Distinguishing Infections on Different Graph Topologies

Chris Milling, Constantine Caramanis, Shie Mannor and Sanjay Shakkottai

Abstract

The history of infections and epidemics holds famous examples where understanding, containing and ultimately treating an outbreak began with understanding its mode of spread. Influenza, HIV and most computer viruses spread person to person, device to device, through contact networks; Cholera, Cancer, and seasonal allergies, on the other hand, do not. In this paper we study two fundamental questions of detection. First, given a snapshot view of a (perhaps vanishingly small) fraction of those infected, under what conditions is an epidemic spreading via contact (e.g., Influenza), distinguishable from a "random illness" operating independently of any contact network (e.g., seasonal allergies)? Second, if we do have an epidemic, under what conditions is it possible to determine which network of interactions is the main cause of the spread – the *causative network* – without any knowledge of the epidemic, other than the identity of a minuscule subsample of infected nodes?

The core, therefore, of this paper, is to obtain an understanding of the *diagnostic power of network information*. We derive sufficient conditions networks must satisfy for these problems to be identifiable, and produce efficient, highly scalable algorithms that solve these problems. We show that the identifiability condition we give is fairly mild, and in particular, is satisfied by two common graph topologies: the d-dimensional grid, and the Erdös-Renyi graphs.

I. 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 if something is indeed an epidemic that is best described through contact-network spreading, and secondly, 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" [3], [4]. Understanding the causative network has greatly contributed to controlling the worldwide spread of the virus.

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, or in the case of human viruses, may masquerade as a mostly random sickness, such as the common cold or allergies.

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/hybernation 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. Therefore in this paper, 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.

A. Setting and Results

We model the infection agents (e.g. people or devices) as a set of n nodes, V, of a graph. The nodes in V become infected by an epidemic that spreads according to either graph $G_1 = (V, E_1)$, or $G_2 = (V, E_2)$, propagating along the edges of these graphs, according to an SI model of infection [5]. Given a (potentially small) sub-sample of the infected nodes at a single snapshot in time, our objective is to determine the network over which the epidemic is spreading. If one of the graphs, say G_2, is a star

C. Milling, C. Caramanis and S. Shakkottai are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, USA, Emails: cmilling@utexas.edu, constantine@utexas.edu, shakkott@austin.utexas.edu. S. Mannor is with the Department of Electrical Engineering, Technion, Israel, Email: shie@ee.technion.ac.il. This work was partially supported by NSF Grants CNS-1017525, CNS-0721380, EFRI-0735905, EECS-1056028, DTRA grant HDTRA 1-08-0029 and Army Research Office Grant W911NF-11-1-0265. Early versions of this paper have appeared in the Proceedings of ACM Sigmetrics, June 2012 [1], and the Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing, October 2012 [2].

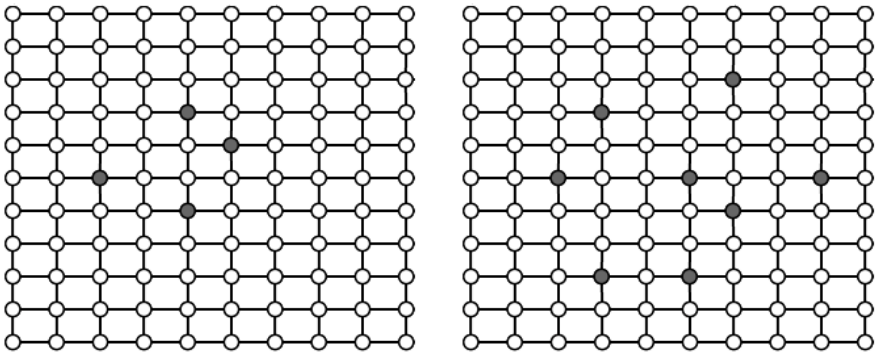

Fig. 1. Grid graphs with infected nodes darkened. The left hand graph shows a possible Type I error, with randomly sick nodes unfortunately clustered. If there are very few reporting sick nodes, such errors are impossible to rule out, hence our results impose an assumption that at least $\log n$ nodes 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.

graph, where each node has a single edge to an external infection source, this models the problem of distinguishing an epidemic spreading on G_1, from a random illness spreading according to no network structure.

This paper is about understanding when the two processes – spreading on G_1 or G_2 – are statistically distinguishable, and moreover finding sufficient conditions for when this can be done *by an efficient algorithm*. Evidently, in certain regimes, no algorithm can distinguish between the two processes. First, the graphs need to be sufficiently different. We quantify this precisely in Section II. Beyond this, certainly, if (almost) everyone is infected, or if (almost) none of those infected report, then nothing can be done. Our results are presented in terms of these two quantities: we are interested in understanding the maximum number of nodes (people/devices) that can be infected, and simultaneously the minimum number of these that actually report they are infected, so that our algorithms correctly distinguish the true spreading process, with high probability.

There are two regimes of graph topologies we consider: the setting where G_2 is a star graph – we call this the 'infection vs. random sickness' problem – and then the setting where both G_1 and G_2 exhibit nontrivial network structure – we call this the 'graph comparison' problem. For the sake of the mathematical exposition, we find it more natural to present first the graph comparison problem, and then the infection vs. random sickness problem.

In the case of the 'infection vs. random sickness' problem, there are two possible errors. 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 1 provides example of when a Type I and a Type II error might occur. The 'graph comparison' problem involves similar errors.

We provide efficiently computable algorithms to answer the above questions, and then provide sufficient conditions on the regimes where our algorithms are guaranteed to succeed, with high probability. Specifically, our main contributions are as follows:

(i) **Algorithm:** We develop efficiently computable algorithms for both problems. For inferring the causative network in the graph comparison problem, we develop what we call the Comparative Ball Algorithm. For the 'infection vs. random sickness', we develop two algorithms: the Threshold Ball Algorithm and the Threshold Tree Algorithm. These algorithms build on the intuition that infected nodes are clustered more strongly on the true causative network. If on one network, the clustering is tighter, it is more likely that it is driving the infection. We quantify clustering based on the ball radius that contains the infected nodes.

(ii) **Guarantees for General Graphs:** For the graph comparison problem, we identify two natural graph conditions that we use to give very general performance guarantees for our Comparative Ball Algorithm. The first property is called the *(a) Speed condition*; a graph satisfies this if the epidemic ball radius increases linearly in time. The second key property is called the *(b) Spread condition*; a graph satisfies this if a randomly selected collection of nodes are sufficiently spread apart, with respect to the natural metric induced by the graph. For any two graphs that satisfy both *(a)* and *(b)*, we derive upper bounds on the number of total infected nodes, and lower bounds on the number of reporting nodes, so that our Comparative Ball Algorithm is guaranteed to correctly determine the causative network (as $n \to \infty$ and with high probability).

(iii) **Grids and the Erdös-Renyi Random Graphs:** For both d-dimensional grids, and the giant component of the Erdös-Renyi random graph (with constant asymptotic average degree), and for both the graph comparison and infection vs. random sickness problem, we derive bounds on the parameters associated with the speed and spread conditions, thus, providing sufficient conditions on the regime where we can determine the causative network.

B. Related Work

The infection model we consider in this paper is the susceptible-infected (SI) model where nodes transition from *susceptible* to *infected* according to a memoryless process [5]. Much of the work on this model has focused on the predictive or analytic side, focused on characterizing the spread of the infection under various different settings. For example, [6] considers graphs with multiple mixing distances (that is, local and global spreading), while [7] considers the setting where the infected nodes are mobile. There are other approaches to modeling infection, and while interesting to extend the current ideas and analysis there, we do not consider these in the present work.

Our work, in contrast, lies on the inference side, where given (partial) information about the realization of an epidemic, the goal is to infer various properties or parameters of the spreading process. While quite different in terms motivation and goals, a few recent works have also considered epidemic inference. In [8], the authors provide a Bayesian inference approach for estimating the transmission rates of the infection. Alternatively, one can use MCMC methods to estimate the model parameters [9], [10]. A similar problem is considered in [11], [12], where, given a set of infected nodes, one seeks to determine which node is most likely to be the original source of the infection.

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 related to the problem in [13], [14]. In that work, the authors consider a hypothesis testing problem where *every node reports* an i.i.d. (zero-mean) standard normal random variable, except for a cluster of nodes reporting a normal with positive mean; the cluster of nodes with a positive mean is chosen from a pre-specified class of possible clusters. In their work, the collection of clusters are exactly specified, and could be very large (i.e., even the inclusion/exclusion of a few nodes makes it a different set). Thus a key technical complexity in [13] is to deal with potentially a very large number of sets, and leverage geometric structure (through ϵ-nets) to derive their results. Our focus is complementary – given a generative model (SI) of a spreading process on a graph and *sparse samples* of data, a key contribution is to derive the appropriate sets that need to searched over to distinguish between the hypotheses. In our case, our focus in on finding a small collection of approximate sets with a generative model for the spread (small so that a union bound works, and approximate because we have sparse number of samples where we could miss a large fraction of the samples).

On the technical side, several of our results are related to first-passage percolation. In the first-passage percolation basic formulation, there is a (lattice) graph of infinite size. For each edge, an independent random variable is generated that represents the time taken to traverse that edge. Some node is denoted as the source, and the time taken to reach another node is the minimum of the total time to traverse a path over all paths between the source and that destination. This is equivalent to an infection traveling through the network as considered here. Work has been done to analyze various characterizing properties of this percolation, such as the *shape* of the infection and the *rate* at which it spreads. In the sequel, we find particularly useful percolation results on trees [15] and lattices [16].

C. Outline of the Paper

The paper is organized as follows. In Section II, we define precisely the infection model as well our two main problems: determining the causative infection network between two graphs, and between a graph and a random sickness. Section III contains our analysis of the problem of distinguishing infections between two different graphs. We provide an efficient algorithm, and then the success criteria of this algorithm for distinguishing between epidemics on general graphs. We show that the sufficient conditions we provide are satisfied by a general class of graphs, that include two standard graph topologies, d-dimensional grids and Erdös-Renyi graphs. Then, in Section IV, we turn to the problem of distinguishing an infection from a random sickness. Recall that this is equivalent to taking one of the two graphs to be the star graph. Star graphs, however, do not have non-trivial neighborhoods, and hence the algorithm and analysis from the previous part do not immediately carry over. We develop two new algorithms for this setting, and provide success guarantees for each. We consider grids, trees and Erdös-Renyi graphs. Finally, Section V contains the simulations data for each of these problems and illustrates the empirical performance of our algorithm on these graphs. Our results demonstrate that on synthetic data, empirical performance recovers the theoretical results. We also test our algorithms on a real-world graph, and our simulations show that here too, our algorithms are quite effective.

