importantly, we want to constrain the neighborhood size as the graph size increases.

Definition 7.2.2. A graph family $\mathcal{G}$ satisfies the *bounded spread condition* with concave increasing spreading functions $b_{(-)}(x)$ and $b_{(+)}(x)$, $1 \leq x$ if, for graph $G^{(n)}$ drawn from this family, with probability tending to 1 the following holds for each node i and number of nodes $x < n$:

$$\left|\text{Ball}(G, i, b_{(-)}(x))\right| < x < \left|\text{Ball}(G, i, b_{(+)}(x))\right| .$$

7.2.3 Additional Constraints

In addition to the basic problem, we consider two additional variants. First, there may be false positives, uninfected nodes that report a sickness regardless. Second, there may be unknown edges of the graph. Though both of these constraints have been considered previously, we now consider them in the context of weighted graphs.

We use the same false positive model as in Chapter 4. The number of false positives is set by fixing the ratio between the number of reporting infected nodes and the number of false positives. For a constant $f \geq 0$ and

$|S_{\text{rep}}|$ truly reporting nodes, we set the number of false positives to be (approximately) $f\,|S_{\text{rep}}|$. For each of the $\lfloor f\,|S_{\text{rep}}|\rfloor$ false positives, we independently choose a random node from the entire graph and that node reports an infection, where repeats are allowed.

The second constraint is the some edges of the graph are not known. From this perspective, the infection may appear to 'jump' between two nodes. We consider the same cases as in Chapter 6. We define the length of a missing edge as follows. For a removed edge e connecting nodes i and j, we say the length of e is $\text{dist}_{\bar{G}}(i, j)$, the weighted distance between i and j on the graph with missing (unknown) edges. For a constant ℓ, removed edges are considered short if their length is at most ℓ. Otherwise, they are called long edges.

7.2.4 Algorithm

Our approach to solving this problem involves characterizing the shape of an infection. The distance between two nodes appears to be a good approximation of how easily an epidemic can spread from one node to the other. The shorter the (weighted) distance, the faster the infection spreads. However, this ignores the topological considerations: the number of short paths also matters. Nevertheless, we show that the distance measure is sufficient to approximate the shape of an epidemic in this situation, and thereby distinguish an epidemic from a random sickness.

The heterogeneity in edges fundamentally changes the way we need to think about inference in this setting. From an algorithmic viewpoint, earlier work that addressed this kind of inference problem [46, 47, 48] did so by essentially detecting the boundary of the infected region – in essence, they compare the radius of a ball that 'covers' the reporting infected nodes to a fixed threshold. We call this algorithm the Threshold Ball Algorithm. If the radius is small, then they report that there is an epidemic. However such a test is sub-optimal, both analytically as well as in simulations when the network edges have heterogeneity. Analytically, this occurs because estimates of the radius of a ball covering the infected nodes does not have sufficient probabilistic concentration guarantees for our inference purposes. Intuitively, this happens because with edge non-homogeneities, the 'boundary' of infection can have large protuberances (think of ray-like objects flaring out of the ball-like footprint of infected nodes). These can cause outer radius estimates to be poor. However, taking a volume inside the infected region and estimating infection *densities* turns out to be much more robust. See Figure 7.1 for an example.

The Threshold Ball Algorithm performs especially poorly if it uses hop count to measure the ball radius. For some graph topologies, such as a grid with diagonal edges, it is possible that the outer ball around the epidemic covers the entire graph, even when the epidemic is relatively small. Figure 7.2 shows an example of this phenomenon. In that example, the radius of the ball necessary to surround the infection is equal to the radius of the entire graph, even if the epidemic is fairly small. Therefore, the Threshold Ball Algorithm cannot distinguish this epidemic from a random sickness in this case, even if

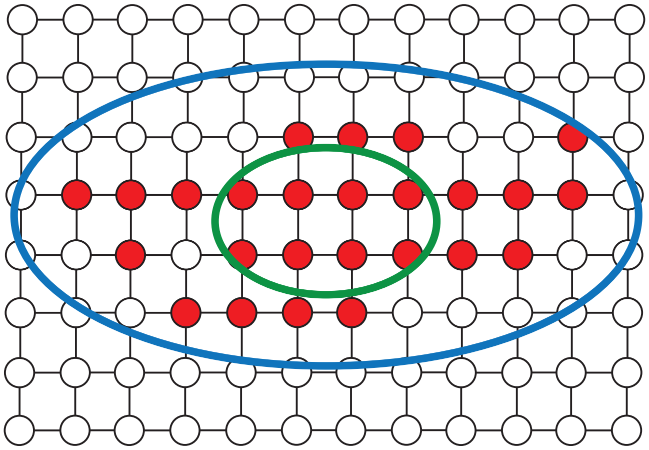


Figure 7.1: A weighted grid with infected nodes colored red, where the infection travels faster in the horizontal direction. Due to the weights, the ball surrounding the infected nodes (blue) is excessively large compared to the more robust internal ball (green).

all nodes report. However, the inner ball approach we use still succeeds.

Our algorithm is called the Ball Density Algorithm. The algorithm takes parameters m and d. The algorithm searches through the graph, and determines whether any ball of radius m has a density of reporting nodes at least d. As before, a ball of radius m is defined as all nodes within some distance m of some central node. If there is a ball with sufficient density, the reporting nodes appear sufficiently clustered and the infection is labeled an 'epidemic.' Otherwise, it is labeled a 'random sickness.'

The Ball Density Algorithm is similar to the scan statistic considered by Arias-Castro et al. [1, 2]. In that work, each node v reports a standard Gaussian X_v except for possibly in a cluster $K \in \mathcal{K}_m$, which report i.i.d.

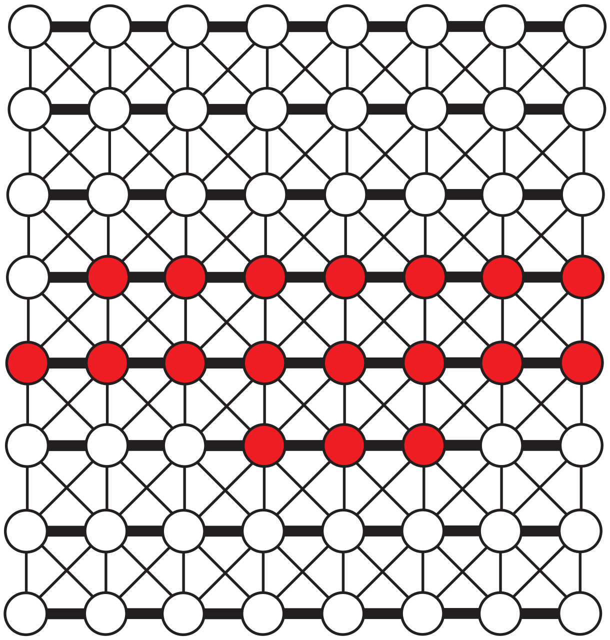


Figure 7.2: An epidemic on a weighted grid including diagonals with infected nodes colored red. The infection travels faster in the horizontal direction, denoted by the thicker lines. The Ball Density Algorithm will likely succeed for this infection, but the Threshold Ball Algorithm will not.

Gaussians with positive mean μ_K. The scan statistic is defined as

$$\max_{K \in \mathcal{K}_m} \frac{1}{\sqrt{|K|}} \sum_{v \in K} X_v.$$

If this statistic is above a threshold, then there is likely an anomalous cluster. For our problem, this cluster would correspond to the epidemic. Unlike in our case however, all possible clusters are known. Also, in [1], they set $\mu_K = |K|^{-1/2}\Gamma_K$ and analyze necessary and sufficient bounds on Γ_K. That is, the mean deviation of the nodes in that cluster (μ_K) may decay on the order of the square root of number of nodes in that cluster. Because of this, it is better to divide by $\sqrt{|K|}$ (as opposed to $|K|$) in the scan statistic. However, in our case, the reporting probability is always q regardless of the infection size, so we must divide by the ball size to find the mean reporting rate (the density). In addition, computing the scan statistic may be computationally intensive if $\mathcal{K}_m$ is large, even when restricted to an ϵ-net. The Ball Density Algorithm on the other hand is always efficient.

Ideally, we want the density threshold d to be close to the expected density in the infected set, q. However, q may not be known. In that case, we can estimate the required density by comparing it to the density outside the ball. If the infection density within the ball is sufficiently higher than the density outside the ball, that ball is likely within an epidemic. Along these lines, we use a modified form of the algorithm called the Relative Ball Density Algorithm. In this algorithm, if the density within a ball of radius m exceeds the density outside the ball by a factor of at least $\beta > 1$, we label the infection an 'epidemic.'

Algorithm 8 Ball Density Algorithm

```
Input: Graph $G$; Set of reporting infected nodes $S_{\text{rep}}$;
Parameters: Density $d$, Radius $m$
Output: EPIDEMIC or RANDOM

  for all $i \in V$ do
    if $|\text{Ball}(G, i, m) \cap S_{\text{rep}}| \,/\, |\text{Ball}(G, i, m)| \geq d$ then
      return EPIDEMIC
    end if
  end for
  return RANDOM
```

Algorithm 9 Relative Ball Density Algorithm

```
Input: Graph $G$; Set of reporting infected nodes $S_{\text{rep}}$;
Parameters: Relative Ratio $\beta$, Radius $m$
Output: EPIDEMIC or RANDOM

for all $i \in V$ do
    ExternalBall $\leftarrow V \setminus \text{Ball}(G, i, m)$
    $d \leftarrow \beta\, |\text{ExternalBall} \cap S_{\text{rep}}| \,/\, |\text{ExternalBall}|$
    if $|\text{Ball}(G, i, m) \cap S_{\text{rep}}| \,/\, |\text{Ball}(G, i, m)| \geq d$ then
        return EPIDEMIC
    end if
end for
return RANDOM
```

7.3 Results

We show that the Ball Density Algorithm can distinguish between a random sickness and an epidemic on a weighted graph under the specified conditions. In addition, we also show that this algorithm is reliable. The algorithm still succeeds even if there are false positives or some edges of the graph are not known.

7.3.1 Basic Problem

The fundamental case is when we have access to the entire graph G and the reporting nodes S_{rep} with no false positives. Later sections include the case when there are false positives, and when some graph edges are unknown. We demonstrate that the Ball Density Algorithm and Relative Ball Density Algorithm can succeed in determining the type of infection with asymptotic probability 1, and characterize the range of infection sizes for which this is possible.

