Local Detection of Infections in Heterogeneous Networks

Chris Milling*, Constantine Caramanis†, Shie Mannor‡, Sanjay Shakkottai§
*The Univ. of Texas at Austin, Email: cmilling@utexas.edu
†The Univ. of Texas at Austin, Email: constantine@utexas.edu
‡ Technion – Israel Institute of Technology, Email: shie@ee.technion.ac.il
§The Univ. of Texas at Austin, Email: shakkott@austin.utexas.edu

***Abstract*—In many networks the operator is faced with nodes that report a potentially important phenomenon such as failures, illnesses, and viruses. The operator is faced with the question: Is it spreading over the network, or simply occurring at random? We seek to answer this question from highly noisy and incomplete data, where at a single point in time we are given a possibly very noisy subset of the infected population (including false positives and negatives). While previous work has focused on uniform spreading rates for the infection, heterogeneous graphs with unequal edge weights are more faithful models of reality. Critically, the network structure may not be fully known and modeling epidemic spread on unknown graphs relies on non-homogeneous edge (spreading) weights. Such heterogeneous graphs pose considerable challenges, requiring both algorithmic and analytical development. We develop an algorithm that can distinguish between a spreading phenomenon and a randomly occurring phenomenon while using only local information and not knowing the complete network topology and the weights. Further, we show that this algorithm can succeed even in the presence of noise, false positives and unknown graph edges.**

I. Introduction

Detecting failures and infections spreading over a network requires being able to distinguish a phenomenon that is spreading from node to node through a contact process, from a collection of random failures occurring by chance, or perhaps driven by an external source or event. The importance and the key challenges of correctly diagnosing a spreading epidemic has been observed and studied in several recent papers, including [24], [25], [16], [19], [20], [21], [10]. The epidemic can represent the spread of malware or a virus through a network, but can equally capture the spread of a human virus, or an idea, behavior or preference in a human network. The importance of correct diagnosis of a spreading phenomenon – i.e., understanding that that there is indeed a spreading epidemic, and properly detecting ¡the contact network over which it spreads – has been well documented in the history of human virus epidemiology and computer networks alike [4], [18], [26].

Key challenges addressed in the papers referenced above include working only with local information, as well as in the face of large proportions of false negatives/positives among the data. One of the requirements in all the above referenced work (see also Section I-A below), however, is *knowledge of the contact network* (at-least locally). Moreover, a related key assumption is homogeneity of the spreading network; that is, the epidemic is assumed to spread at a constant (probabilistic) rate. In real world networks, both these key assumptions typically do not hold. For starters, close relations transmit infection more readily than distant connections. More troubling is the assumption that the contact network is known. While some network connections may be known (e.g., nuclear family), others can only be estimated and should be best modeled by probabilistic connections of different strength, especially from publicly available data. For example, publicly (or relatively easily) available data may include a list of coworkers, but typically would not include statistics on pairwise daily interaction times among employees. While a model assuming a known uniform weight among all coworkers equaling the edges among family members may well be inaccurate, one that assigns weighted edges that capture whatever partial knowledge may be available, can be significantly more accurate and representative.

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

A. Related Work

There are several recent lines of work in the space of identifying statistical phenomena on large graphs. A well-studied problem is that of rumor source detection. Starting with the study in [24], there have been extensive studies on determining the source of a rumor (the epidemic process) in various epidemic contexts [25], [16], [17], [7], [13], [15], [27]. Another important line of work is that of network graph inference [23], [11], where the graph itself is learned by observing the spread of the epidemic. Related work also includes estimating the parameters of the spread [5], [6], and in estimating the fraction of nodes that are infected by the epidemic [22].

In [2], [3], the authors consider the setting in which all nodes in a network report an i.i.d Gaussian. However, in the alternative hypothesis, for a collection of sets $\mathcal{K}$, all nodes in some set $K \in \mathcal{K}$ report a Gaussian with a different mean. They

develop conditions on when the hypotheses are asymptotically separable, using a variation of the scan statistic. Their focus is on exceptionally large $|\mathcal{K}|$, necessitating the use of geometric arguments to reduce the complexity. Our problem is similar, in that we seek to characterize the structure of the infection set from an epidemic and use that structure to distinguish an epidemic from a random sickness (however, we have only sparse samples and potentially a noisy graph). We scan over a small number sets that may represent the interior of the epidemic and use the maximum infection density to determine the cause. The problem we consider here is most related to that in [19], [20], [21], where a hypothesis testing approach is used to distinguish between two unweighted graphs; where possible and appropriate, we borrow notation defined in these papers. The results in this line of work hinge upon probability concentration results for infection spread [14], [1]. A local algorithm (that looks for many hotspots of infection) for the same problem is explored in [10]. This paper extends these works to the case where the graph is weighted, so that the infection does not simply travel at the same speed between all connected nodes. We develop a new algorithm to solve this problem, which provides superior performance compared to the previous work (see also discussion in Section I-B).