II. The Model

We consider a collection of n nodes (vertices V) which are members of two different networks (graphs). These graphs are denoted by $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$; they share the same vertex set but have different edge sets. For example, G_1 could represent the n vertices arranged on a $d-$dimensional grid, and G_2 could be an Erdös-Renyi graph. Note that G_2 does *not* need to have qualitatively different structure from G_1; indeed G_2 could also be a $d-$dimensional grid, but with a different node-to-edge mapping.

A. Objective

We assume that the two graph topologies, G_1 and G_2 are known. At some point in time, an epidemic begins at a random node and spreads according to the edges of one of the two graphs, following the infection model described below in Section

II-B. At some snapshot in time, a small random subset of the infected nodes report their infection. From the knowledge of the graph topologies and the identity of the reporting nodes (but without knowledge of the other infected nodes) our objective is to design an algorithm that (asymptotically, as the size of the problem scales) correctly determines which graph the epidemic is spreading on.

We first study the setting where both G_1 and G_2 have non-trivial neighborhoods, and the goal is to detect which graph is responsible for spreading the epidemic; we call this the Graph Comparison Problem. We then consider the setting where G_2 is the star graph, hence modeling the problem of distinguishing an epidemic from a random illness.

B. 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 [5]). 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 – a random variable. 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 the entire graph G_i is infected.

If the graph is a star graph, then every node is incident to a single external node. Consequently, nodes become sick at the same rate, and independently of every other node. This process, then, is stochastically equivalent to a random illness, where by a given time t, each node has become sick independently with some fixed probability $\hat{q}$.

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 $q^{(n)}$. When the infection is from an epidemic on a well structured graph, $S^{(n)}$ will be a clustered, connected set of nodes. On the other hand, when the graph is a star graph (so the infection is a random sickness), $S^{(n)}$ will simply be a random 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. We consider both when t is known and when t is unknown, requiring us to estimate t from the infection size.

C. Graph Structure

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*.[1] This essentially means that given a node, v, its neighborhood in G_1 and its neighborhood in G_2 are *independent*. We make this precise by the following construction, and thus definition.

Definition 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

[1] We note that we can envision other conditions based on clustering of epidemics on the two graphs which could also serve as alternate sufficient conditions. For simplicity, we restrict ourselves to the 'random node index' condition in this paper.

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.

III. Graph Comparison Problem

The graph comparison problem consists of distinguishing the causative graph for an infection spreading on one of two *structured* graphs G_1 and G_2. We make precise what we mean by *structured graphs* below, but intuitively, both graphs have non-trivial neighborhood structure, in contrast to the star graph. This is the key technical feature that differentiates the comparative graph problem from the infection vs. random sickness problem, which we take up in Section IV. As the algorithm reveals, the key in the comparative graph problem is that, under appropriate conditions, the infection, or epidemic, is clustered on either G_1 or G_2. In the case where G_2 is the star graph, there is no notion of clustering there, so our algorithms must detect clustering vs. absence of clustering.

We turn to the details of the comparative graph problem. 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 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 need first a simple definition:

Definition 2: Given a graph $G = (V, E)$ and a subset of its nodes, $S \subseteq V$, let $\mathrm{RadiusBall}(G, S)$ denote the radius of the smallest ball that contains S.

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

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 $\left(\mathcal{G}^{(n)}, \sigma(\mathcal{G}^{(n)}), P^{(n)}\right)$. Concrete examples include the set of d-dimensional grid graphs, Erdös-Renyi graphs with bounded expected degree, d-regular trees, etc.

Definition 3: 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 any sequence $\{G^{(n)}\}$ 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 $\mathrm{RadiusBall}(G^{(n)}, S^{(n)}) < s_{\mathcal{G}} t^{(n)}$ with probability tending to 1 as n increases.
>
> **Spread Condition**: First, $\mathrm{diam}(G^{(n)}) = \Omega(\log n)$. Define $S^{(n)}$ as a set of nodes chosen uniformly at random from all nodes in $G^{(n)}$ (as in a random sickness), with $\mathrm{card}(S^{(n)}) > \beta_{\mathcal{G}} \log n$. We require that $\mathrm{RadiusBall}(G^{(n)}, S^{(n)}) > b_{\mathcal{G}} \mathrm{diam}(G^{(n)})$ 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.

Discussion: Computing the Constants. Computing the constant for the speed condition exactly, may sometimes be difficult. One simple method that is applicable to graphs with maximum degree d, upper bounds the infection process by an infection on a degree d tree. See Section III-C2 for additional detail regarding this technique. Then we can use a bound in [17] to find that a degree d tree satisfies the speed condition with speed $1.1(d+1)$. Therefore, the original graph satisfies it with the same speed. Depending on the graph structure, this bound may be weak. For our results on the graph comparison problem, knowledge of the spread and speed constants is not explicitly used in our Comparative Ball Algorithm (which we present next, in Section III-A. Rather, these constants control only the regime where our results guarantee algorithm correctness, and hence a conservative estimate would result not in a weaker algorithm, but rather in an overly pessimistic view on when the algorithm is guaranteed to perform correctly. For the Infection vs. Random Sickness problem of Section IV, however, the setting is more delicate, and conservative estimates of the speed constants may result in weaker algorithm performance. We quantify this effect, and hence the sensitivity/robustness to having loose bounds on the speed constant, in Section IV.

A. The Comparative Ball Algorithm

We provide an algorithm for the Comparative Graph Problem, called the *Comparative Ball Agorithm*, and then give a theorem with sufficient conditions guaranteeing its success. 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.

To specify our algorithm precisely, we require the following definitions. Given a graph G, a node v, and a radius r, we denote by $Ball_{v,r}(G)$ the collection of all nodes on the graph G that are at most a distance r from node v (graph distance measured by hop-count). As we have done above, we denote the diameter of the graph by $\mathrm{diam}(G)$. Given any collection of nodes S, we denote by $Ball(G,S)$ the smallest-radius ball that contains all the nodes in S, and we use $\mathrm{RadiusBall}(G,S)$ as in the definition above, to denote its corresponding radius.

Algorithm 1 Comparative Ball Algorithm

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

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

B. Main Result: General Graphs

We prove that if $\mathcal{G}_1$ and $\mathcal{G}_2$ satisfy the speed and spread conditions given above (i.e., they have finite speed and spread constants), then the Comparative Ball Algorithm can distinguish infections on any two such graphs (with probability 1, as $n \to \infty$). The speed and spread conditions turn out to be fairly mild. In Section III-C we show that, among many others, two commonly encountered, standard types of graphs satisfy these properties: $d-$dimensional grids and Erdös-Renyi graphs. More generally, the proof that Erdös-Renyi graphs satisfy the speed and spread conditions immediately implies that bounded-degree graphs also satisfy speed and spread conditions.

Our results are probabilistic, guaranteeing correct detection with probability approaching 1, as the number of nodes n in the graphs (recall the vertex sets of the two graphs are the same – it is on these nodes that the infection is spreading) scales. Therefore, our results are properly stated on a pair of families of graphs, $\{(G_1^{(n)}, G_2^{(n)})\}$, where each $G_1^{(n)}$ comes from some family $\mathcal{G}_1$, and similarly for $\mathcal{G}_2$. For notational simplicity, we refer simply to G_1 and G_2 to denote both specific graphs in this sequence, and the entire sequence as well. Thus, by $\mathrm{diam}(G_1)$ we mean the diameter of the specific graph $G_1^{(n)}$, hence this is a value that depends on n, where as the quantities $s_{\mathcal{G}_1}$, $b_{\mathcal{G}_1}$ and $\beta_{\mathcal{G}_1}$ depend on the family, and are independent of n. The infection time is $t^{(n)}$, and we require $t^{(n)} \to \infty$. As we do for the graphs, we drop the superscript for clarity and use t to denote the infection time.

Theorem 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 $\{(G_1^{(n)}, G_2^{(n)})\}$ 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 $t < b_{\mathcal{G}_1}\mathrm{diam}(G_2)/s_{\mathcal{G}_2}$, 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 $\mathrm{card}(S_{\mathrm{rep}}) > \beta_{\mathcal{G}_2} \log n$. 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, $\mathrm{RadiusBall}(G_1, S_{\mathrm{rep}}) < s_{\mathcal{G}_1} t$ and

$\mathrm{RadiusBall}(G_2, S_{\mathrm{rep}}) > b_{\mathcal{G}_2}\mathrm{diam}(G_2)$. Then the score for the first graph satisfies $\mathrm{score}(G_1) < s_{\mathcal{G}_1}t/\mathrm{diam}(G_1) < b_{\mathcal{G}_2}$ by hypothesis. Similarly, $\mathrm{score}(G_2) > b_{\mathcal{G}_2}\mathrm{diam}(G_2)/\mathrm{diam}(G_2) = b_{\mathcal{G}_2}$. Therefore, the algorithm correctly identifies an infection. ■

For the above result, note that the infection takes place on (i.e., spreads over) exactly one of the graphs G_1 or G_2 and therefore in a particular case, only one of the bounds on t is relevant to determine whether the algorithm will likely correctly determine the infection network. If the time t satisfies both bounds, then no matter which is the causative network, the algorithm performs well.

To better understand this result, and also the role of speed/spread constants and how good is an available approximation to these, it is useful to consider what it means when t is exactly at the bounds provided in the above theorem. Suppose without loss of generality that the infection is in fact spreading on G_1. Then from Theorem 3.1, the Comparative Ball Algorithm successfully identifies that the infection occured on G_1 if $t < b_{\mathcal{G}_2}\mathrm{diam}(G_1)/s_{\mathcal{G}_1}$. Suppose that in fact, $t = b_{\mathcal{G}_2}\mathrm{diam}(G_1)/s_{\mathcal{G}_1}$. Then from the speed condition, $\mathrm{RadiusBall}(G_1, S)$ may be as high as $s_{\mathcal{G}_1}t = b_{\mathcal{G}_2}\mathrm{diam}(G_1)$. That is, the infection may spread at least a constant factor of the diameter of the graph. Conservative estimates on the speed and spread constants, therefore lead us to potentially *underestimate* the critical times, after which the infection will be too diffuse for us to solve the detection problem. We emphasize again, however, that the Comparative Ball Algorithm does not take the spread and speed constants as input.

Therefore, the guarantee with respect to the infection time provided in Theorem 3.1 is strong if the bounds in the speed and spread conditions are strong. For instance, if G_1 and G_2 are both the same topology (e.g., both are grids) and the bounds are tight, then the Comparative Ball Algorithm determines the correct infection graph up to the point the infection is as spread out as a random set of nodes might be. This example is made precise and highlighted in the following corollary.

Corollary 1: Consider two identical graph families $\mathcal{G}_1$ and $\mathcal{G}_2$ of $2-$dimensional grids with independent neighborhoods. That is, $G_1^{(n)}$ is a $\sqrt{n} \times \sqrt{n}$ grid, and likewise $G_2^{(n)}$. Suppose $\log n/q < t^{(n)} < \sqrt{n}/24$. Then the Comparative Ball Algorithm correctly diagnoses the causative network with probability tending to 1, as n grows.