Our results require the fact that the number of reporting nodes in a set is highly clustered around its expectation. This follows from the following well-known Chernoff bound:

Lemma 7.3.1. *Suppose in a set U of nodes, each node reports an infection independently with probability q. Let U_{rep} be the set of reporting nodes inside U. Then for any $\delta > 0$,*

$$P\left(|U_{\text{rep}}| \geq (1+\delta)q\,|U|\right) < \exp(-\delta^2 q\,|U|/3)$$

and

$$P\left(|U_{\text{rep}}| \leq (1-\delta)q\,|U|\right) < \exp(-\delta^2 q\,|U|/2).$$

We begin by limiting the density of a random sickness and of an epidemic. We use the fact that, when all balls of a specified radius contain at least $\log^2 n$ nodes, every such ball has density close to its expectation. Roughly speaking, the following two theorems provide the conditions for the Type I and Type II error probabilities to tend to 0.

Theorem 7.3.2. *Consider an acceptable graph G of size n with random sickness S_{rep}. Let $\epsilon > 0$ be a small constant. Consider ball radius m satisfying $b_{(+)}(\log^2 n) < m$ and density threshold $d = (1-\epsilon)q$. If the expected number of infected nodes is less than $(1-2\epsilon)n$, the density of every ball of radius m is less than d with probability tending to 1.*

Proof. Note that a ball of radius m contains at least $\log^2 n$ nodes. By hypothesis, the expected reporting node density over the entire network is less than $(1-2\epsilon)q$. Therefore, for any collection of nodes, the expected density of infected nodes in that region is less than $(1-2\epsilon)q$. Let $\delta = (1-\epsilon)/(1-2\epsilon)-1$. From the Chernoff bound Lemma 7.3.1, for a set of nodes of size k, the probability the density of reporting nodes in the set is over $(1+\delta)(1-2\epsilon)q = (1-\epsilon)q$ is less than $\exp(-\delta^2(1-2\epsilon)qk/3)$. Hence, for $k \geq \log^2 n$ (that is, for balls of radius m), and with $q = \omega(1/\log n)$, this probability decays to 0 faster than $1/n$. Using a union bound over the n balls of radius m (one for each central node), each with at least $\log^2 n$ nodes by the condition on m, we find that all of them contain density less than $(1-\epsilon)q$ with probability tending to 1. □

Theorem 7.3.3. *Consider an acceptable graph G of size n with reporting infected set S_{rep} from an epidemic. Let $\epsilon > 0$ be a small constant. For time $t > b_{(+)}(\log^2 n)/s_{(-)}$, ball radius $b_{(+)}(\log^2 n) < m < s_{(-)}t$ and density threshold $d = (1-\epsilon)q$, the density of nodes within a ball of radius m around the infection origin is at least d.*

Proof. From the speed condition, with probability tending to 1, the infection contains all nodes within distance $s_{(-)}t$ of the origin. In particular, it contains the ball of radius m. The expected density in that ball is q (the reporting probability). As in Theorem 7.3.2, since the ball size is at least $\log^2 n$, the probability the density is less than $(1-\epsilon)q$ decays to 0 using Lemma 7.3.1. □

Combining these two results gives the conditions for when the Ball Density Algorithm succeeds. That is, the infection time must be large enough that the 'inner ball' of the epidemic (that is, the largest ball completely contained in the epidemic) includes at least $\log^2 n$ nodes. Second, the expected infection size must be no more than a constant factor less than n. By setting the density threshold closer to q, the factor can be improved, so that the algorithm succeeds when nearly the entire network is infected.

Theorem 7.3.4. *Suppose G is an acceptable graph with size n, and let $\epsilon > 0$ be a small constant. In addition, suppose that the expected number of infected nodes is at most $(1-\epsilon)n$ and $t > b_{(+)}(\log^2 n)/s_{(-)}$. Using the Ball Density Algorithm with parameters m satisfying $b_{(+)}(\log^2 n) < m < s_{(-)}t$ and density $d = (1-\epsilon/2)q$, the algorithm successfully distinguishes a random sickness and an epidemic with probability tending to* 1.

Proof. First, consider a random sickness. From Theorem 7.3.2, all balls of radius m have density less than d with probability approaching 1. In this case, the algorithm corrects label the infection a random sickness. Now consider an epidemic. From Theorem 7.3.3, there is a ball of radius m contained in the epidemic with density at least d with high probability. Again, the algorithm successfully labels it an epidemic. Therefore, both the Type I and Type II error probability tend to 0. □

We require that the expected infection size is at most a small factor less than the size of the network and spreads at least enough to contain $\log^2 n$ nodes. Since it is impossible to distinguish a random sickness from an epidemic when the entire network is infected, this is at least order-wise optimal in the maximum infection size. However, to set the density parameter, we assume that q is known. When it is unknown, we must instead use the Relative Ball Density Algorithm, where the minimum density is set to be a factor of β higher than the density in the rest of the network. The Relative Ball Density Algorithm succeeds in a similar range of times as the previous algorithm.

Theorem 7.3.5. *Let G be an acceptable graph of size n and $\epsilon > 0$ be a small constant. Let $\beta > 1$. Suppose that the expected number of infected nodes is at least $\log^2 n$, and that $t < s_{(+)}^{-1} b_{(-)}(n/(\beta + \epsilon))$. Apply the Relative Ball Density Algorithm with radius m satisfying $b_{(+)}(\log^2 n) < m < s_{(-)}t$ and relative factor β. Then the algorithm correctly identifies the type of infection with probability approaching* 1*.*

Proof. Suppose the infection is a random sickness. Let $k = E[|S_{\text{rep}}|]$. Then the expected density in any set of nodes is k/n. Let $\delta = \frac{\beta-1}{\beta+1}$, so $\beta = \frac{1+\delta}{1-\delta}$. Applying the same method as in Theorem 7.3.2, with probability tending to 1, for each ball, the density within the ball is less than $(1 + \delta)k/n$ and the density outside the ball is at least $(1-\delta)k/n$. Therefore, the ratio between the two is less than β so the algorithm correctly identifies it is a random sickness.

Next, suppose the infection is an epidemic. Let $\delta = \epsilon/(\beta + \epsilon)$. Using Theorem 7.3.3, the infection contains an m radius ball with density at least $(1-\delta)q$. From Lemma 7.3.1, the density of the entire infected set is at most $(1+\delta)q$. From the speed condition, we know with high probability, the epidemic is within a ball of radius $s_{(+)}t$, containing at most $n/(\beta+\epsilon)$ nodes by assumption. No nodes outside that ball report an infection. Therefore, the external density is at most $(1+\delta)q/(\beta+\epsilon)$. After some calculation, we find the ratio of the internal and external density $(1-\delta)(\beta+\epsilon)/(1+\delta)$ is at least β. Hence, the algorithm identifies it as an epidemic with probability tending to 1. $\square$

We only prove the Relative Ball Density Algorithm succeeds for time such that the *maximum* epidemic spread covers nearly up to the network size, in contrast to the time when the expected epidemic size is nearly n for the original algorithm. There may be a constant factor between these times, depending on the network topology. That is, the algorithm may only be order-wise optimal in infection time, not infection size.

Note that the entire network is (likely) infected for $t = s_{(-)}^{-1}b_{(+)}(n)$. In addition, from concavity, $b_{(-)}(n/(\beta+\epsilon)) > 1/(\beta+\epsilon)b_{(-)}(n)$. From this, we conclude the Relative Ball Density Algorithm is order-wise optimal in infection time so long as for some constant C, for all x, $b_{(+)}(x) < Cb_{(-)}(x)$. That is, as long as the lower and upper bounds on neighborhood size are similar. For example, this is true for grids and trees. In addition, for some graphs, such as grids, being order-wise optimal in infection time is the same as being order-wise optimal in infection size. However, for tree graphs, it means success is only guaranteed for infection sizes up to n^{γ} for some $\gamma < 1$. Nevertheless, we do not need knowledge of the reporting rate for this algorithm.

7.3.2 False Positives

For most data sources, the knowledge of the infected nodes is likely to be unreliable. We already include the possibility that there are false negatives, but there are also likely to be false positives, i.e., nodes that report being infected when they are not.

Recall that the number of false positives is parameterized as a factor f of the number of actual infected nodes. Thus, there are at most $f\,|S_{\mathrm{rep}}|$ false positives, and these are spread randomly over the network. We show that our algorithms can tolerate an arbitrary number of randomly located false positives, though the maximum solvable infection size is reduced.

Theorem 7.3.6. *Consider an acceptable graph G of size n, and an infection on the graph, with false positive ratio f. Let ϵ be some small constant. Suppose the infection time is such that $t > b_{(+)}(\log^2 n)/s_{(-)}$ and the expected infection size is less than $(1-\epsilon)n/(1+f)$. Then the Ball Density Algorithm, with parameters m in the range $b_{(+)}(\log^2 n) < m < s_{(-)}t$ and density $d = (1-\epsilon/2)q$, determines the type of infection with probability that tends asymptotically to 1.*

Proof. First, note that adding false positives only increases the density of nodes. Then clearly the Type II error probability decays to 0 as shown in Theorem 7.3.4. The remaining case is when the infection is a random sickness. As compared to the case without false positives, the density is increased by a factor of up to $(1+f)$, for an expected density of $q(1+f)E[|S|]/n$. As before, as long as d is greater than this quantity, the Type I error probability decays to 0. By assumption, $q(1+f)E[|S|] < q(1-\epsilon) < d$, so we are done. $\square$

The Relative Ball Density Algorithm can also succeed in this setting. Again, it can tolerate an arbitrary number of false positives, as long as the infection size is sufficiently low. The maximum infection time is order-wise the same as that in the case without false positives, and hence is also order-wise optimal with balanced neighborhood size bounds.

Theorem 7.3.7. *Suppose G is a size n acceptable graph. Let $\epsilon > 0$ be a small constant, and let $\beta > 1$. Assume that the infection time t satisfies*

$$b_{(+)}(\log^2 n)/s_{(-)} < t < s_{(+)}^{-1} b_{(-)}\left(\frac{n}{(1+f)(\beta+\epsilon)}\right).$$

By using the Relative Ball Density Algorithm with radius m satisfying the inequality $b_{(+)}(\log^2 n) < m < s_{(-)}t$ and with relative factor β, the type of infection can be determined with probability approaching 1*.*

Proof. For this theorem, the random sickness case is the easiest. The composition of false positives and the random sickness is similar to a random sickness with higher reporting rate. Just as in Theorem 7.3.5, the density inside and outside any ball is close to its expectation (and equal for both regions) and hence the Type I error probability tends to 0.