Proof: A loose upper bound on the speed can be obtained by looking at a tree with constant degree 4, over which the infection will spread at a speed stochastically dominating the speed on the grid. Using the method from the beginning of this section, we find both $\mathcal{G}_1$ and $\mathcal{G}_2$ satisfy the speed condition with $s_{\mathcal{G}_1} = s_{\mathcal{G}_2} = 6$ (a weak, but sufficient bound on the optimal speed). The diameter of these graphs is $\sqrt{n}$. Using the lower bound on the ball radius specified in Proposition 2 below, the graphs satisfy the spread condition with $\beta_{\mathcal{G}_1} = \beta_{\mathcal{G}_2} = 1$ and $b_{\mathcal{G}_1} = b_{\mathcal{G}_2} = 1/4$.

Let $G_1^{(n)}$ and $G_2^{(n)}$ be graphs from $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively. Since the infection spreads at least at rate 1, at time $t^{(n)}$, the expected number of reporting nodes $E[S_{\mathrm{rep}}^{(n)}] > qt^{(n)} = \log n$. Also, $t^{(n)} < \sqrt{n}/24 = b_{\mathcal{G}_1}\mathrm{diam}(G_2)/s_{\mathcal{G}_2} = b_{\mathcal{G}_2}\mathrm{diam}(G_1)/s_{\mathcal{G}_1}$. The corollary follows immediately from Theorem 3.1. ■

C. Speed and Spread Conditions: Grids and the Erdös-Renyi Graph

In this section we show that the spread and speed conditions are fairly mild, by demonstrating that they hold on two common types of graphs: the d-dimensional grid, and the Erdös-Renyi graph. The proof of the Erdös-Renyi case immediately shows that the spread and speed conditions hold for all bounded-degree graphs, which includes grids; however we give an independent proof for grids because we find it helps build intuition, but also because it makes direct use of a shape theorem from first passage percolation, which itself is useful to us in the sequel.

These two specific families of graphs are important in their own right. The d-dimensional grid graph is an example (and its spreading behavior representative) of a contact graph where the infection spreads between nodes in spatial proximity (e.g., the Bluetooth virus, human sickness). The second topology is an Erdös-Renyi graph, a random graph forming a network with low diameter. This topology models "small world networks" and captures the setting where an infection spreads over possibly 'long hops,' such as the Internet, or social networks. We show that both of these networks satisfy the spread and speed conditions, and hence that the Comparative Ball Algorithm successfully determines the causative network on these graphs. As mentioned above, our proofs for the Erdös-Renyi graphs immediately carry over to all bounded-degree graphs.

1) $d-$Dimensional Grids: Let the graph $G = \mathrm{Grid}(n, d)$ be a grid network with n nodes and dimension d, so the side length is $n^{1/d}$. We avoid edge effects by wrapping around the grid (a torus). This avoids dealing with non-essential complexities resulting from the choice of the initial source of the infection.

First, we establish limits on the speed of the infection after time t has passed. Next, we show lower bounds on the spread, i.e., the ball size needed to cover a random selection of nodes of sufficient size. Together, these show that grid graphs satisfy the speed and spread conditions.

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 [16]). 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 1 ([16]): 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

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

That is, the shape of the infected set $B(t)$ can be well-approximated by the region tB_0.

Moreover, one can show that this set B_0 is regular in that it contains an ℓ^1-ball and is contained in an ℓ^∞ ball: $\{x : \|x\|_1 \leq \mu\} \subset B_0 \subset [-\mu, \mu]^d$, where $\mu \stackrel{\triangle}{=} \sup_x \{(x, 0, ..., 0) \in B_0\}$. That is, μ is effectively the rate the infection spreads along an axis [16].[2] 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. We use this result to establish the outer bound on the shape of the infection.

Proposition 1: Let $G^{(n)} = \mathrm{Grid}(n, d)$ and let $t^{(n)}$ denote any sequence of increasing times, $t^{(n)} \to \infty$. As defined above, $S^{(n)}_{\mathrm{rep}}$, denotes the (random) subset of nodes infected by the epidemic, that report their infected status. Then there exists a constant μ such that

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

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

Proof: We drop the indexing w.r.t. n, since the context is clear. Let $\mu \stackrel{\triangle}{=} \sup_x \{(x, 0, ..., 0) \in B_0\}$ and $m = 1.1 d\mu t$. Then we must show $\mathrm{RadiusBall}(G, S_{\mathrm{rep}}) < m$ 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 $\mathrm{RadiusBall}(G, S_{\mathrm{rep}}) \geq m$. Therefore, using Lemma 1,

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

Equation 1 follows from Lemma 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, we see that $\mathrm{RadiusBall}(G, S_{\mathrm{rep}})$ satisfies the required bound with high probability. ∎

The following theorem provides a lower bound on the radius of the ball needed to cover a collection of random nodes uniformly selected from the grid. We require that the number of random nodes grows at least as $\log n$.

Proposition 2: Let $G^{(n)} = \mathrm{Grid}(n, d)$. Let $S^{(n)}$ be a collection of nodes chosen uniformly at random from $G^{(n)}$, such that $\mathrm{card}(S^{(n)}) > \log n$ for sufficiently high n. Then

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

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 $\mathrm{card}(S) > \log n$. Define $X = \mathrm{card}(S)$. 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

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

[2]The infection spreads in other directions as well, but at different rates.

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, $\mathrm{RadiusBall}(G,S) > n^{1/d}/4$ 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 (Proposition 1) and the spread condition (Proposition 2), and hence the Comparative Ball Algorithm performs well on grid graphs.

2) Erdös-Renyi Graphs and Bounded Degree 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.

We establish two results in this section. We first prove an upper bound on the ball size for an infection up to a limited time, and next, we demonstrate a lower bound on the ball size for a random collection of nodes.

Note that 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 [17] and that the number of nodes within distance m from a given node is $O(m^3c^m \log n)$. 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.

Proposition 3: 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 $S^{(n)}_{\mathrm{rep}}$ denote the random subset of nodes reached by the epidemic, that also report. Then there exists a constant C_6 such that

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

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 [17]. 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, $\tilde{S}_{\mathrm{rep}}$, 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 $\tilde{S}_{\mathrm{rep}}$ (on G). Note that the $\tilde{S}_{\mathrm{rep}}$ stochastically dominates the true infected set S. That is, for all sets T, $P(T \subset \tilde{S}_{\mathrm{rep}}) \geq P(T \subset S_{\mathrm{rep}})$.

This stochastic dominance result follows from the fact that the transition rates are universally equal or higher for the induced set. Hence, $\mathrm{RadiusBall}(G, S_{\mathrm{rep}})$ is also stochastically dominated by $\mathrm{RadiusBall}(G, \tilde{S}_{\mathrm{rep}})$, 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,

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

■

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

Proposition 4: Let $G^{(n)} = G(n,p)$, and let $S^{(n)}$ denote a collection nodes sampled uniformly at random from $G^{(n)}$, such that $\mathrm{card}(S^{(n)})$ 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 $1 - o(n^{-1})$ [19]. Then the probability of all nodes fitting inside one such ball is at most

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

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

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

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

The diameter of the giant component of an Erdös-Renyi graph is $\Theta(\log n/\log c)$ [18]. Thus, Propositions 3 and 4 establish that an Erdös-Renyi graph satisfies both the speed and spread conditions respectively.

IV. Infection vs. Random Sickness

We now turn to the setting where G_2 is the star graph. This is the problem of distinguishing an epidemic spreading on a structured graph, from a random illness affecting any given node independently of the infection status of any of its neighbors. As discussed above, and as with the graph comparison problem, distinguishing these two modes of infection becomes difficult when many nodes are infected, and when only a small fraction of the infected nodes report their infection.

For this problem, we label the structured graph G. In an infection, the sick nodes will be clustered on G. On the other hand, in the case of random illness, the infection is not guaranteed to exhibit clustering on any graph. Moreover, the star graph, of course, fails to satisfy the spread conditions. Therefore, the graph comparison algorithm and its analysis cannot suffice. Instead, we must find a test for the absence of clustering. It is most natural to use a simple threshold test for the degree of clustering. This threshold, however, itself depends on the parameters of the problem, in particular, the rate at which infected nodes report their condition (the parameter q), and the time elapsed since the epidemic began propagating, or, equivalently, the expected infection size. We consider first the setting where these parameters are explicitly known, and then turn to the setting where time (and hence, the expected infection size) is not known. In this case, we demonstrate that the time can be estimated with sufficient accuracy, based on the reporting nodes. In both cases, we assume that the reporting probability q is known. If neither the time nor (at least bounds on) q are known, the picture becomes more difficult. Moreover, in practical settings, q can likely be estimated from previous infections.

A. Threshold Algorithms

We now present two algorithms for this inference problem. As with the Comparative Ball Algorithm, these are computationally simple to run, as we demonstrate in Section V, where we run them on large-size synthetic and real-world graphs.

1) The Threshold Ball Algorithm: The Threshold Ball Algorithm is quite similar to the Comparative Ball Algorithm. Our goal is to return either INFECTION or RANDOM if the sickness is from an infection on G or a random sickness respectively. It uses a threshold parameter, that represents the degree of clustering, where here we use the radius as a proxy for this level of clustering. This threshold may be calculated from the time t if known, or estimated from the reporting sick nodes otherwise.

Algorithm 2 Threshold Ball Algorithm

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

$k \leftarrow \text{RadiusBall}(G, S_{\text{rep}})$
if $k \leq m$ **then**
  return INFECTION
else
  return RANDOM
end if

Algorithm 3 Threshold Tree Algorithm

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

$k \leftarrow \text{SizeTree}(G, S_{\text{rep}})$
if $k \leq m$ **then**
  return INFECTION
else
  return RANDOM
end if

2) The Threshold Tree Algorithm: The Threshold Tree Algorithm is similar, but rather than use ball-radius as a proxy for degree of clustering, it uses the weight of a minimum-weight spanning tree connecting all reporting infected nodes. We denote the weight of this tree on graph G for set S as SizeTree(G, S). This algorithm also requires a threshold parameter. As before, the appropriate threshold may be calculated using the time t, or estimated from the set of reporting sick nodes.

B. Summary of Results

We analyze this inference problem and in particular the performance of our two algorithms, the Threshold Ball Algorithm and the Threshold Tree Algorithm, on three types of graphs. First, we consider an infection on a d-dimensional grid. In this case, both our algorithms are able to (asymptotically) eliminate Type I and Type II error, for *up to a constant fraction of sick nodes, even when only a logarithmic fraction report sick*. Orderwise, it is clear that this is the best any algorithm (regardless of computational complexity) can hope to achieve. Our empirical results verify this performance, and also show that the Ball Algorithm outperforms the Tree Algorithm on the grid.