Now consider an epidemic on G. From the lower bound on t, the expected infection size is at least $\log^2 n$. Using the upper bound on t as in Theorem 7.3.5, the density of true reporting nodes over the network is at most

$q(1+f)^{-1}(\beta+\epsilon)^{-1}$. Since the false positives increase this expected density by at most a factor of $(1+f)$, the outer density is at most $q/(\beta+\epsilon)$. As before, the expected density of the ball contained in the infection is q, plus additional density from the false positives. Hence, as desired, the ratio between the densities is at least β with probability tending to 1. □

7.3.3 Unknown Edges

Another source of error is incomplete knowledge of graph structure. Complete knowledge of contact networks may be difficult to determine, and there may be unknown edges. Nevertheless, if these unknown edges are not too numerous, then it is still possible to distinguish epidemics and random sicknesses. We consider two types of missing edges. There may be a large number of missing edges, but they are 'short.' On the other hand, there may be a few missing 'long' edges.

First we consider the case where there are many short edges. That is, suppose that for some constant ℓ, each missing edge e_{ij} satisfies $\text{dist}_{\bar{G}}(i,j) \leq \ell$ as in Section 6.3.1. As before, using this property, we find that the distance between any two nodes i and j on $\bar{G}$ increases by a factor of at most ℓ over the distance on G, since the length of each edge on the shortest path connecting the two nodes increases by at most that factor. Additionally, removing edges only lengthens the distance between nodes, never decreases it. By accounting for the possible increase in distance, we again show that the Ball Density Algorithm can distinguish the infection types.

Theorem 7.3.8. *Let G be an acceptable graph with size n. Suppose the only unknown edges on G are short edges with length at most ℓ. Let $\epsilon > 0$. Assume that the expected number of infected nodes is at most $(1-\epsilon)n$ and $t > b_{(+)}(\log^2 n)/(\ell s_{(-)})$. For the Ball Density Algorithm, use parameters radius m and density d with $\ell b_{(+)}(\log^2 n) < m < s_{(-)}t$ and density $d = (1-\epsilon/2)q$. Then this algorithm correctly determines whether the infection is a random sickness or an epidemic with probability approaching* 1.

Proof. As compared to Theorem 7.3.4, the lower bound on m is scaled up by a factor of ℓ. The ball on $\bar{G}$ of radius m must contain at least $\log^2 n$ nodes, because it contains the ball on G of radius m/ℓ, which by assumption contains at least $\log^2 n$ nodes. Hence, from Theorem 7.3.2, the density of a random sickness on all of these balls is no more than $(1-\epsilon/2)q$, an upper bound on the overall density. Therefore, the Type I error probability goes to 0.

In addition, the ball of radius m on $\bar{G}$ is contained in the ball of radius m on G, since distances only increase. Therefore, in an epidemic, this ball is contained within the infected set and has density greater then $(1-\epsilon/2)q$ by Theorem 7.3.3. From this, the Type II error probability also vanishes. □

From Theorem 7.3.8, we see that by simply increasing the minimum ball size to ensure we cover a sufficient portion of the network even with edges missing, the Ball Density Algorithm succeeds as before. Therefore, we conclude it is very tolerant of missing short edges. A similar result holds for the Relative Ball Density Algorithm.

Theorem 7.3.9. *Consider an acceptable graph G of size n, and let $\epsilon > 0$ be a small constant. Set $\beta > 1$. In an infection, suppose that the number of infected nodes is at least $\log^2 n$, and that $t < s_{(+)}^{-1} b_{(-)}(n/(\beta + \epsilon))$. Using the Relative Ball Density Algorithm with radius m in the range $\ell b_{(+)}(\log^2 n) < m < s_{(-)}t$ and relative factor β, the infection type is correctly determined with probability tending to 1.*

Proof. Just as in Theorem 7.3.8, a ball of radius m on $\bar{G}$ contains at least $\log^2 n$ nodes. In addition, such a ball around the source of an epidemic is contained within the epidemic with high probability as $m < s_{(-)}t$. These are the conditions necessary for the error probability to decay to 0 as shown in Theorem 7.3.5. □

Now consider the case when there are few, but arbitrary length unknown edges. Since these edges are not known, the infection appears to jump across the graph when it traverses on one of these edges. Then suppose there is a bound on the number of these edges, K. Therefore, there are at most K jumps (with at most one per edge), and at most $K + 1$ clustered epidemics on $\bar{G}$. However, each of these clusters has a high density, and the algorithm still succeeds with a slight modification. Namely, we only consider balls containing at least $\log^2 n$ nodes in the algorithm. If there are no such balls at that radius, the infection is labeled a random sickness, though this case will not occur with the radius specified.

Theorem 7.3.10. *Let G be an acceptable graph with size n, and suppose all but K edges are known. Let $\epsilon > 0$ be a small constant. Consider an infection with expected size at most $(1-\epsilon)n$ and duration $t > 2b_{(+)}((K+1)\log^2 n)/s_{(-)}$. Apply the Ball Density Algorithm, setting the parameters m so that $b_{(+)}((K+1)\log^2 n) < m < s_{(-)}t/2$ and density $d = (1-\epsilon/2)q$, with the additional requirement that the number of nodes within any considered ball must be at least $\log^2 n$. Then a random sickness and an epidemic can be distinguished with probability approaching* 1.

Proof. From our additional condition, we know the balls contain $\log^2 n$ nodes. As in previous theorems, we know from Theorem 7.3.2 that the random sickness density is less than d and the Type I error probability goes to 0. Next consider an epidemic. We know the ball on $\bar{G}$ is contained within the ball on G of the same radius. Split the infection into two phases, each of length $t/2$. From the speed condition, for each node within distance $s_{(-)}t/2$ from the infection origin, the ball of radius less than $s_{(-)}t/2$ around that node is contained in the infection. Applying Theorem 7.3.3, we see that, if any such ball has at least $\log^2 n$ nodes, it has the required density.

The main fact to be proved is that there is such a ball of radius m on $\bar{G}$ containing at least $\log^2 n$ nodes. The ball of this radius on G contains at least $(K+1)\log^2 n$ nodes by hypothesis. This ball can be split into 'clusters', where a cluster is a ball around the node on the far side of one of the unknown edges. There are at most $(K+1)$ of these clusters, and therefore, at least one of them has $\log^2 n$ nodes. Then, the ball of radius m around the center of that cluster both is contained in the infection, and contains $\log^2 n$ nodes as desired. □

The range of infection sizes for which we succeed is very similar to case without missing edges. The radius used in the algorithm has a tighter range, and the minimum infection time is larger. Note that the number of missing edges K we can tolerate must satisfy (at least) $K < n/\log^2 n$. The Relative Ball Density Algorithm behaves in a similar way.

Theorem 7.3.11. *Suppose G is an acceptable graph of size n, with at most K unknown edges. Let $\epsilon > 0$ and $\beta > 1$. Assume that the expected number of infected nodes is at least $\log^2 n$ and $t < s_{(+)}^{-1} b_{(-)}(n/(\beta+\epsilon))$. Use the Relative Ball Density Algorithm with radius m in range $b_{(+)}((K+1)\log^2 n) < m < s_{(-)}t/2$ and relative factor β, with the additional requirement that we consider only balls containing at least $\log^2 n$ nodes. This algorithm accurately distinguishes whether the infection is a random sickness or an epidemic with probability going to 1.*

Proof. From the additional algorithm condition, the ball contains at least $\log^2 n$ nodes, so in the same way as Theorem 7.3.5, we see that the Type I error probability goes to 0. For the epidemic, using the result from Theorem 7.3.10, we know there is a ball contained within the infection of radius m on $\bar{G}$ with at least $\log^2 n$ nodes. This ball satisfies the necessary conditions for the same approach as in Theorem 7.3.5 to work. Then the Type II error probability tends to 0. □

7.4 Simulations

We now provide simulation results that confirm our analytic results. In addition, these simulations provide additional insight into how the probability of error changes with variations in the parameters. First, we compare the performance of the Ball Density Algorithm and the relative version with other algorithms. In the next section, we illustrate the effect that changing the weights of the graph has on the probability of error. Finally, we show the probability of error for various numbers of missing edges.

For these simulations, we consider a grid graph where all the horizontal edges have one weight, and the vertical edges have another. Note that structure is desired in these weights. If the weights were simply random, then the infection behavior would be nearly the same as an unweighted infection with a modified edge traversal time distribution. We use graph size $n = 4900$. The reporting probability is $q = 0.25$, and no false positives or missing edges are used unless specified. The ball radius parameter is set to be the optimum value as determined empirically. For the Ball Density Algorithm, we set the density threshold to $d = 0.245$, close to q. For the Relative Ball Density Algorithm, we use a relative ratio of $\beta = 2$. After 1000 trials, the overall probability of error is determined by the average of the error probabilities of both the random sickness and epidemic cases. Other problem parameters are stated in each section below.

7.4.1 Algorithm Comparison

In this paper, we present two algorithms to distinguish random sicknesses from epidemics: the Ball Density Algorithm with fixed density and the relative density of that algorithm. For this section, we denote these the 'Density' and 'Rel. Density' algorithms respectively. Though we show both of these algorithms succeed over similar ranges of infection sizes, we have not directly compared these algorithms analytically. To compare them, we have simulated both on a grid graph, with weights in $\{1, 10\}$. In addition, there are other algorithms to consider. Our algorithms use weighted balls, but it is also possible to use balls where the distance is measured in hop counts. We denote this variation of the Relative Ball Density Algorithm as 'Rel. Density with Hops.' Another possible algorithm is the Ball Algorithm as presented in [46]. In this algorithm, the infection is labeled an epidemic if all the infected nodes can be contained within a ball of a specified radius. Note that this algorithm is (nearly) equivalent to the Relative Ball Density Algorithm with infinite relative factor β. This algorithm is denoted 'Ball', and the version where hop counts are used for the distance is denoted 'Ball with Hops.'

The simulation results are presented in Figure 7.3 (the 'Ball with Hops' algorithm is omitted for clarity). There is a clear ordering of the algorithm performance. From best to worst, the algorithms are 'Rel. Density', 'Ball', 'Rel. Density with Hops', 'Ball with Hops' and finally 'Density.' For example, when around 89% of network is infected, the error probabilities are approximately 1%, 2%, 3%, 4%, and 5% respectively. Then we see that on this graph, the

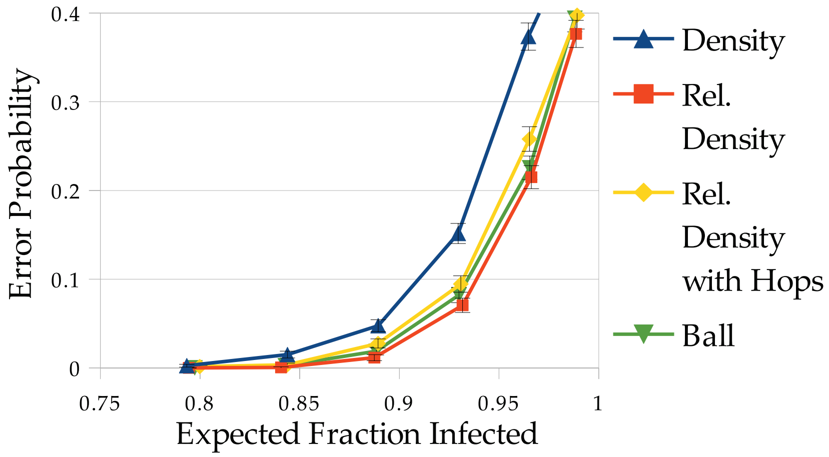


Figure 7.3: This figure shows the overall error probability for a grid graph ($n = 4900$) with the weights on horizontal edges of 1, and on vertical edges of 10 over a range of infections sizes for different algorithms.

Relative Ball Density Algorithm performs better than the other algorithms, including the Ball Algorithm from prior work. We also see that including the effects of the weights in the graph is necessary for optimal performance. The regular Ball Density Algorithm lags behind, partially due to the inability to adapt as well to larger infection sizes, enabling a random sickness to more easily exceed the specified density threshold.

7.4.2 Weights

As the difference in edge weights increases, the more skewed the infection becomes towards the larger edge weights. To examine how tolerant our algorithm is towards different edge weights, we simulated the Relative Ball

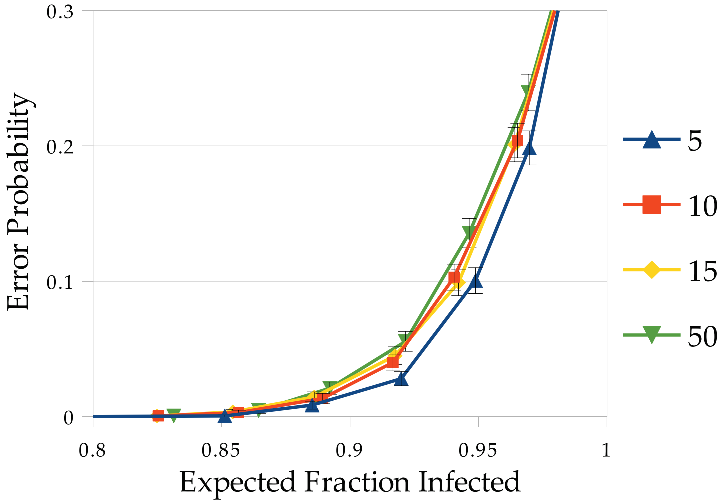


Figure 7.4: This figure illustrates the overall error probability for the Relative Ball Density Algorithm on a grid of size $n = 4900$. The edge weights on the horizontal edges are 1, and the weights on the vertical edges are given in the legend.

Density Algorithm on a grid, fixing the weight of the horizontal edges at 1 and varying the weights of the other edges. The probability of error is shown in Figure 7.4. As the figure shows, though the error probability increases slightly as the weights increase, the performance of the algorithm is very similar regardless of edge weight distribution on this graph. Then we conclude the Relative Ball Density Algorithm appropriately adapts to the weight distribution in this case.

7.4.3 Unknown Edges

One key feature of our algorithm is that it is robust against unknown edges. We simulated the Relative Ball Density Algorithm for various numbers of missing edges to confirm this analytic result. The simulations use a grid graph with edge weights 1 and 10, but add a variable number of long distance edges between nodes chosen uniformly at random from the grid, each with weight 1. These edges are unknown to the algorithm, causing an epidemic to appear as multiple clusters. The probability of error for different numbers of these missing edges is shown in Figure 7.5. Note that due to this construction, the epidemic also spreads somewhat faster the more missing edges there are. As the figure shows, though the error probability increases significantly at smaller infection sizes compared to the case without missing edges, it is still low until a majority of the network is infected. In addition, the error probability increases very slowly as the number of missing edges increases.

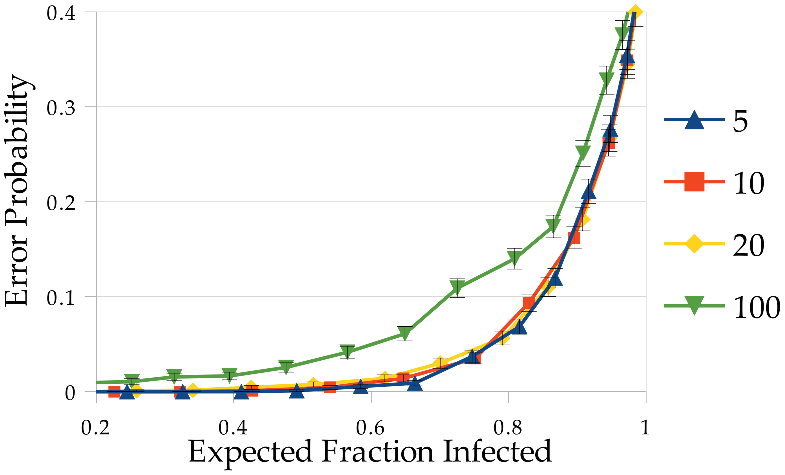


Figure 7.5: This figure presents the overall error probability when using the Relative Ball Density Algorithm on a grid graph with $n = 4900$ with horizontal and vertical edge weights of 1 and 10 and additional unknown random edges of weight 1, for various numbers of missing edges.

Chapter 8

Conclusions and Future Work

In this thesis, we have considered the problem of distinguishing between two infection processes when only limited and unreliable information is available. The fundamental case of this problem is when an infection appearing among the population may represent just a random sickness, or an epidemic that must be identified. Quickly determining when an epidemic occurs is frequently essential is rapidly understanding, containing, and curing it. We present two algorithms to solve this problem, the Threshold Ball Algorithm and the Threshold Tree Algorithm, both of which use the idea of clustering of the infected nodes to determine if an epidemic is present. We show that both of these algorithms achieve asymptotically vanishing error probabilities over a large range of infection sizes on three standard graph topologies, grids, trees, and Erdös-Renyi graphs. In fact, for grids, our maximum achieveable infection size is order-wise optimal. From our analytical and simulation results, we conclude the Threshold Ball Algorithm has superior or nearly equal performance to the Threshold Tree Algorithm, and is more efficient to implement.

We also consider the case when we must distinguish between two epidemics with differing network structure. To solve this problem, we develop two conditions on the graphs. First, it must satisfy a speed constraint, stating roughly that the epidemic cannot travel faster than a constant speed on that graph. Second, the graph must be sufficiently spread out so that a set of random nodes isn't clustered on the graph. When those conditions holds, we show that the Relative Ball Algorithm can determine which epidemic is the causative process with high probability. These two conditions, the speed and spread conditions, ensure that the epidemic is well behaved and we apply related conditions for the rest our results.

In order to make our algorithm more robust, we develop the Quantile Ball Algorithm and the Multiple Ball Algorithm. This modification is sufficient to handle the case when there are false positives. In fact, we show that we can handle the case when an arbitrarily large fraction of the reporting nodes are false positives if they are located randomly, and up to the maximum possible fraction of false positives if they are placed adversarially. The Multiple Ball Algorithm is also sufficiently robust to handle unknown edges in the epidemic's graph, including an arbitrarily large number of short edges. We also show that in the case of two possible mixed infections, where the random sickness and epidemic occur simultaneously, this algorithm can determine which whether the infection occured due to the more infectious process provided the processes are sufficiently distinct.

Finally, we develop a new algorithm for weighted graphs termed the Ball Density Algorithm. We demonstrate that this algorithm is able to distinguish between a random sickness and an epidemic under these more challenging circumstances. In addition, this algorithm is shown to be robust to both false positives as well as unknown edges of the graph. We find empirically that this algorithm slightly outperforms the Threshold Ball Algorithm.

8.1 Future Work

Though we have considered many variations of the problem of distinguishing infection processes, as always there are still many question left unanswered. Throughout this work, we only consider epidemics based on the SI infection model, where once nodes become sick, they never recover. However, in many cases, the susceptible-infected-recovered (SIR) model would be more appropriate, especially in the case of diseases. In this model, where nodes recover from their infection, the expected 'shape' of the infected region is not simply a cluster of nodes, but rather contains nodes only at the edges of a ball, and appears closer to a torus. Due to the reduction in the number of reporting nodes (since less nodes are infected at a given time), our current algorithms may have poor performance. An improved algorithm designed for this case would be valuable in many settings.

Though we have shown that in the majority of cases, our algorithms can determine the causative infection process for maximum infection times that are order-wise optimal, this does not mean they are order-wise optimal in infection size. For graphs in which the infection spreads rapidly such as tree-like graphs, these algorithms only succeed when up to n^{β} nodes are infected for some constant β (depending on the graph, algorithm, and other parameters). This raises the question, how large can the infection be such that it is still possible to distinguish a random sickness from an epidemic for these graph topologies? Developing a converse result determining the necessary conditions for our algorithm to succeed would help answer this question. In addition, it would be useful to know exactly how the performance of our algorithm compares to the theoretical optimal algorithm. Along these lines, work could focus on developing alternative algorithms that may have superior performance, perhaps by considering a more restrictive class of graphs.

Appendices

Appendix A

Chapter 2 Proofs

A.1 Proof of Theorem 2.3.3

Proof of Theorem 2.3.3(a). To prove this theorem, we prove the following more general statement. Let m be the threshold for the Ball Algorithm and suppose $(2m/d + 1) < n^{1/d}$. If for some $\epsilon > 0$,

$$t < \frac{m}{d\mu(1+\epsilon)},$$

the Type II error probability decreases to 0 as t, m, and n increase. In addition, the Type I error probability also decreases to 0 in the limit if

$$t^d q \left(\frac{n^{1/d} - 2m/d - 1}{n^{1/d}} \right) = \omega(\log n).$$

We begin with the Type II error probability, which we denote by E_{II}: the probability we mistake an epidemic for a random sickness. As long as m is chosen as in the statement of the theorem, we are guaranteed that if the sickness is in fact from an epidemic, then using the above lemma, the spread of the infection is limited to the subgrid $[-m/d, m/d]^d$ with high probability, where the origin is set to be the original infected node. Consequently, all nodes must be within m steps of the origin since the grid is d-dimensional. That is, we have

$$E_{II} < 1 - P\{B(t) \subset [-m/d, m/d]^d\}$$
$$< C_1 t^{2d} e^{-C_2 t^{-1/2}(m/(d\mu)-t)},$$

from Lemma 2.3.1, where we use $x = \min(t^{-1/2}(m/(d\mu) - t), t^{1/2})$. Therefore, given $\epsilon > 0$, $t < \frac{m}{d\mu(1+\epsilon)}$, indeed the error goes to 0 as t and n increase.