Next we consider tree graphs. Here we show that the Tree Algorithm can correctly discriminate between infections and random sickness for larger numbers of reporting sick nodes than the Ball Algorithm is able to handle. Finally, we analyze Erdös-Renyi graphs under two different connectivity regimes: a low-connectivity regime with edge probability close to the critical threshold when the giant component emerges; and a high connectivity regime the produces densely connected graphs. Again, we show that each algorithm can identify an infection with probabilities of error that decay to 0 as the network size goes to infinity, for appropriate ranges of parameters. Not surprisingly, the more densely connected, the more difficult it becomes to obtain a good measure of 'clustering.' Consequently, in these latter regimes, we find that one needs to intercept the sickness much earlier in order to hope to accurately discriminate between the two potential sickness mechanisms. To be more exact, in order to distinguish the type of infection on trees and Erdös-Renyi graphs, the number of infected nodes must be $O(n^{\beta})$ for some $\beta < 1$ rather than $O(n)$ as in the case for grids. The exponent depends on the algorithm and type of graph. In the Erdös-Renyi setting, we are unable to find direct analytic results to compare our two algorithms. However, in Section V we evaluate them empirically and find that the Ball Algorithm tends to perform better, despite its relative algorithmic simplicity.

C. Multidimensional Grids

Let $G^{(n)}$ be a n-node d-dimensional grid network, with side length $n^{1/d}$. As before, to avoid edge effects, we let the opposite edges of the grid connect, so that the graph forms a torus, thereby eliminating any dependence of our results on the initial source of an infection. In this section, we show that both the Threshold Ball Algorithm and the Threshold Tree Algorithm can successfully distinguish an epidemic from a random illness, even when many nodes are infected, yet very few report the infection.

We consider first the Threshold Ball Algorithm. The key result here is the Shape Theorem given in Lemma 1, which, recall, essentially says that with high probability, the *shape* of the set of infected nodes closely resembles a ball. The key quantity, then,

is the radius of this ball, i.e., the threshold the algorithm chooses in order to decide if the underlying cause of the illness is a spreading epidemic, or a random illness.

Like before, we denote the set of reporting nodes $S_{\rm rep}(n)$. We first assume that in addition to the reporting likelihood, q, we know the time $t^{(n)}$ that has elapsed since the first infection (or, equivalently, the expected size of the infection). The threshold the algorithm uses is then a simple (linear) function of $t^{(n)}$. We then give an adaptive algorithm, that estimates $t^{(n)}$ and hence the optimal threshold to use, from the number of infected nodes reporting, and the reporting likelihood. We omit the superscript n when it is clear from context.

The next result says that as long as the number of reporting sick nodes is at least $\log n$, then even if a constant fraction of nodes are infected, the Threshold Ball Algorithm can successfully distinguish the cause of the illness, provided that the time t is known. We note that this requirement on the number of reporting sick nodes is essentially tight, i.e., the result cannot be improved orderwise. We also note that this requirement on the number of reporting nodes, along with the time t, implicitly constrains the underlying parameters of the problem setup, namely q. We also prove the algorithm succeeds under similar (but slightly more restrictive) conditions when t is not known. We use μ to denote the expected rate that an infection travels along an axis on the grid, as in Section III-C. By axis, we refer to a series of consecutive edges along the same direction, i.e. a row of the grid. As remarked above, this rate μ is only a function of the dimension of the graph, since we assume the spreading rate to be normalized. While it is an input to the algorithm, we show that our results are robust in that they hold even if we only have an upper bound on μ. We quantify the degradation in the results as our upper bound weakens. We thus have the following.

Theorem 4.1: Suppose the infection spreads on a grid, and we use the Threshold Ball Algorithm (Algorithm 2). Suppose that the expected number of reporting nodes scales at least as $\log n$.

(a) Suppose t is known. Set the threshold $m = 1.1d\mu t$. Then if the expected number of infected nodes is less than $n/(4d^2)^d$,

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

In fact, for any $\kappa \geq 1$, if $m = 1.1\kappa d\mu t$, then if the expected number of infected nodes is less than $n/(4d^2\kappa)^d$, the probability of error tends to 0.

(b) Next, suppose time t is unknown. Let $X_{\rm rep}$ be the number of nodes reporting an infection, $\text{card}(S_{\rm rep})$. Use threshold $m = 1.1d^2(X_{\rm rep}\log\log n/q)^{1/d}$. Then provided that the expected number of infected nodes is less than $n/((4.4d^2)^d(\log\log n)^2)$,

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

In other words, an infection can be identified in both cases with probability approaching 1 as n tends to infinity. Note that the guarantee is nearly identical, up to the $(\log\log n)^2$ factor in the denominator; this is the price we pay for not explicitly knowing the initial time of the infection. Hence for a grid, an infection can be distinguished from a random sickness even when the infection size is $\Theta(n)$. Since this task is impossible (statistically unidentifiable) when the entire network is infected, this condition is order-wise optimal. The constant in the theorem could be improved with further work. However, in most practical circumstances, we are interested in identifying an infection while it is still fairly small, where optimizing this constant is not critical.

Proof of Theorem 4.1(a):

This proof follows along similar lines as those in Section IV-C. First consider the Type II error probability, the probability a spreading infection is labeled a random sickness. Since increasing the threshold m only decreases the Type II error probability (as more sets of reporting nodes will be labeled an infection), we need only consider the case $m = 1.1d\mu t$. The result follows from the intuitive fact that an epidemic cannot spread at a rate that is a constant factor faster than μ, its expected rate of spread. Indeed, from Proposition 1, the infection is contained in a $[-1.1\mu t, 1.1\mu t]^d$ region around the origin so

$$\text{RadiusBall}(G, S_{\rm rep}) < 1.1d\mu t,$$

with probability tending to 1 as $n \to \infty$. This is equivalent to the Type II error probability tending to 0.

Now consider the Type I error probability, namely that a random sickness is mistaken for an infection. Suppose $m = 1.1\kappa d\mu t$ for a constant $\kappa \geq 1$. From Proposition 2, since the number of reporting sick nodes, $\text{card}(S_{\rm rep})$, satisfies $\text{card}(S_{\rm rep}) > \log n$, the smallest ball that contains these random nodes satisfies, with high probability,

$$\text{RadiusBall}(G, S_{\rm rep}) > n^{1/d}/4.$$

Now we bound the time t to show $m < n^{1/d}/4$. From the shape theorem of Lemma 1, we know that if the reporting sick nodes were in fact due to an epidemic, then nearly all the nodes within the radius μt ball around the source would in fact be sick. In fact, for any $\epsilon > 0$, all the nodes with distance $(1-\epsilon)\mu t$ will be infected with high probability, so therefore at least $(2(1-\epsilon)\mu t/d)^d$ will be infected. In particular, $(1.1\mu t/d)^d$ expected nodes will be infected. Then $(1.1\mu t/d)^d < E[\text{card}(S)] < n/(4d^2\kappa)^d$ using the hypothesis. Hence

$$n^{1/d}/4 > 1.1\mu t d\kappa = m.$$

Hence, the Type I error probability also tends to 0. ■

We now use the previous result to prove that the adaptive threshold, where we use the number of reporting nodes to estimate t, also works. First we state a simple lemma to characterize the number of sick nodes.

Lemma 2: If at least X nodes are sick, then the number of reporting nodes is at least $(1-\delta)qX$ with probability at least $1-\exp(-\delta^2 qX/2)$. Similarly, the number of reporting nodes is at most $(1+\delta)qX$ with probability at least $1-\exp(-\delta^2 qX/3)$.

Proof: This is a well known Chernoff bound. ■

Theorem 4.1(b) follows from this in a simple manner.

Proof of Theorem 4.1(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). From the previous lemma, we have

$$P(\bar{X}/(\log\log n) < \text{card}(S) < \bar{X}\log\log n) \to 1.$$

Let μ be the asymptotic rate at which an infection travels, as before. Let $\epsilon > 0$. From the proof of Theorem 4.1(a), at time t, we know for $\delta > 0$

$$P(\text{card}(S) \geq (2(1-\epsilon)\mu t/d)^d) \to 1.$$

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 $m \geq 1.1d\mu t_{\max} = \frac{1.1d^2\mu(\bar{X}\log\log n)^{1/d}}{2(1-\epsilon)\mu} = \frac{1.1d^2(\bar{X}\log\log n)^{1/d}}{2(1-\epsilon)}$, then from Theorem 4.1(a), the adaptive thresholding algorithm has Type II error probability approaching 0. Since the size of the infection is concentrated around its mean from Lemma 1, $\bar{X}/(\log\log n) < E[\text{card}(S)]$ with high probability. By hypothesis, $E[\text{card}(S)] < \frac{n}{(4.4d^2)^d(\log\log n)^2}$. Therefore, we have

$$\bar{X}/(\log\log n) < \frac{n}{(4.4d^2)^d(\log\log n)^2}$$

so

$$(1.1d^2)^d\bar{X}\log\log n < \frac{n}{4^d}.$$

Taking the d^{th} root of both sides gives $n^{1/d}/4 > 1.1d^2(\bar{X}\log\log n)^{1/d} = m$. Since, as established previously, the random sickness has radius at least $n^{1/d}/4$, the sickness will be correctly diagnosed with high probability. Hence, the Type I error probability also tends to 0. ■

In the above theorem, we consider a threshold based on a speed parameter μ. Note however, that we demonstrate that thresholds a constant factor higher also work, and therefore we need only an upper bound on μ to set the threshold. This fact is necessary since the exact value of μ depends on d and may be difficult to calculate, though it can be approximated using simulations. Of course, as would be expected, the range of infection sizes for which the algorithm succeeds is decreased when larger thresholds are used, but the maximum infection size is still $\Theta(n)$ (with t known). Therefore, using the simple bounds on μ from Section III of $1 < \mu < 1.1(2d+1)$, we have the following simplified corollary for $d = 2$.

Corollary 2: Consider an infection on a $\sqrt{n}\times\sqrt{n}$ grid, and apply the Threshold Ball Algorithm with t known. Use a threshold of $m = 12.1t$. If the expected number of infected nodes is at least $\log n$ and less than $n/88^2$, the probability of error tends to 0.

Proof: Note $m = 1.1 \times 5.5 \times 2t$, with $d = 2$ and bounding μ by 5.5. The corollary follows immediately from Theorem 4.1 using the bound on μ to see $\kappa < 5.5$. ■

D. Trees

We consider the problem on tree graphs with constant branching ratios. Unlike grid graphs (and more generally, geometric graphs), these trees have exponential spreading rates, and hence manifest fundamentally different behavior. Indeed, while simple, tree graphs convey the key conceptual point of this section: the difficulty of distinguishing an epidemic from a random sickness on graphs where the infection spreads quickly. In addition, while the results do not immediately carry over, the behavior on a tree provides an intuition for the behavior of an infection on an Erdös-Renyi graph, which we cover in the next section.