Next, we consider Type I error, E_I: the probability we mistake a random sickness for an infection process. This happens if all the reporting sick nodes happen to fall inside the ball of radius m/d. Recall the expected size of the random sickness is the same as that of the epidemic. We can get a lower bound on this number for the infection process (and hence for the random sickness process) this time using the inner bound on B_0. For the infection process, the second part of Lemma 2.3.1 asserts that the infected region contains all nodes within the $l1$-ball of radius $w = (1 - C_3 t^{-1/(2d+4)}(\log t)^{1/(d+2)})\mu t$ with probability at least $1 - P_1$, where

$$P_1 = C_4 t^d \exp(-C_5 t^{(d+1)/(2d+4)}(\log t)^{1/(d+2)}).$$

Therefore at least $2\lfloor w/d \rfloor^d$ nodes will be sick with that probability, and hence there will be on average, at least $2q\lfloor w/d \rfloor^d$ sick nodes reporting. What is the probability that the random sickness model with (at least) this many sick nodes will have all reporting nodes inside the sub grid $[-m/d, m/d]^d$? There are $L = (2m/d + 1)^d$ nodes in that region. Evidently, any given sick node satisfies that property with probability L/n, so they all satisfy it with probability at most $(L/n)^{2q(w/d)^d}$. Note that any dependence between sick nodes only reduces the probability. After this, we use a union bound to find that the probability no such region contains all sick nodes is at most $P_2 = n(L/n)^{2q(w/d)^d}$.

Putting it all together, we have,

$$\begin{aligned} E_I &< 1 - (1 - P_1)(1 - P_2) < P_1 + P_2 \\ &< C_4 t^d \exp\left(-C_5 t^{(d+1)/(2d+4)} (\log t)^{1/(d+2)}\right) \\ &+ n \left(\left(\frac{2m/d + 1}{n^{1/d}} \right)^d \right)^{2q(w/d)^d} . \end{aligned}$$

and

$$2(w/d)^d \geq 2d^{-d} \mu^d t^d (1 - dC_3 t^{-1/(2d+4)} (\log t)^{1/(d+2)}).$$

Note that P_2 dominates as n increases. We want to find the regime when this probability tends to 0. That is, we want

$$\begin{aligned} n \exp(2d^{-d} \mu^d t^d qd \ln \left(1 - \frac{n^{1/d} - 2m/d - 1}{n^{1/d}} \right) \\ (1 - dC_3 t^{-1/(2d+4)} (\log t)^{1/(d+2)})) \to 0. \end{aligned}$$

Using a Taylor expansion and some simplification, we find a sufficient condition for this is that

$$t^d q \left(\frac{n^{1/d} - 2m/d - 1}{n^{1/d}} \right) = \omega(\log n).$$

This completes the proof of the general statement. In addition, the Type I error can be shown to dominate in the range of interest. Theorem 2.3.3(a) follows immediately using the threshold provided.

□

Proof of Theorem 2.3.3(b). Let X_{rep} be the number of reporting sick nodes, and let $\bar{X} = X_{\text{rep}}/q$ (that is, $\bar{X}$ is basically the expected number of sick nodes based on the number reporting). Recall S is the complete set of sick nodes. From the Lemma 2.3.2, we have

$$P(\bar{X} \log\log n < |S|) \to 0$$

Let μ be the asymptotic rate at which an infection travels as before, and let $\epsilon > 0$. From the proof of Theorem 2.3.3(a), at time t, we know for $\delta > 0$

$$P(|S| < (2(1-\epsilon)\mu t/d)^d) \to 0$$

Hence $t < \frac{(\bar{X} \log\log n)^{1/d}}{2(1-\epsilon)\mu/d}$ with high probability. Naturally increasing t only increases the infection size, so it is only necessary to consider the maximum likely t. In particular, if the threshold $m = 1.1d\mu t_{max} = \frac{1.1d^2 \bar{X} \log\log n)^{1/d}}{2(1-\epsilon)}$, then from Theorem 2.3.3(a), the adaptive thresholding will work with Type I error probability approaching 1. In addition, if $\bar{X}$ is $\omega(\log n)$, the Type II error probability will decay to 0 as well from the same theorem. □

A.2 Proof of Theorem 2.3.4

Proof of Theorem 2.3.4(a). First consider the Type II error probability. Using Lemma 2.3.1 like in Theorem 2.3.3, the epidemic is contained in a ball of radius $1.5\mu t$ with high probability and hence the size of the epidemic is less than $(3\mu t)^d = m$. Since the nodes in the epidemic are connected, the reporting nodes can clearly be connected by a Steiner tree with size less than that of the epidemic. Therefore, the Type II error probability decays to 0.

Now consider Type I errors, so assume the infection is caused by a random sickness. With Lemma 2.3.1, we find that the ball of radius $0.5\mu t/d$ is contained in the epidemic, and therefore at least $\left(\frac{\mu t}{2d}\right)^d$ nodes are infected. Hence, $E[|S|] > (2d/3)^d m$. Assume by hypothesis that $E[|S|] < n/(8 \times (3/d)^d \log logn/q)^d$. Divide the grid into blocks of size $L = \frac{n \log\log n}{qE[|S|]}$ (that is, the regions should be grid sections of side length $L^{1/d}$). Note that the expected number of reporting nodes in each block is $\log\log n$. It is easy to see that at least half of the blocks contain a reporting node, for example with a Chernoff bound. Consider the shortest path (duplicate edges allowed) connecting all the reporting nodes. This is no more than twice the length of the Steiner tree since a path can traverse a tree by traveling along each edge twice.

For each $2 \times 2 \times \cdots \times 2$ region, color each block a different color, and each such region in the same pattern. Note that there are 2^d colors used. Consider the sequence of colors of the blocks the path travels through. Since blocks of the same color are separated by a distance of $L^{1/d}$, whenever a color is repeated, the path must travel at least that distance. Because there are only 2^d colors and at least $n/(2L)$ blocks, there are at least $n/(2^{d+1}L) - 1$ such repetitions (subtracting the first instance of the colors). Therefore, the path has length at least

$$
\begin{aligned}
\left(\frac{n}{2^{d+1}L} - 1\right) L^{1/d} &= \left(\frac{qE[|S|]}{2^{d+1}\log\log n} - 1\right)\left(\frac{n\log\log n}{qE[|S|)]}\right)^{1/d} \\
&> \frac{qE[|S|]}{2^{d+2}\log\log n}\left(\frac{n\log\log n}{qE[|S|]}\right)^{1/d} \\
&> \frac{n^{1/d}(qE[|S|])^{1-1/d}}{2^{d+2}\log\log n} \\
&> \frac{n^{1/d}q^{1-1/d}d^d m}{4(3^d)\log\log n E[|S|]^{1/d}} \\
&> 2q^{-1/d}m > 2m
\end{aligned}
$$

where the last line uses our hypothesis on maximum number of infected nodes. Hence, the Steiner tree has size at least m and the Type I error probability approached 0. □

Proof of Theorem 2.3.4(b). By a Chernoff bound, we see that m is larger than the number of infected nodes with high probability. Since the Steiner tree is smaller than the number of infected nodes, the Type II error probability clearly decays to 0. Turning to the Type I error rate, we can apply the same approach as for the non-adaptive case. In this case, set $L = \frac{n\log\log n}{X_{\text{rep}}}$. Applying the same reasoning as before, the length of the Steiner tree connecting random nodes is at least

$$
\begin{aligned}
\frac{X_{\text{rep}}}{2^{d+3}\log\log n}\left(\frac{n\log\log n}{X_{\text{rep}}}\right)^{1/d} &> \frac{n^{1/d}X_{\text{rep}}^{1-1/d}}{2^{d+3}\log\log n} \\
&= \frac{n^{1/d}qm}{3\times 2^{d+3}X_{\text{rep}}^{1/d}(\log\log n)^2}
\end{aligned}
$$

Therefore, we are done if $3 \times 2^{d+3}(\log\log n)^2 X_{\text{rep}}^{1/d} < qn^{1/d}$, that is, if $X_{\text{rep}} < q^d n/(3 \times 2^{d+3}(\log\log n)^2)^d$. From standard Chernoff bounds, we see $X_{\text{rep}} < qE[|S|]\log\log n \leq E[|S|]\log\log n$ with high probability. From our hypothesis, using the appropriate constant C_1, $E[|S|] < n/(3 \times 2^{d+3})^d(\log\log n/q)^{3d}$. Therefore, $X_{\text{rep}} < \frac{q^{3d}n}{(3\times 2^{d+3})^d(\log\log n)^{3d-1}} < q^d n/(3 \times 2^{d+3}(\log\log n)^2)^d$ as desired. □

A.3 Proof of Theorem 2.3.5

Proof of Theorem 2.3.5(a). To prove this theorem, we prove the following more general statement:

For some constant $\beta < 1$, if $qE[|S|] = \omega(1)$ and $E[|S|] < n^\beta$, then the Type I error probability tends to 0. Next, there exists a constant b such that if $b_0 > b$ and the threshold $m > b_0 t$ for all n, then the Type II error probability converges to 0 asymptotically, as the tree size scales.

The Type II error bound follows from the fact that the epidemic speed is no more than a constant b [7].

The Type I error result follows simply as well. Given the branching ratio, c, there are $\frac{c^{m+1}-1}{c-1}$ nodes within a distance m from the root. The probability of a Type I error is (approximately) $(\frac{c^m}{n})^{|S_{\text{rep}}|}$ – the probability that the randomly sick nodes are closer than the threshold m to the root. Then if c^m is $o(n)$, it is sufficient that the probability that $|S_{\text{rep}}| = 0$ goes to 0. This occurs if the expected number of reporting sick nodes is $\omega(1)$. That is, we need $qE[|S|] = \Theta(qe^{(c-1)t}) = \omega(1)$, calculating $E[|S|]$ with a simple differential equation (shown at the end of this proof). Alternatively, if $c^m = \alpha n$ for some constant $\alpha < 1$, then we require $|S_{\text{rep}}|$ to increase with n with probability 1. The same condition as before is sufficient for this to be true. This completes the Type I result.

Using both these results, there is a choice of m such that both error types become rare as long as $c^{b_0 t} < \alpha n$, so $c^t < (\alpha n)^{1/b_0}$. The theorem follows using a particular threshold.

Now we conclude by showing how we can calculate $E[|S|]$ with the following differential equation. Let t' be a variable infection time. Let $X(t')$ be the number of infected nodes and $Y(t')$ be the number of 'border' nodes, uninfected nodes adjacent to an infected node. When a new node becomes infected, $Y(t')$ increases by $c-1$. Because of this, and since border nodes become infected at rate 1, $Y(t') = (c-1)X(t') + 1$ and $dE[Y(t')]/dt = (c-1)E[Y(t')]$. Solving this equation gives $E[Y(t')] = ce^{(c-1)t'}$ and $E[X(t')] = c/(c-1)e^{(c-1)t'} - 1/(c-1) > e^{(c-1)t'}$. Therefore, we find $E[|S|] \approx c/(c-1)e^{(c-1)t}$. □

Proof of Theorem 2.3.5(b). First, note that $E[|S|]$ scales at least as $e^{(c-1)t}$ (until the infection reaches the leaves of the graph). In fact, for any fixed $\epsilon > 0$, $|S| > e^{(c-1)t/(1+\epsilon)}$ with probability approaching 1 (for example, see [25]). Now we can proceed as in the proof of Theorem 2.3.3(a).

As before, let X_{rep} be the number of reporting sick nodes and $\bar{X} =$

X_{rep}/q so $\bar{X} \log \log n < |S|$ with high probability. Then we conclude $t_{max} = 1/(c-1) \log(X_{\text{rep}}/q(\log \log n)^2)$. Hence, by setting $b_2 = b/(c-1)$, we see the Type II error probability converges to 0 by Theorem 2.3.5(a). Using the same theorem, we see the Type I error will also go to 0. □

A.4 Proof of Theorem 2.3.6

Proof of Theorem 2.3.6(a). We prove the following generalization of the theorem: The Type I error probability converges to 0 for any choice of the threshold $m = o(qE[|S|] \log n)$ with $qE[|S|] = O(n^\alpha)$ for some $\alpha < 1$. In addition, the Type II error probability converges to 0 if $m = \omega(E[|S|])$.

To prove the Type II error result (mistaking an infection for a random sickness), note that the size of the infection is $E[|S|] \leq e^{(c-1)t}$. Since the Steiner tree containing the reporting nodes can be no larger than the infection itself, the Type II error converges to 0 as long as we use a threshold $m = \omega(E[|S|])$ from Markov's inequality.

Next, we evaluate the Type I error probability (mistaking a random sickness for an infection). This requires estimating the size of the Steiner tree containing the reporting sick nodes. Suppose there is an $\alpha < 1$ such that $E[S_{\text{rep}}] = O(n^\alpha)$. Since the number of sick nodes increases with n, the probability that there are sick nodes on at least two subtrees of the root node goes to 1, hence the root of the tree is in the Steiner tree connecting the randomly sick nodes with high probability. Given this, we see that a node is in the Steiner tree if and only if it is infected or a node below it in the tree is infected. Let $N = |S_{\text{rep}}|$. Since $E[|S_{\text{rep}}|]$ is $\omega(1)$, N is $\omega(1)$ with high probability. Choose the first level in the tree that has at least N/c nodes. Then there are between N/c and N subtrees below that level. It is straightforward to show that each sick node in the tree has at least a 1/2 probability of being a leaf node since $c \geq 2$. Since at least N nodes are sick, at least $N/4$ of the leaf nodes are sick and distributed independently among the at most N subtrees. Therefore, the total number of subtrees with sick nodes at the bottom is at least $N/(8c)$. In addition, each leaf node in a separate subtree requires a path at least up to the aforementioned level in the Steiner tree. This gives us the following high probability bound on the Steiner tree size.

$$\begin{aligned}\text{Steiner Tree Size} &> \frac{N}{8c}(\log_c n - \log_c N) \\ &> N\frac{(1-\alpha)\log_c n}{8c} \\ &= |S_{\text{rep}}|\,\frac{(1-\alpha)\log_c n}{8c}.\end{aligned}$$

For any $w = o(E[|S_{\text{rep}}|])$, we know that $|S_{\text{rep}}| > w$ with probability approaching 1. Also, if $E[|S_{\text{rep}}|] = O(n^{\alpha})$, then $S_r = O(n^{\alpha})$ with high probability. Therefore, if $m = o(qw \log_c n)$, which is equivalent to $m = o(E[|S_{\text{rep}}|] \log n)$, the Type I error probability tends to 0. □

Proof of Theorem 2.3.6(b). Let X_{rep} be the number of reporting sick nodes, $\bar{X} = X_{\text{rep}}/q$. Then $\bar{X} \log\log n$ upper bounds $|S|$ with high probability. Like before, $|S| \log\log n > E[|S|]$ with probability approached 1. Then from Theorem 2.3.6(a), we see that both probability of errors will decrease to 0 asymptotically. □

A.5 Proof of Theorem 2.3.7

Proof of Theorem 2.3.7(a). The proof follows similar lines as in the previous section, so we omit most details. In particular, we show the following: Using a threshold $m < \frac{\log n}{3 \log c}$ and $qE[|S|] = \omega(1)$, the probability of a Type I error is at most $o(n^{-1})$. In addition, the probability of a Type II error converges to 0 as long as $m > bt$ for a constant b specified in the proof.

We bound the probability of a Type II error again using the notion of the fastest sustainable transit rate from first-passage percolation [7]. As in Theorem 2.3.5, the constant b comes from the calculation of the infection spreading rate, and the results follow similarly.

To control the probability of a Type I error, we have to bound the probability that all randomly sick nodes are within a ball of radius m on the graph. A sufficient condition for this is that all nodes are within distance $2m$ from a given sick node, or there are 0 nodes sick. The latter probability is simply $(1-q)^n$ which decays exponentially. Also, with probability $1 - o(n^{-1})$, the number of nodes within a distance $2m$ from a given sick node is no more than $16m^3c^{2m}\log n$ [11]. Then the error probability in this case is at most $\left(1 - \frac{16m^3c^{2m}\log n}{n}\right)^n$. Then this decays exponentially as long as $c^{2m} = o(n)$, which occurs when $m < \frac{\log n}{3 \log c}$. Therefore, it is sufficient to show $m < \frac{\log n}{3 \log c}$. Since the infection size is $o(n)$, we use a branching process approximation to find that for some λ, $E[|S|] \to e^{\lambda t}$ [18]. Define $\beta_2 = \lambda/(3 \times 1.1^2 b \log c)$. Assume

$E[|S|] < n^{\beta_2}$ as hypothesized. Then asymptotically with high probability,

$$\lambda t < 1.1\beta_2 \log n.$$

With some computation, $m = 1.1bt < \log n/(3\log c)$. Hence, the Type I error probability also decays to 0. □

Proof of Theorem 2.3.7(b). As shown in [18], $E[|S|]$ scales asymptotically as $e^{\lambda t}$ for some constant λ. In particular, for abitrary constant $\epsilon > 0$, $E[|S|] > e^{\lambda t/(1+\epsilon)}$ with probability approaching 1. Then let X_{rep} be the number of reporting sick nodes, and $\bar{X} = X_{\text{rep}}/q$, so $\bar{X} \log\log n$ will upper bound $|S|$ with high probability. From this, we conclude $t_{max} = 1/\lambda \log(X_{\text{rep}}/q(\log\log n)^2)$. Then by Theorem 2.3.5(a), with $b_2 = b/\lambda$, we see that the Type II error probability converges to 0. From the same theorem, the Type I error will go to 0 as well. □

A.6 Proof of Theorem 2.3.8

Proof of Theorem 2.3.8(a). We show the following more general statement: The Type II error probability decays to 0 if the threshold is chosen as $m = \omega(E[|S|])$ and $E[|S|] = o(n)$. The Type I error probability goes to 0 when $m < kqE[|S|]$ for some constant $k = o(\log(n/(qE[|S|])^2))$ and $qE[|S|] = o(\sqrt{n})$.

First, if the sickness is from an infection, the smallest tree connecting the reporting sick nodes must have size no more than the actual number of sick nodes. Hence, to bound the Type II error, it is sufficient to bound the probability the number of infected nodes is over a certain size. This probability decreases to 0 as long as m is $\omega(E[|S|])$ when $E[|S|] = o(n)$. To see this, recall that in this regime, the graph looks locally tree-like. Consequently, we can bound the maximum number of infected nodes using bounds on the distance an infection can travel (e.g., see [7]). Again, Markov's inequality provides the exact error bound in the theorem statement.

To control Type I error probability, that a random sickness is mistaken for an infection, we must lower bound the size of the Steiner tree of a random sickness. For $v \in S_{\text{rep}}$, let d_v denote the distance from that node to the nearest other sick node. First we show that $\sum_{v \in S_{\text{rep}}} d_v \leq 2\text{SizeTree}(G, S_{\text{rep}})$. Note that the bound is attained for some graphs, such as a star graph with the central node uninfected.

Consider the Steiner tree subgraph, and duplicate all edges on it. Since the degree of each node in the subgraph is even, there is a cycle that connects all these nodes. Naturally, the length of this cycle, which is twice the size of the Steiner tree, is larger than the length of the smallest cycle connecting all sick nodes. In addition, the length of this cycle is at least $\sum_{v \in S_{\text{rep}}} d_v$, since the distance from one sick node to the next sick node in the cycle is clearly no smaller than the distance from that sick node to the closest sick node. This establishes that $\sum_{v \in S_{\text{rep}}} d_v \leq 2\text{SizeTree}(G, S_{\text{rep}})$.

Now we simply need to bound d_v. To do this, we need an understanding of the neighborhood sizes in a $G(n, p)$ graph. But as the size of the graph scales, this is also straightforward to do: recalling that the probability of an edge is c/n and hence the expected degree of each node is (asymptotically) c, then for typical nodes and arbitrary constant $\epsilon > 0$, there are no more than $((1+\epsilon)c)^d$ nodes within distance d provided that $d = \omega(\log\log n)$, using a branching process approximation.