Thus, let $G^{(n)}$ be a balanced tree with n nodes, constant branching ratio $c \geq 2$, and a single root node. In the case of an infection, instead of choosing a node at random to be the original source of the infection, we always choose the root of the tree. This is the most interesting case, since otherwise a constant fraction of the nodes are very far from the infection source and bottlenecked by the root node. Also, this precisely models the scenario for locally tree-like graphs, such as Erdös-Renyi graphs. We again omit the indexing on n when it is clear by context.

First we examine the performance of the Threshold Ball Algorithm on this graph. Again recall the meaning of t: it is the time at which the sicknesses are reported, and also a proxy for the expected number of infected nodes.

Theorem 4.2: Suppose G is a balanced tree with constant branching ratio and the Threshold Ball Algorithm (Algorithm 2) 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 ball algorithm with threshold $m = 1.1bt$ succeeds:

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

In general, for constant $\kappa \geq 1$, if $m = 1.1\kappa bt$ and the expected number of infected nodes is less than $n^{\beta/\kappa}$, the probability of error tends to 0.

(b) On the other hand, suppose t is not known. Define X_{rep} as $\text{card}(S_{\text{rep}})$. Then there exists constants b_2 and β, with the threshold set $m = 1.1b_2 \log(X_{\text{rep}}(\log\log n)^2/q)$, 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).
We note that as with Theorem 4.1, we quantify the cost of having only an upper bound on μ. Whereas in Theorem 4.1 the cost is linear, here we see if affects the exponent.

Proof of Theorem 4.2(a): To prove this theorem, we prove the following more general statement:

For some constant $\beta < 1$, if $qE[\text{card}(S)] = \omega(1)$ and $E[\text{card}(S)] < n^\beta$, 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 results in first passage percolation [17]. 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 [17] 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 T_v denote the time it takes the infection to reach node v. Then the *fastest-sustainable transit rate* is defined as:

$$\lim_n \inf_{p \in \delta(\Gamma_n)} \limsup_{v \in p} \frac{T_v}{\text{depth}(v)}.$$

Basic results [17] show that this quantity exists and is finite, which 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 to 0 in the size of the tree. This establishes the Type II result.

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. Again letting S_{rep} denote the number of reporting sick nodes, the probability of a Type I error is controlled by $(\frac{c^m}{n})^{S_{\text{rep}}}$ – 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 $qE[\text{card}(S)] = \omega(1)$. As shown below, $E[\text{card}(S)] > e^{(c-1)t}$, so it suffices that $t = \omega(1)$. Alternatively, if $c^m = \alpha n$ for some constant $\alpha < 1$, then we require S_{rep} to increase with n without bound with probability 1. The same condition as before is sufficient for this to be true.

Therefore, the only remaining step is to show, for some $\epsilon > 0$, $m < (1-\epsilon)\log_c n$. For $\kappa \geq 1$, we have set $m = 1.1\kappa bt$, where b is the speed of the infection. This speed can be considered the 'outer speed', the speed that the farthest node travels. Now, $E[\text{card}(S)] > e^{(c-1)t}$. Set $\beta = \frac{0.5(c-1)}{1.1b\log c}$, and suppose $E[\text{card}(S)] < n^{\beta/\kappa}$. Therefore, we solve to find

$$t < \frac{\beta \log n}{(c-1)} = \frac{0.5 \log_c n}{1.1\kappa b}.$$

From here, it is easy to see $m = 1.1\kappa bt < 0.5\log_c n$ as desired. Therefore, the Type I error also decreases to 0.

Now we conclude by showing how we can calculate $E[\text{card}(S)]$ 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 $E[\text{card}(S)] \approx c/(c-1)e^{(c-1)t}$.

■

Proof of Theorem 4.2(b): First, note that $E[\text{card}(S)]$ scales at least as $e^{(c-1)t}$ (until the infection reaches the leaves of the graph). In fact, for any fixed $\epsilon > 0$, $\text{card}(S) > e^{(c-1)t/(1+\epsilon)}$ with probability approaching 1 (for example, see [20]). Now we can proceed as in the proof of Theorem 4.1(b).

As before, let X_{rep} be the number of reporting sick nodes, and $\bar{X} = X_{\text{rep}}/q$. Then we conclude $t_{\max} = (1+\epsilon)/(c-1)\log(\bar{X}(\log\log n)^2)$. Hence, by setting $b_2 = (1+\epsilon)b/(c-1)$, we see the Type II error probability converges to 0 by Theorem 4.2(a). Using the same theorem, we see the Type I error also goes to 0. ■

Thus, the Threshold 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 infection 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 $\log_c n/b$. That is, it succeeds up to a time that is some fraction of the time for the entire network to be infected. Therefore, the algorithm is order-wise optimal in infection time, but not order-wise optimal in number of nodes infected. We clarify the previous theorem by providing the following corollary for the special case of a binary tree ($c = 2$). Note that for our rough speed upper bound, the speed b satisfies $1 \leq b \leq 1.1 \times 4 = 4.4$.

Corollary 3: Suppose $G^{(n)}$ is a binary tree, and the infection time t is known. Use the Threshold Ball Algorithm with threshold $m = 4.84t$. Then if the expected number of nodes is at least $\log n$ and less than $n^{0.149}$, the algorithm will succeed with probability of error tending to 0.

Proof: This follows from Theorem 4.2 with $\kappa \leq 4.4$. From the proof, we know $\beta/\kappa = \frac{0.5}{4.84 \log 2} > 0.149$. Therefore, since by hypothesis less than $n^{0.149}$ nodes are expected to be infected, the theorem applies. ∎

The Threshold Tree Algorithm, however, is better suited for this setting. We consider this next, and show that 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 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 also demonstrated by simulations in Section V.

We note that the threshold in the results below on the Tree Algorithm, depends on $E[\mathrm{card}(S)]$ instead of depending explicitly on t, but as discussed previously, these are essentially equivalent, and we switch between the two merely to simplify notation and the exposition.

Theorem 4.3: Consider a balanced tree G with constant branching ratio and suppose that the Threshold Tree Algorithm (Algorithm 3) 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 $m = E[\mathrm{card}(S)] \log \log n$,
$$P(\text{error}) \to 0.$$
The same result holds for $m = \kappa E[\mathrm{card}(S)] \log \log n$ for any constant $\kappa \geq 1$.

(b) Suppose t is not known. Set $X_{\mathrm{rep}} = \mathrm{card}(S_{\mathrm{rep}})$, the number of nodes reporting an infection. Use threshold $m = X_{\mathrm{rep}}/q(\log \log n)^3$. Then if for any constant $\alpha < 1$, the expected number of infected nodes is less than n^α,
$$P(\text{error}) \to 0.$$

Proof of Theorem 4.3(a): We prove the following generalization of the theorem: The Type I error probability converges to 0 for any choice of the threshold $m = o(qE[\mathrm{card}(S)] \log n)$ with $qE[\mathrm{card}(S)] = O(n^\alpha)$ for some $\alpha < 1$. In addition, the Type II error probability converges to 0 if $m = \omega(E[\mathrm{card}(S)])$.

First we prove the Type II error result (mistaking an infection for a random sickness). 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 $m = \omega(E[\mathrm{card}(S)])$ 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. By assumption, the number of reporting 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. By assumption, $E[\mathrm{card}(S_{\mathrm{rep}})] > \log n$. Let $X_{\mathrm{rep}} = \mathrm{card}(S_{\mathrm{rep}})$, and hence X_{rep} is $\omega(1)$. Choose the first level in the tree that has at least X_{rep}/c nodes. Then there are between X_{rep}/c and X_{rep} 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 X_{rep} nodes are sick, at least $X_{\mathrm{rep}}/4$ of the leaf nodes are sick and distributed independently among the at most X_{rep} subtrees. Therefore, the total number of subtrees with sick nodes at the bottom is at least $X_{\mathrm{rep}}/(8c)$. 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.

$$\begin{aligned}\mathrm{SizeTree}(S_{\mathrm{rep}}) &> \frac{X_{\mathrm{rep}}}{8c}(\log_c n - \log_c X_{\mathrm{rep}}) \\ &> X_{\mathrm{rep}} \frac{(1-\alpha)\log_c n}{8c}.\end{aligned}$$

For any $w = o(E[X_{\mathrm{rep}}])$, we know that $X_{\mathrm{rep}} > w$ with probability approaching 1 since the number of sick nodes in a random sickness is highly concentrated. Therefore, if $m = o(E[X_{\mathrm{rep}}] \log_c n)$, which is equivalent to $m = o(qE[\mathrm{card}(S)] \log n)$, the Type I error probability tends to 0. ∎

Proof of Theorem 4.3(b): Let $X_{\mathrm{rep}} = \mathrm{card}(S_{\mathrm{rep}})$. Let $\bar{X} = X_{\mathrm{rep}}/q$, roughly the expected number of total sick nodes. Then $\bar{X} \log \log n$ upper bounds $\mathrm{card}(S)$ with high probability as shown previously. In addition, like before, $\mathrm{card}(S) \log \log n > E[\mathrm{card}(S)]$ with probability approached 1. Then from Theorem 4.3(a) with $m = \bar{X}(\log \log n)^3$, we see that both probability of errors decrease to 0 asymptotically. ∎

As shown in the above theorem, the Threshold Tree Algorithm works even with most of the network infected, though not quite up to $\Theta(n)$ infected nodes like the Threshold Ball Algorithm achieved for a grid network. Interestingly, even if you heavily overestimate the threshold, the algorithm will still succeed for the same range asymptotically. However, the probability of error will still be higher in finite sized instances. Note that the threshold depends on $E[\mathrm{card}(S)]$. Using our earlier calculation, $e^{(c-1)t} < E[\mathrm{card}(S)] < \frac{c}{c-1}e^{(c-1)t}$. Therefore you can set the threshold to $m = \frac{c}{c-1}e^{(c-1)t}\log\log n$ and achieve the same asymptotic performance. For the special case of a binary tree, we provide the following corollary.

Corollary 4: Suppose $G^{(n)}$ is a binary tree and the Threshold Tree Algorithm is used with infection duration t known. Assume $q = \omega(\log\log n/\log n)$. Set threshold $m = 2e^t \log\log n$. Then if the expected infection size is greater than $\log n$ and less than $n^{0.9}$, the algorithm correctly distinguishes a random sickness from an epidemic with probability tending to 1.

Proof: This follows immediately from Theorem 4.3. ■

E. Erdös-Renyi Graphs

In this section, we consider Erdös-Renyi graphs. A notable difference in the topology of Erdös-Renyi graphs and grids is that the diameter of the former scales much more slowly (logarithmically) with graph size. That is, Erdös-Renyi graphs are more highly connected, in the sense that no two nodes are too far apart. This makes distinguishing an infection from a random sickness more difficult on these graphs.

We consider two connectivity regimes: the regime where the giant component first emerges, and each node has a constant expected number of edges, and then a much more highly connected regime, where the graph demonstrates different local properties, and discrimination between random sickness and infection is harder still.

1) Detection with Approximately Constant Average Degree: We first consider Erdös-Renyi graphs with nearly constant average degree. Define the graph $G^{(n)} = G(n,p)$ to be the graph with n nodes, where 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 (approximately) the 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 infection and random sick nodes to the giant component. We show that unlike the case of trees, our algorithms are unable to distinguish infection from random sickness when nearly a constant fraction of nodes are infected. 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. This allows us to leverage some results from the previous section, although direct translation is not possible, particularly in the analysis of our second algorithm. We will drop the index on n for clarity.

Again we note that in the next two theorems, the threshold depends on t and $E[\mathrm{card}(S)]$, respectively. As discussed, these are essentially equivalent, and the choice amounts to ease of notation and exposition.

Theorem 4.4: Suppose we use the Threshold Ball Algorithm (Algorithm 2) 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, β where, using threshold $m = 1.1bt$ and with expected number of infected nodes less than n^{β},
$$P(\text{error}) \to 0.$$
In more generality, for constant $\kappa \geq 1$, if the threshold $m = 1.1\kappa bt$ and the expected number of infected nodes is less than $n^{\beta/\kappa}$, the probability of error approaches 0.

(b) Consider unknown t. We set X_{rep} to be the number of nodes reporting an infection, $\mathrm{card}(S_{\mathrm{rep}})$. Then there exists constants b_2 and β such that for threshold $m = b_2\log(X_{\mathrm{rep}}/q(\log\log n)^2)$ and if the expected number of infected nodes is less n^{β},
$$P(\text{error}) \to 0.$$

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

Proof of Theorem 4.4(a):

Consider the Type II error probability. In this case, from Proposition 3, there is a constant b (the speed) such that, with probability converging to 1,
$$\mathrm{RadiusBall}(G, S_{\mathrm{rep}}) < 1.1bt = m.$$
Therefore, the Type II error probability tends to 0. This is of course true for larger thresholds as well.

Now we bound the Type I error probability. Consider $m = 1.1\kappa bt$ for constant $\kappa \geq 1$ and suppose $E[\mathrm{card}(S)] < n^{\beta/\kappa}$. From Proposition 4, with probability tending to 1,
$$\mathrm{RadiusBall}(G, S_{\mathrm{rep}}) > \frac{\log n}{3\log c}.$$

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, $E[\mathrm{card}(S)] \to e^{\lambda t}$. We note $\lambda > c/2$. Define $\beta = \lambda/(3 \times 1.1^2 b \log c)$. Assume $E[\mathrm{card}(S)] < n^{\beta/\kappa}$ as hypothesized. Then asymptotically with high probability,

$$\lambda t < 1.1\beta \log n/\kappa.$$

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

Proof of Theorem 4.4(b): As is shown above, $E[\mathrm{card}(S)]$ scales asymptotically as $e^{\lambda t}$ for some constant λ. In particular, for abitrary constant $\epsilon > 0$, $E[\mathrm{card}(S)] > e^{\lambda t/(1+\epsilon)}$ with probability approaching 1. Then let X_{rep} be the number of reporting sick nodes and let $\bar{X} = X_{\mathrm{rep}}/q$, so $\bar{X} \log \log n > \mathrm{card}(S)$ with probability tending to 1 as shown previously. From this, we conclude $t_{\max} = (1+\epsilon)/\lambda \log(X_{\mathrm{rep}}/q(\log \log n)^2)$. Then by Theorem 4.2(a), with $b_2 = (1+\epsilon)b/\lambda$ and $m = b_2 \log(X_{\mathrm{rep}}/q(\log \log n)^2)$, we see that the Type II error probability converges to 0. From the same theorem, the Type I error goes to 0 as well. ■

Therefore, like for tree graphs, when using the Threshold Ball Algorithm on an Erdös-Renyi graph, the maximum expected infection size is only up to n^β for some β. Since the ball algorithm does not match the infection shape as well as for grid graphs, the algorithm is not as accurate for these graphs. However, it is still order-wise optimal in terms of the infection time, since the infection grows exponentially (for sufficiently small times). From this perspective, it is a good result. We provide a corollary for the case where the graph is $G(n, 2/n)$, that is, for $c = 2$. Recall our loose bound on the speed for average degree 2 of $1.1(1+2)$.

Corollary 5: Consider an infection on graph $G(n, 2/n)$ and assume the infection time t is known. For the Threshold Ball Algorithm, use threshold $m = 3.63t$. Then if the expected number of infected nodes is at least $\log n$ and less than $n^{0.083}$, the probability of error will tend to 0.

Proof: We use $\kappa b = 3.3$, the upper bound on the speed b. Then we calculate the constant

$$\begin{aligned}\beta/\kappa &= \lambda/(3 \times 1.1^2 \kappa b \log c) \\ &> c/(2 \times 11.98 \log c) > 0.083\end{aligned}$$

so from Theorem 4.4, the probability of error tends to 0. ■

The Tree Algorithm is more complex to analyze for this graph. The more delicate analysis comes from the challenge of bounding the size of the Steiner tree for the random sickness process, needed to control Type I error.

Theorem 4.5: Suppose $G = G(n, p)$. Also suppose the Threshold Tree Algorithm (Algorithm 3) 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 $m = E[\mathrm{card}(S)] \log \log n$. For any $\alpha < 1/2$, if the expected number of infected nodes scales as less than n^α,
$$P(\mathrm{error}) \to 0.$$
The condition also guarantees asymptotically 0 error probability for thresholds $m = \kappa E[\mathrm{card}(S)] \log \log n$ for some $\kappa \geq 1$ a constant.

(b) Suppose we have unknown t. Define X_{rep} as $\mathrm{card}(S_{\mathrm{rep}})$. In this case, set the threshold to be $m = (X_{\mathrm{rep}}/q)(\log \log n)^3$. Then like before, for any constant $\alpha < 1/2$, if the expected number of infected nodes is less than n^α,
$$P(\mathrm{error}) \to 0.$$

Proof of Theorem 4.5(a): We show the following more general statement: The Type II error probability decays to 0 if the threshold is chosen as $m = \omega(E[\mathrm{card}(S)])$ and $E[\mathrm{card}(S)] = o(n)$. The Type I error probability goes to 0 when $m < kqE[\mathrm{card}(S)]$ for some value $k = o(\log(n/(qE[\mathrm{card}(S)])^2))$ and $qE[\mathrm{card}(S)] = o(\sqrt{n})$. Note these conditions are satisfied for $m = \kappa E[\mathrm{card}(S)] \log \log n$ where $\kappa \geq 1$ is a constant.

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 $\omega(E[\mathrm{card}(S)])$ when $E[\mathrm{card}(S)] = o(n)$. 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 [17]). 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 $v \in S_{\mathrm{rep}}$, let d_v denote the distance from that node to the nearest other sick node. First we show that $\sum_{v \in S_{\mathrm{rep}}} d_v \leq 2\mathrm{SizeTree}(G, S_{\mathrm{rep}})$. 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 $\sum_{v \in S_{\mathrm{rep}}} d_v$, 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 $\sum_{v\in S_{\rm rep}} d_v \leq 2\text{SizeTree}(G, S_{\rm rep})$.

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 $d = \omega(1)$, using a branching process approximation.

Let $X_{\rm rep}$ be the number of reporting sick nodes. Now assume $X_{\rm rep} = o(\sqrt{n})$. Let $\epsilon > 0$ and $l = \epsilon n/X_{\rm rep}^2$. Let $k = o(\log(n/X_{\rm rep}^2))$. 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 $(1 - X_{\rm rep}/n)^l \to e^{-\epsilon/X_{\rm rep}} \to 1 - \epsilon/X_{\rm rep}$. 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 $\text{SizeTree}(G, S_{\rm rep}) \geq (1/2)\sum d_v = (1/2)kqE[\text{card}(S)]$, with probability decaying to zero. Therefore, the Type I error probability tends to 0 as long as the threshold satisfies $m < kqE[\text{card}(S)]/2$, for $k = o(\log(n/(qE[\text{card}(S)])^2))$. Using this result, we find that the Tree Algorithm can succeed so long as $q\log(n/(qE[T])^2) = \omega(1)$. This is a complex condition, though the conditions given in the theorem are sufficient for it to be true. ■

Proof of Theorem 4.5(b): As in previous sections, we let $X_{\rm rep}$ be the number of reporting sick nodes, and define $\bar{X} = X_{\rm rep}/q$. Then as in Theorem 4.5(a), $\bar{X}\log\log n$ upper bounds $\text{card}(S)$ and $\text{card}(S)\log\log n > E[\text{card}(S)]$ with probability approaching 1. Then from Theorem 4.5(a), we see that for the specified threshold, both probability of errors decrease to 0 asymptotically. ■

Therefore, the Threshold Tree Algorithm can successfully determine an infection approximately up to when $\sqrt{n}$ nodes in the graph are infected. Like the Threshold Ball Algorithm, this is order-wise optimal in the infection time, though not in the number of nodes infected. We provide the following corollary, a counterpart to Corollary 5, to clarify the bounds, using graphs $G(n, 2/n)$, so $c = 2$. Note that from our speed upper bound $b < 3.3$, and our neighborhood size bound, $E[\text{card}(S)] < 2 \times (3.3t)^3 \times 2^{3.3t} < 71.88t^3 2^{3.3t}$.

Corollary 6: Consider graph $G^{(n)} = G(n, 2/n)$ and the Threshold Tree Algorithm with infection time t known. Choose threshold $m = 71.88t^3 2^{3.3t}\log\log n$. Then if the expected number of infected nodes is at least $\log n$ and less than $n^{0.4}$, the probability of error will tend to 0.

Proof: This follows immediately from Theorem 4.5. ■

2) Detection on Dense Graphs: Now we consider the case of an Erdös-Renyi graph with a denser set of edges. Higher connectivity means the infection spreads faster, making it more difficult to distinguish between spreading mechanisms. The performance depends critically on the exact scaling regime. We consider the regime where there exists $d \in \mathbb{Z}$ and constants $\epsilon, h \in \mathbb{R}$ such that $\epsilon < n^{d-1}p^d < h$ holds for all n as $n \to \infty$. This connectivity regime has been studied in various places – see, for example, [21] for further discussion of this scaling regime and properties of these dense graphs. The next result bounds the size of the Steiner tree on a random collection of nodes, and is the key result for bounding the Type I error.

Lemma 3: Suppose nodes become sick, independently of each other, with probability $n^{1/d}/n$, so that the expected number of reporting sick nodes is $qn^{1/d}$. Further suppose $G = G(n,p)$ whose parameters satisfy $\epsilon < \lim_{n\to\infty} n^{d-1}p^d < h$ for $d > 4$. Let Z be the size of the minimum Steiner tree connecting the reporting sick nodes. Also, let $m < (d-3)qn^{1/d}/2$ be the threshold for the Steiner tree size in the Tree Algorithm. Then Z satisfies the following probabilistic limit: $\lim_{n\to\infty} Pr(Z < m) = 0$.