Let X_{rep} be the number of reporting sick nodes. Now assume $X_{\text{rep}} = o(\sqrt{n})$. Let $\epsilon > 0$ and $l = \epsilon n/X_{\text{rep}}^2$. Let $k = o(\log(n/X_{\text{rep}}^2))$. Using the above distance distribution calculation, we find that each sick node v, there are less than l nodes within distance k. As the sick nodes are randomly selected, the probability that none of these are within a distance k from v is bounded by $(1 - X_{\text{rep}}/n)^l \rightarrow e^{-\epsilon/X_{\text{rep}}} \rightarrow 1 - \epsilon/X_{\text{rep}}$. Thus the distance to the closest sick node to v is at least k, i.e., $d_v > k$, with high probability, and using a simple union bound, the same is true, simultaneously, for all sick nodes. Hence the Steiner tree joining the set of *reporting* sick nodes is of size at least $\text{SizeTree}(G, S_{\text{rep}}) \geq (1/2)\sum d_v = (1/2)kqE[|S|]$, with probability decaying to zero. Therefore, the Type I error probability tends to 0 as long as the threshold satisfies $m < kqE[|S|]/2$, for $k = o(\log(n/(qE[|S|])^2))$. Using this result, we find that the Tree Algorithm can succeed so long as $q\log(n/(qE[|S|])^2) = \omega(1)$. This is a complex condition, though the conditions given in the theorem are sufficient for it to be true. □

Proof of Theorem 2.3.8(b). As before, let X_{rep} be the number of reporting sick nodes, $\bar{X} = X_{\text{rep}}/q$. Then $\bar{X}\log\log n$ upper bounds $|S|$ with high probability. As in Theorem 2.3.8(a), $|S|\log\log n > E[|S|]$ with probability approaching

1. Then from Theorem 2.3.8(a), we see that both probability of errors will decrease to 0 asymptotically. □

Appendix B

Chapter 5 Proofs

B.1 Proof of Theorem 5.3.1

Proof. First we show that no infection (from a single seed) spreads farther than a distance C_3, so each infection contains at most a constant $\bar{d}^{C_3+1}$ nodes (where, recall, $\bar{d}$ is a bound on the maximum degree of the graph). Consider an arbitrary seed a and all paths of length $C_3 + 1$ beginning at a. There are at most $\bar{d}^{C_3+1}$ such paths. An infection from a must spread over one such path in time t_0 to spread farther than distance C_3. Since the traversal time of an edge has distribution $\text{Exp}(\eta_0)$, the probability the infection can spread over the edge in time t_0 is $1 - e^{-\eta_0 t_0} < \eta_0 t_0$. Then using a union bound, the probability that the infection spreads more than a distance C_3 is less than $(\bar{d}\eta_0 t_0)^{C_3+1}$. Let ϵ_2 satisfy $0 < \epsilon_2 < 1$. By hypothesis, the expected number of seeds is $\gamma_0 t_0 = \omega(\log n)$ (as $q \leq 1$), so since the number of seeds is binomially distributed, from standard concentration results, the number of seeds is at most $1 + (1 + \epsilon)\gamma_0 t_0$ with probability tending to 1. Let P be the probability the infection spreads farther than distance C_3. Then from a final union bound,

$$
\begin{aligned}
P &< (1+(1+\epsilon)\gamma_0 t_0)\left(\bar{d}\eta_0 t_0\right)^{C_3+1} \\
&= o\left(2\gamma_0 t_0 \bar{d}^{C_3+1}(\gamma_0 t_0)^{-1}\right) \qquad \text{(B.1)} \\
&= o(2\bar{d}^{C_3+1}).
\end{aligned}
$$

Eq. (B.1) follows from our hypothesis $\eta_0 t_0 = o\left((\gamma_0 t_0)^{-1/(1+C_3)}\right)$. Therefore, $P \to 0$ so the infection travels no more than a distance C_3 with probability tending to 1.

Now we need to show no collection of β ball of radius m contains over an α fraction of the reporting nodes. This is sufficient even for the Scaling Multiple Ball Algorithm since the m is an upper bound on the radius of each ball. We first consider all infected nodes. Let $\epsilon > 0$ be a constant as specified in the theorem statement. For convenience, let $C_4 = \bar{d}^{C_3+1}$, the maximum number of nodes in a ball of radius C_3. Consider an arbitrary node a, and let $B_{\text{inner}} = \text{Ball}(a, m)$, $B_{\text{outer}} = \text{Ball}(a, m + C_3)$. Then from the previous result, any seed that has an infection that spreads to a node in B_{inner} must be inside B_{outer} (since it can only travel a distance C_3). By the hypothesis that $m + C_3 < b\left(\frac{\alpha n}{C_4 \beta(1+\epsilon)}\right)$, $|B_{\text{outer}}| < \frac{\alpha n}{C_4 \beta(1+\epsilon)}$. Therefore, β balls contain less than $\frac{\alpha n}{C_4(1+\epsilon)}$ nodes. Let u be the number of seeds, so $u = \omega(\log n)$, again by hypothesis. Then from Lemma 4.2.2, the number of seeds within B_{outer} is less than $\frac{\alpha u}{C_4(1+\epsilon/2)}$ with probability greater than $1 - 1/n^2$. Each of these seeds infects less than C_4 nodes, so the total number of infected nodes within B_{inner} (which must all be from seeds in B_{outer}) is less than $\frac{\alpha u}{1+\epsilon/2}$. Hence, this ball contains less than a $\frac{\alpha}{1+\epsilon/2}$ fraction of the infected nodes.

Finally, we need to show the reporting process does not significantly impact the fraction of infected nodes seen in the balls. We consider an equivalent method of choosing the reporting nodes: first the number of reporting nodes is chosen (with the appropriate distribution), and then these are distributed uniformly over the infected nodes. Let X_{rep} be the number of reporting nodes $|S_{\text{rep}}|$. Then we need to find the probability that αX_{rep} reporting nodes are within B_{inner}. As we just showed, the probability that any particular reporting node is within that region is at most $\frac{\alpha}{1+\epsilon/2}$. From a standard balls-in-bins argument like in Lemma 4.2.2, since $\alpha X_{\text{rep}} = \omega(\log n)$, $P(|S_{\text{rep}} \cap B_{\text{inner}}| > \alpha X_{\text{rep}}) < 1/n^{\beta+1}$. That is, the probability that at least αX_{rep} of the reporting nodes are in that region is at most $1/n^{\beta+1}$.

Since each collection of balls contains over an α fraction of the reporting nodes with probability no more than $1/n^{\beta+1}$, from a union bound, we find the probability that any of the n^{β} possible set of balls exceeds this bound is at most $1/n$. In this case, our algorithm correctly labels it 'RANDOM'. Therefore, the Type I error probability decays to 0 as desired. □

B.2 Proof of Theorem 5.3.2

Proof. First we consider the Multiple Ball Algorithm, we show an upper bound on the number of seeds (recall seeds are the nodes randomly infected). The number of seeds is equal to one (the initially infected node) plus a Binomial random variable with mean $\gamma_1 t_1$. Let U be the set of seeds. Since $\frac{\beta}{\alpha} =$

$\omega(1+\gamma_1 t_1)$, from the distribution, $\frac{\beta}{\alpha} > |U|$ with probability scaling to 1.

Define the function $R(a)$ for seed a as the radius of the epidemic that began at a. Formally, let ω be a realization of the epidemic process, with ω_e defined as the time it takes for the epidemic to spread across edge e and $t_\omega(a)$ as the time from when the seed a became infected to the end of the infection (time t_1). Then for any node v, let $\tilde{t}_\omega(a,v)$ be the distance from a to v on G with edge weights equal to ω_e, which is the time it would take the epidemic to spread from a to v. Finally, $R(a) \overset{\triangle}{=} \max\{\text{dist}_G(a,v) : \tilde{t}_\omega(a,v) \leq t_\omega(a)\}$.

From the speed definition, there exists a constant λ such that for each seed a,

$$P(R(a) > s\eta_1 t_1) < e^{-\lambda \eta_1 t_1}.$$

Now we apply a union bound to see that,

$$\begin{aligned} P(\exists a \in U : R(a) > s\eta_1 t_1) &< \frac{\beta}{\alpha} e^{-\lambda \eta_1 t_1} \\ &< e^{\lambda \eta_1 t_1/2} e^{-\lambda \eta_1 t_1} \qquad \text{(B.2)} \\ &= e^{-\lambda \eta_1 t_1/2} \to 0, \end{aligned}$$

where Equation B.2 follows from the fact that $\log(\beta/\alpha) = o(\eta_1 t_1)$. Therefore, each seed spreads no farther than a distance m with probability tending to 1.

We now show that our algorithm returns ‘EPIDEMIC’ in this case. Cover the seed with the largest (reporting) infection using a ball of radius m, which we showed covers the entire infection for that seed. If $\beta > 1$, we cover the seed with the second largest infection with the next ball, and so on. From our previous result, each such ball covers the infection from the seed entirely. Since there are at most β/α seeds total, β of which are covered, the fraction of reporting infected nodes covered is at least $\beta/\frac{\beta}{\alpha} = \alpha$. Therefore, an α fraction of the reporting infected nodes has been covered a ball of radius m, so the Multiple Ball Algorithm returns 'EPIDEMIC' as desired.

Now we prove the same result for the Scaling Multiple Ball Algorithm. Divide up the total infection time t_1 into β evenly sized sections. Each section has duration t_1/β, and therefore, since $\frac{1}{\alpha} = \omega(1+\gamma_1 t_1/\beta)$, the number of seeds in any region is less than $\frac{1}{\alpha}$ from a union bound and accounting for the initial infected node. Now consider the i^{th} region, starting at time $(\beta-i)t_1/\beta$, and let U_i be the set of all seeds that became infected during that time range. Then each seed $a \in U_i$ has an infection duration less than it_1/β. From the speed condition, in the same way as before, for each seed $a \in U_i$,

$$\begin{aligned} P(R(a) > s\eta_1 it_1/\beta) &< e^{-i\lambda\eta_1 t_1/\beta} \\ &\le e^{-\lambda\eta_1 t_1/\beta}. \end{aligned}$$

From a union bound like before,

$$\begin{aligned} P(\exists i, a \in U_i : R(a) > s\eta_1 it_1/\beta) &< \frac{\beta}{\alpha} e^{-\lambda t_1/\beta} \\ &< e^{\lambda\eta_1 t_1/(2\beta)} e^{-\lambda\eta_1 t_1/\beta} \\ &= e^{-\lambda\eta_1 t_1/(2\beta)} \to 0. \end{aligned}$$

Then with probability tending to 1, for each i and seed $a \in U_i$, the infection from a can be contained in the i^{th} ball of the Scaling Multiple Ball

Algorithm, ordering from smallest to largest. Now, for each $1 \leq i \leq \beta$, cover the largest infection (in terms of reporting nodes) from seeds in U_i with the ball of radius $s\eta_1 t_1 i/\beta$. Therefore, since each region has at most $\frac{1}{\alpha}$ nodes, for each region, at least α fraction of the reporting nodes from infections starting during that time frame are contained in a ball. Thus, the Scaling Multiple Ball Algorithm can contain at least an α fraction of the reporting nodes in the collection of balls, and hence returns ‘EPIDEMIC’. $\square$

Bibliography

[1] Ery Arias-Castro, Emmanuel J. Candès, and Arnaud Durand. Detection of an anomalous cluster in a network. *The Annals of Statistics*, 39:278–304, 2011.