Proof: Using precisely the same argument as above, we can lower-bound the size of the Steiner tree by $\sum d_v \leq 2Z$, where the sum is over all reporting sick nodes, and as before, d_v denotes the minimum distance from a reporting sick node v to the nearest other reporting sick node. To lower bound the size of this sum, we rely on a result from [21] that shows that in this scaling regime, the asymptotic distribution of the distance between two random nodes is positive on only d and $d+1$. That is, *almost all nodes* are either at distance d or $d+1$ from any given node v, and thus the distance distribution concentrates sharply around d. To put this another way, let F_d be the probability that a random node is at distance more than d from A. Then for any $\hat{d} > 1$, if $n^{\hat{d}-1}p^{\hat{d}} < h$, we have

$$\lim F_{\hat{d}} = \lim_{n\to\infty} \exp^{-n^{\hat{d}-1}p^{\hat{d}}}.$$

Recall $\lim_{n\to\infty} n^{\hat{d}-1}p^{\hat{d}}$ is bounded between ϵ and h.

Now we condition on the number of sick nodes, $\text{card}(S)$. Using the same definite as before, let $X_{\rm rep}$ be the random variable with $X_{\rm rep} = \text{card}(S_{\rm rep})$. Note $E[\text{card}(S)] = n^{1/d}$ and the expected number of reporting sick nodes $E[X_{\rm rep}] = qE[\text{card}(S)]$. We can compute the probability that the closest sick node is at distance more than $\hat{d}$ from a sick node v simply as $F_{\hat{d}}^{X_{\rm rep}} \to \exp^{-(X_{\rm rep}/n)(np)^{\hat{d}}}$. Using our scaling regime, we know that $(\epsilon n)^{1/d} < np < (hn)^{1/d}$. To simplify notation, let $h' = h^{1/d}$. We have

$$\begin{aligned} F_{d-3}^{X_{\rm rep}} &\to 1 - X_{\rm rep}/n(np)^{d-3} \\ &> 1 - X_{\rm rep}/h'nn^{(d-3)/d}. \end{aligned}$$

Using a simple union bound, we find that the probability that some reporting sick node is within distance $d-3$ of another reporting sick node is at most $X_{\rm rep}^2/h'nn^{(d-3)/d}$. Since $X_{\rm rep}$ is a binomial random variable (since we condition on $\mathrm{card}(S)$), it concentrates about its mean: for any $\epsilon'>0$, $Pr((1-\epsilon')E[X_{\rm rep}]<X_{\rm rep}<(1+\epsilon')E[X_{\rm rep}])\to 1$. When $X_{\rm rep}$ is within this range, we find that $\sum d_v > (d-3)(1-\epsilon')E[X_{\rm rep}]$ with probability at least $1-(1+\epsilon')^2h'E[X_{\rm rep}]^2n^{-3/d} > 1-Cn^{-1/d}$ for some constant C. This converges to 1 for large enough n. Thus, we have shown the desired result. ■

Now the probability of error calculations and hence the proof of correctness for the Tree Algorithm follows directly from the above.

Theorem 4.6: For graph G as above, suppose the expected number of reporting sick nodes is $qn^{1/d}$ and t is known. Then for the Threshold Tree Algorithm, the probability of a Type I error converges to 0, as long as the threshold satisfies $m<(d-3)qn^{1/d}/2$. The probability of a Type II error upper bounded by $2/(d-3-\epsilon)$ as long as the threshold satisfies $m>(d-3-\epsilon)qn^{1/d}/2$, for any value of $\epsilon>0$ such that $\epsilon+3<d$. This bound converges to 0 as $d\to\infty$.

Proof: Consider first the probability of a Type I error. This is the probability that a random sickness has a Steiner tree of size less than m. From Theorem 3, this probability converges to 0 if $E[\mathrm{card}(S)]=O(n^{1/d})$.

Second, consider the probability of a Type II error. As we have argued before, the size of this tree is no more than the total number of infected nodes, so it is sufficient to find the probability there are more than m infected nodes. The Type II error probability bound follows from using Markov's Inequality. ■

V. Simulations

The above sections give theoretical guarantees for the correctness of our algorithms, and thus characterize their ability to distinguish the cause of an illness – be it detecting one graph versus another as the causative network, or the determination that a sickness is an epidemic or a random illness. In this section, we explore these questions empirically. We validate our theoretical analysis on graphs that are generated from the ensembles we address in our theorems (grids, random graphs, trees) and then also consider epidemics on real-world graphs, and demonstrate that on these topologies as well, our algorithms perform well.

A. Graph Comparison

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 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 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 indicates that one could distinguish between G(n,p) and Grid graphs for a significantly larger range. We plan to study a systematic approach for such scalings as future work.

B. Infection vs. Random Sickness

In this section we provide simulation-based evidence of the theoretical results for the Threshold Ball Algorithm and Threshold Tree Algorithm. The simulations aim to demonstrate, in particular, three facts. First, the thresholds specified in Section IV do actually work empirically, and 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. Next, we seek to describe the relative performance

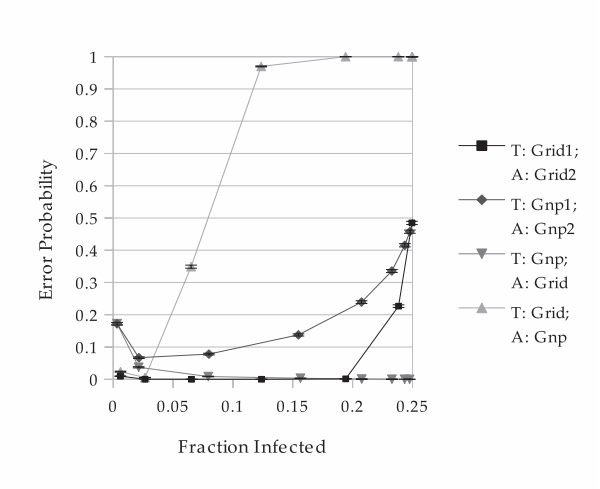

Fig. 2. This figure shows the error probability for the algorithm on pairs of standard graphs. Various (conditional) error probabilities are illustrated – 'T:' corresponds to the true network, and 'A:' corresponds to the algorithm output.

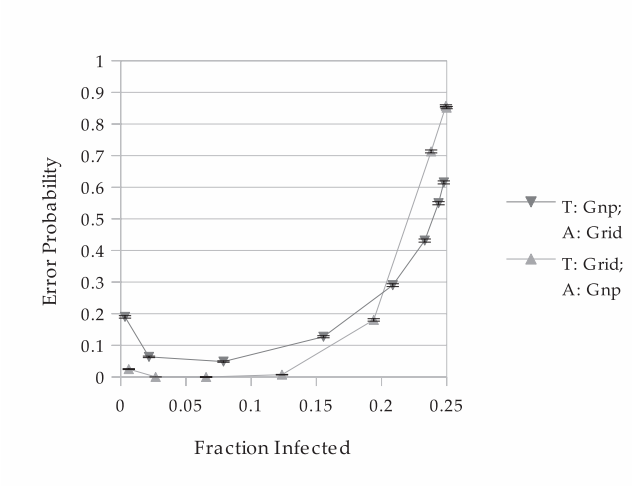

Fig. 3. This figure shows the error probability for the G(n,p) vs. Grid graphs for the scaled diameter setting (diameter of G(n,p) graph is scaled by 1.6).

of each algorithm, and show that it is as described above. Thus, we show that the Threshold Ball Algorithm outperforms the Threshold Tree Algorithm on a grid; the Threshold Tree Algorithm performs better than the Threshold Ball Algorithm on a balanced tree; and on an Erdos-Renyi graph, the performances are similar, with the Threshold Ball Algorithm performing slightly better. We accomplish this by determining the probability of error for a range infection sizes. The larger the fraction of infected nodes, the more difficult the problem becomes; hence we call an algorithm superior if it works for a larger fraction of infected nodes. The final property we illustrate is how the error probability is affected by the reporting probability q. We find that as the reporting probability increases, the error rate rapidly decreases due to the increased knowledge of the infected nodes. After reaching a minimum reporting probability, having additional nodes report their infection does not significantly reduce the error probability.

We note that to perform our simulations, it was necessary to use an approximate Steiner tree algorithm to perform the Threshold Tree Algorithm in a reasonable time frame. 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 [22]. 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 $10,000$ 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 $10,000$ 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 95% confidence.

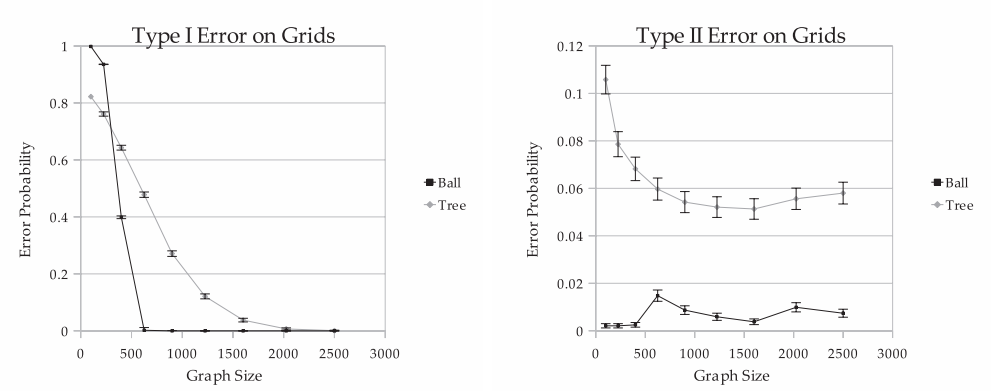

Fig. 4. Empirical Type I and Type II error probability vs graph size for grid graphs. The sample size is $10,000$ and infection size scales linearly with n.

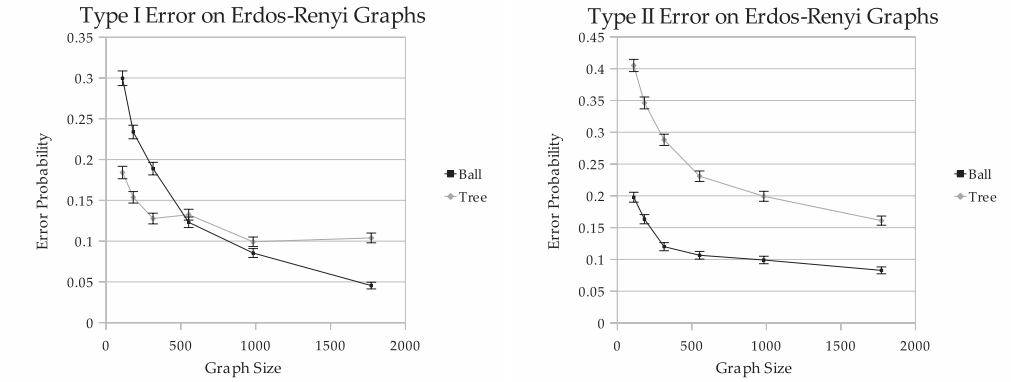

Fig. 5. Empirical Type I error probability vs graph size for graphs $G(n, 2/n)$. The sample size is $10,000$ and infection size scales orderwise as $\sqrt{n}$.

1) 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 4 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 true shape of an infection 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 5. 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 V-B2.

2) 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

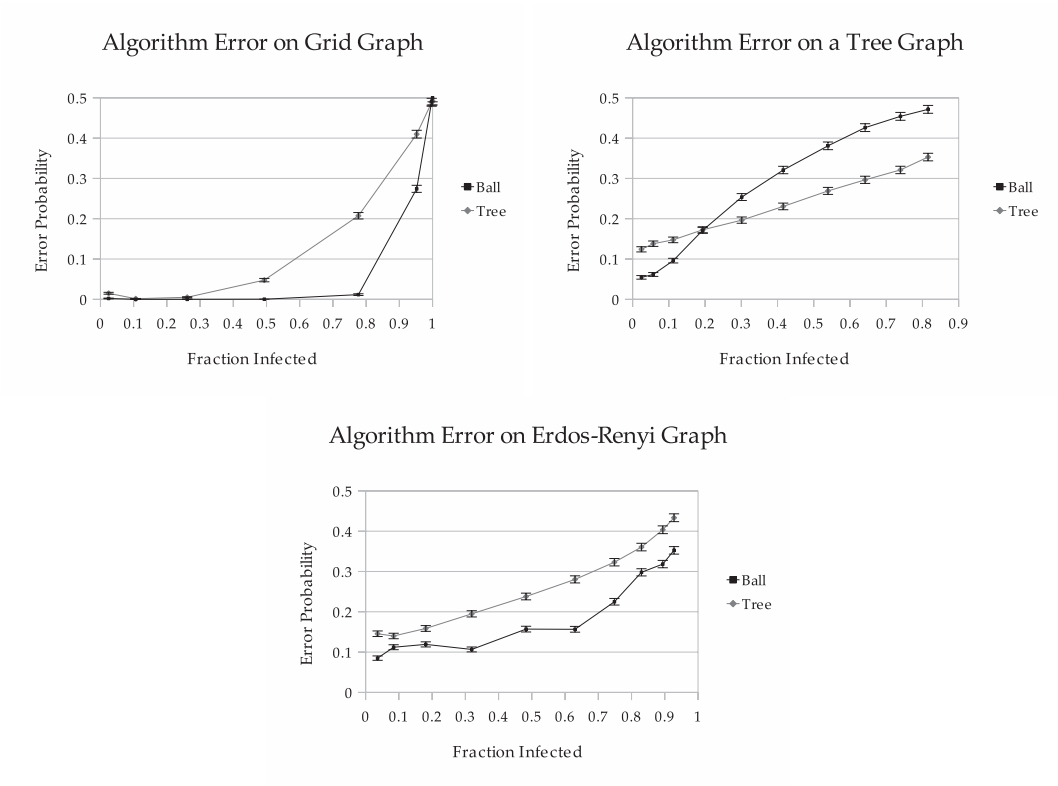

Fig. 6. This figure shows the overall error probability for each algorithm, for each of the three topologies we consider, over a range of infection sizes.

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 6 for grids, trees, and Erdös-Renyi graphs. For each of the graph topologies, we used a graph size of $n = 1,600$. The error probability is plotted against the average infection size 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 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 $n = 13,189$. The probability of error for a range infection sizes are presented in Figure 7. 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 (however, 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.

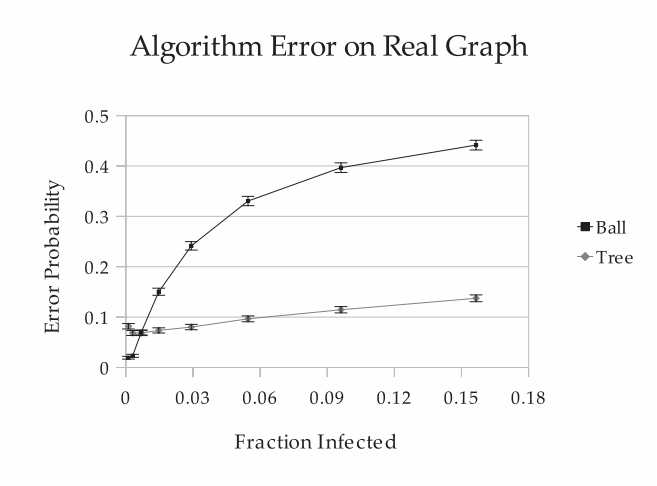

Fig. 7. This figure shows the overall error probability for each algorithm on a real world graph.

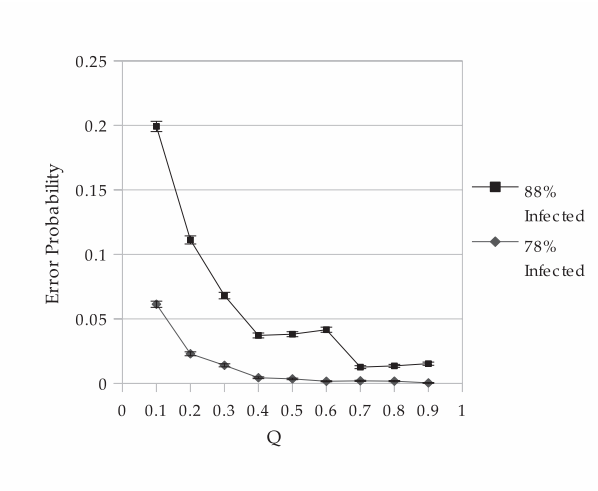

Fig. 8. The error probability of the Threshold Ball Algorithm on a grid graph (n = 1600) for a large range of reporting probabilities, with a sample size of $10,000$.

3) 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 $1,600$ 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 8.

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.

VI. Conclusions

When an infection/virus is seen spreading over a group of people/machines, one may have multiple possibile spreading regimes for the infection in mind, and want to know which the infection is most likely travelling on. We considered this problem both in the case of two well structured graphs, and in the case of comparing an infection from a random sickness. For two structured

graphs, we have shown that this is possible to do with high accuracy if the regimes are independent and satisfy two properties: 1) An infection spreading according the regime should be localized in the contact graph, and 2) A random set of nodes should be spaced far apart on the graph. When these conditions are satisfied (in the sense given in this paper), the correct spreading regime can be detected accurately with high probability by determining on which graph the infection appears to be more clustered. In addition, we have shown two standard types of graphs, grids and Erdös-Renyi graphs, satisfy these properties. In the case of comparing an infection and a random sickness, we developed two algorithms that solve the problem. We proved these algorithms do so with high probability for grids, tree, and Erdös-Renyi graph for ranges of infection sizes dependent on the graph topology. Our simulations here demonstrate the efficacy of our algorithms.

REFERENCES

[1] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, "Network forensics: random infection vs spreading epidemic," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 223–234, June 2012.

[2] ——, "On identifying the causative network of an epidemic," in *In Proceedings of 50th Annual Allerton Conference on Communication, Control, and Computing*, October 2012.

[3] Wikipedia, "HIV/AIDS — Wikipedia, the free encyclopedia," 2012, [Accessed 30-Sept-2012]. [Online]. Available: http://en.wikipedia.org/wiki/HIV/AIDS

[4] J. Cohen, "Making headway under hellacious circumstances," *SCIENCE*, vol. 313, pp. 470–473, July 2006.

[5] A. J. Ganesh, L. Massoulié, and D. F. Towsley, "The effect of network topology on the spread of epidemics," in *INFOCOM*, 2005, pp. 1455–1466.

[6] F. Ball and P. Neal, "Poisson approximation for epidemics with two levels of mixing," *The Annals of Probability*, vol. 32, no. 1B, pp. 1168–1200, 2004.

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

[8] N. Demiris and P. D. O'Neill, "Bayesian inference for epidemics with two levels of mixing," *Scandinavian Journal of Stat.*, vol. 32, pp. 265–280, 2005.

[9] G. Streftaris and G. J. Gibson, "Statistical inference for stochatic epidemic models," in *Proc. 17th International Workshop on Statistical Modeling*, 2002, pp. 609–616.

[10] N. Demiris and P. D. O'Neill, "Bayesian inference for stochastic multitype epidemics in structured populations via random graphs," *Journal of the Royal Stat. Society Series B*, vol. 67, no. 5, pp. 731–745, 2005.

[11] D. Shah and T. Zaman, "Detecting sources of computer viruses in networks: Theory and experiment," *SIGMETRICS Perform. Eval. Rev.*, vol. 86, pp. 203–214, 2010.

[12] ——, "Rumors in a network: Who's the culprit?" *IEEE Transactions on Information Theory*, vol. 57, August 2011.

[13] E. Arias-Castro, E. J. Candès, and A. Durand, "Detection of an anomalous cluster in a network," *The Annals of Statistics*, vol. 39, pp. 278–304, 2011.

[14] E. Arias-Castro, E. J. Candès, H. Helgason, and O. Zeitouni, "Searching for a trail of evidence in a maze," *The Annals of Statistics*, vol. 36, pp. 1726–1757, 2008.

[15] R. Lyons and R. Pemantle, "Random walk in a random environment and first-passage percolation on trees," *The Annals of Probability*, vol. 20, no. 1, pp. 125–136, 1992.

[16] H. Kesten, "On the speed of convergence in first-passage percolation," *The Annals of Applied Probability*, vol. 3, no. 2, pp. 296–338, Nov 1993.

[17] I. Benjamini and Y. Peres, "Tree-indexed random walks on groups and first passage percolation," *Probability Theory and Related Fields*, vol. 98, pp. 91–112, 1994.

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

[19] F. Chung and L. Lu, "The diameter of sparse random graphs," *Adv. in Appl. Math*, vol. 26, pp. 257–279, 2001.

[20] D. R. Grey, "Asymptotic behaviour of continuous time, continuous state-space branching processes," *Journal of Applied Probability*, vol. 11, no. 4, pp. 669–677, December 1974.

[21] V. D. Blondel, J.-L. Guillaume, J. M. Hendrickx, and R. M. Jungers, "Distance distribution in random graphs and application to network exploration," *Physical Review*, vol. 76, no. 066101, 2007.

[22] K. Mehlhorn, "A faster approximation algorithm for the steiner problem in graphs," *Information Processing Letters*, vol. 27, pp. 125–128, 1988.