[2] Ery Arias-Castro, Emmanuel J. Candès, Hannes Helgason, and Ofer Zeitouni. Searching for a trail of evidence in a maze. *The Annals of Statistics*, 36:1726–1757, 2008.

[3] Norman T. J. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Griffin, 1975.

[4] Frank Ball and Peter Neal. Poisson approximation for epidemics with two levels of mixing. *The Annals of Probability*, 32(1B):1168–1200, 2004.

[5] Michel Benaim and Raphael Rossignol. Exponential concentration for first passage percolation through modified poincaré inequalities. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 44(3):544–573, 06 2008.

[6] Itai Benjamini, Gil Kalai, and Oded Schramm. First passage percolation has sublinear distance variance. *The Annals of Probability*, 31(4):1970–1978, 10 2003.

[7] Itai Benjamini and Yuval Peres. Tree-indexed random walks on groups and first passage percolation. *Probability Theory and Related Fields*, 98:91–112, 1994.

[8] Vincent D. Blondel, Jean-Loup Guillaume, Julien M. Hendrickx, and Rapha¨el M. Jungers. Distance distribution in random graphs and application to network exploration. *Physical Review*, 76(066101), 2007.

[9] Béla Bollobás. *Random graphs.* Springer, 1998.

[10] Garrett Brown, Travis Howe, Michael Ihbe, Atul Prakash, and Kevin Borders. Social networks and context-aware spam. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, CSCW '08, pages 403–412. ACM, 2008.

[11] Fan Chung and Linyuan Lu. The diameter of sparse random graphs. *Adv. in Appl. Math*, 26:257–279, 2001.

[12] Jon Cohen. Making headway under hellacious circumstances. *SCIENCE*, 313:470–473, July 2006.

[13] C. Corley, D. Cook, A. Mikler, and K. Singh. Text and structural data mining of influenza mentions in web and social media. *International Journal of Environmental Research and Public Health*, 7:596–615, 2010.

[14] M. Damron, J. Hanson, and P. Sosoe. Sublinear variance in first-passage percolation for general distributions. *arXiv:1306.1197*, June 2013.

[15] Nikolaos Demiris and Philip D. O'Neill. Bayesian inference for epidemics with two levels of mixing. *Scandinavian Journal of Stat.*, 32:265–280, 2005.

[16] Nikolaos Demiris and Philip D. O'Neill. Bayesian inference for stochastic multitype epidemics in structured populations via random graphs. *Journal of the Royal Stat. Society Series B*, 67(5):731–745, 2005.

[17] Wenxiang Dong, Wenyi Zhang, and Chee Wei Tan. Rooting out the rumor culprit from suspects. In *Proceedings of IEEE International Symposium on Information Theory*, pages 2671–2675, 2013.

[18] Rick Durrett. *Random Graph Dynamics*. Cambridge University Press, 2007.

[19] F-Secure. Bluetooth-worm:symbos/cabir, 2012. http://www.f-secure.com/v-descs/cabir.shtml.

[20] Ayalvadi J. Ganesh, Laurent Massoulié, and Donald F. Towsley. The effect of network topology on the spread of epidemics. In *INFOCOM*, pages 1455–1466, 2005.

[21] Manuel Gomez-Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. *arXiv:1105.0697*, 2011.

[22] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *ACM Trans. Knowl. Discov. Data*, 5(4):21:1–21:37, February 2012.

[23] Google Flu Trends, http://www.google.org/flutrends/.

[24] A. Gopalan, S. Banerjee, A. Das, and S. Shakkottai. Random mobility and the spread of infection. In *Proc. IEEE Infocom*, 2011.

[25] D. R. Grey. Asymptotic behaviour of continuous time, continuous state-space branching processes. *Journal of Applied Probability*, 11(4):669–677, December 1974.

[26] Clemens Gröpl, Stefan Hougardy, Till NierHoff, and Hans Jürgen Proömel. *Approximation Algorithms for the Steiner Tree Problem in Graphs*, pages 235–279. Kluwer Academic Publishers, 2000.

[27] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A. Zighed. Information diffusion in online social networks: A survey. *SIGMOD Rec.*, 42(2):17–28, July 2013.

[28] C Douglas Howard. Models of first-passage percolation. In *Probability on discrete structures*, pages 125–173. Springer, 2004.

[29] F. K. Hwang and Dana S. Richards. Steiner tree problems. *Networks*, 22(1):55–89, 1992.

[30] Svante Janson, Tomasz Luczak, and Andrzej Rucinski. *Random graphs*, volume 45. John Wiley & Sons, 2011.

[31] K. Johansson. Transversal fluctuations for increasing subsequences on the plane. *Probab. Theory Related Fields*, 116:445–456, 2000.

[32] Nikhil Karamchandani and Massimo Franceschetti. Rumor source detection under probabilistic sampling. In *Proceedings of IEEE International Symposium on Information Theory*, pages 2184–2188, 2013.

[33] Harry Kesten. Percolation theory and first-passage percolation. *The Annals of Probability*, pages 1231–1271, 1987.

[34] Harry Kesten. On the speed of convergence in first-passage percolation. *The Annals of Applied Probability*, 3(2):296–338, Nov 1993.

[35] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 420–429, New York, NY, USA, 2007. ACM.

[36] Andrey Lokhov, Marc Mèzard, Hiroki Ohta, and Lenka Zdeborovà. Inferring the origin of an epidemic with dynamic message-passing algorithm. *Phys. Rev. E*, 90:012801, 2014.

[37] Wuqiong Luo and Wee Peng Tay. Identifying infection sources in large tree networks. In *9th Annual IEEE Communications Society Conference*

on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), pages 281–289, June 2012.

[38] Wuqiong Luo and Wee Peng Tay. Finding an infection source under the sis model. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2930–2934, 2013.

[39] Wuqiong Luo, Wee Peng Tay, and Mei Leng. Identifying infection sources and regions in large networks. *IEEE Transactions on Signal Processing*, 61(11):2850–2865, June 2013.

[40] Russell Lyons. The ising model and percolation on trees and tree-like graphs. *Communications in Mathematical Physics*, 125(2):337–353, 1989.

[41] Russell Lyons and Robin Pemantle. Random walk in a random environment and first-passage percolation on trees. *The Annals of Probability*, 20(1):125–136, 1992.

[42] New York Times Bits Blog, http://bits.blogs.nytimes.com/2012/12/13/lookout-toll-fraud/.

[43] Robert May and Alun Lloyd. Infection dynamics on scale-free networks. *Phys. Rev. E*, 64:066112, 2001.

[44] Kurt Mehlhorn. A faster approximation algorithm for the steiner problem in graphs. *Information Processing Letters*, 27:125–128, 1988.

[45] Eli A Meirom, Chris Milling, Constantine Caramanis, Shie Mannor, Ariel Orda, and Sanjay Shakkottai. Localized epidemic detection in networks with overwhelming noise. *arXiv:1402.1263*, 2014.

[46] Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. Network forensics: random infection vs spreading epidemic. *SIGMETRICS Perform. Eval. Rev.*, 40(1):223–234, June 2012.

[47] Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. On identifying the causative network of an epidemic. In *Proceedings of 50th Annual Allerton Conference on Communication, Control, and Computing*, pages 909–914, October 2012.

[48] Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. Detecting epidemics using highly noisy data. In *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 177–186, 2013.

[49] Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. Local detection of infections in heterogeneous networks. In *Proceedings of INFOCOM, IEEE*, 2015. (To appear).

[50] Seth A. Myers, Chenguang Zhu, and Jure Leskovec. Information diffusion and external influence in networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 33–41, New York, NY, USA, 2012. ACM.

[51] Praneeth Netrapalli and Sujay Sanghavi. Learning the graph of epidemic cascades. *SIGMETRICS Perform. Eval. Rev.*, 40(1):211–222, June 2012.

[52] Philip D O'neill. Bayesian inference for stochastic multitype epidemics in structured populations using sample data. *Biostatistics*, 10(4):779–791, 2009.

[53] Yuval Peres. Probability on trees: An introductory climb. In Pierre Bernard, editor, *Lectures on Probability Theory and Statistics*, volume 1717 of *Lecture Notes in Mathematics*, pages 193–280. Springer Berlin Heidelberg, 1999.

[54] A. Sasaki, H. Gatewood, and A. Ozonoff et. al. Evidenced-based tool for triggering school closures during influenza outbreaks. *Japan. Emerg Infect Dis.*, 15:1841–1843, november 2009.

[55] D. Shah and T. Zaman. Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory*, 57, August 2011.

[56] Devavrat Shah and Tauhid Zaman. Detecting sources of computer viruses in networks: Theory and experiment. *SIGMETRICS Perform. Eval. Rev.*, 86:203–214, 2010.

[57] J. Snow. *On the mode of communication of cholera.* John Churchill, 1855.

[58] George Streftaris and Gavin J. Gibson. Statistical inference for stochatic epidemic models. In *Proc. 17th International Workshop on Statistical Modeling*, pages 609–616, 2002.

[59] Michel Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathmatiques de l'Institut des Hautes tudes Scientifiques*, 81(1):73–205, 1995.

[60] Remco Van Der Hofstad, Gerard Hooghiemstra, and Piet Van Mieghem. First-passage percolation on the random graph. *Probability in the Engineering and Informational Sciences*, 15(02):225–237, 2001.

[61] Wikipedia. Commwarrior-a — Wikipedia, the free encyclopedia, 2012. [Accessed 30-Sept-2012].

[62] Wikipedia. HIV/AIDS — Wikipedia, the free encyclopedia, 2012. [Accessed 30-Sept-2012].

[63] Kai Zhu and Lei Ying. Information source detection in the sir model: a sample path based approach. In *Information Theory and Applications Workshop (ITA), 2013*, pages 1–9. IEEE, 2013.

Vita

Philip Christopher Milling was born in Boynton Beach, Florida on February 10, 1985. He received his Bachelor of Science degrees in Electrical Engineering and in Mathematics from the University of Texas at Austin in May 2007. He received a M. S. E. degree in Electrical and Computer Engineering also from The University of Texas at Austin in May 2009. Currently he is pursuing a Ph.D. under the supervision of Dr. Sanjay Shakkottai.

Permanent address: milling.chris@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.