B. Main Contributions and Discussion

The main setting is as follows: we consider an infection phenomenon that appears at nodes across a network. This infection either represents the collection of unrelated (independent) events that occur in the network (termed random sickness), or an epidemic, where the infection uses the edges of the network to spread, node-to-node. The goal of our work is to distinguish these two cases. We assume that we are given data at a single snapshot in time: some collection of nodes reports "infected." These nodes may contain non-infected nodes (false positives) and need not contain the complete set of infected nodes (false negatives). The spread of the epidemic is determined by the edges in the network, *and their weights*. A critical factor of our model that makes it at once significantly more broadly applicable, but also more challenging to understand and analyze, is that edge weights, and hence associated spreading time across an edge, need not be uniform across the network.

The heterogeneity in edges fundamentally changes the way we need to think about inference in this setting. From an algorithmic viewpoint, earlier work that addressed this kind of inference problem [19], [20], [21] did so by essentially detecting the boundary of the infected region – in essence, they compare the radius of a ball that 'covers' the reporting infected nodes to a fixed threshold. If the radius is small, then they report that there is an epidemic. However such a test is sub-optimal, both analytically as well as in simulations when the network edges have heterogeneity. Analytically, this occurs because estimates of the radius of a ball covering the infected nodes does not have sufficient probabilistic concentration guarantees for our inference purposes. Intuitively, this happens because with edge non-homogeneities, the 'boundary' of infection can have large protuberances (think of ray-like objects flaring out of the ball-like footprint of infected nodes). These can cause outer radius estimates to be poor. However, taking a volume inside the infected region and estimating infection *densities* turns out to be much more robust. See Figure 1 for an example.

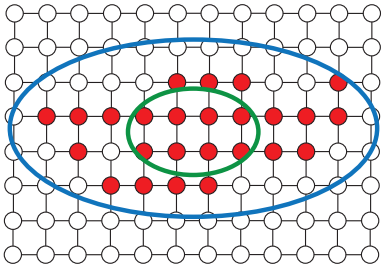

Fig. 1. A weighted grid with infected nodes colored red, where the infection travels faster in the horizontal direction. Due to the weights, the ball surrounding the infected nodes (blue) is excessively large compared to the more robust internal ball (green).

We leverage this insight in our algorithm design – we propose a Ball Density Algorithm and demonstrate its performance both via probabilistic guarantees and simulations. In fact, the maximum infection size (and time) at which our algorithm succeeds is *order-wise optimal*. In addition, we show that the Ball Density Algorithm is robust to outliers and errors in noisy data. Namely, when there are false positives in the provided set of infected nodes, or when some edges are missing from the network, the Ball Density Algorithm can succeed in similar regimes as before.

II. Model and Algorithm

This paper analyzes the epidemic regime where the epidemic travels between different pairs of nodes at different rates. Our model and notation is similar to the infection models considered in [19], [20], [21], except that due to the varying weights, the resulting infection can be highly asymmetric. This infection regime is contrasted with a random spread, where nodes exhibit sickness independently (sickness probability is the same across nodes), so that the graph structure is irrelevant. Keeping with what is now standard terminology [19], [20], [21], we term the former type of infection (where the infection travels between nodes) an *epidemic*, and the latter a *random sickness*. In addition, we consider the case when our knowledge is incomplete or inaccurate, as described above: the set of infected nodes may include both false positives and false negatives. Moreover, *some edges of the graph may be unknown*. We present an algorithm that distinguishes between an epidemic and a random sickness under these conditions.

Our results are concerned with the asymptotic performance of our algorithm, as the graph size n increases. Many of our parameters, such as the infection time, also may vary with the graph size. This is denoted by a superscript (n) (suppressed for notation clarity when the parameter is clear from context).

The following subsections (II-A, II-B and II-C) are based on [19], [20], [21], and are provided here for completeness.

Our model differs in the graph edge asymmetries and partial knowledge (II-D), and the algorithm that follows our new insights on density (II-F).

A. The Infection Process

We describe here precisely the dynamics of the spreading process. We assume the epidemic spreads over a weighted graph $G = (V, E)$ with edge weights $W = \{w_{ij} : (i, j) \in E\}$. When a node becomes infected, all incident edges start exponential clocks with mean equal to their weight; when a clock expires, the node at the other end of the edge becomes infected if not already so. Thus a lower weight between edges means it spreads faster across that edge, and a heavier weight corresponds to slower spread. We assume throughout that edge weights have universal upper and lower bounds, to avoid infinitely fast spreading. Thus, in the event of an epidemic, the infection starts spreading from a randomly selected initial node according to the above process: this is the standard susceptible-infected (SI) model [9], [8], [12]. The infection proceeds in this manner up to time $t^{(n)}$. At this time, we are able to observe the infection (with errors to be described below).

The alternative infection process is a random sickness. In this case, each nodes independently becomes sick with identical probability $p^{(n)}$. Note that in a random sickness, the structure of the graph is irrelevant, since the spread does not occur over a contact network. The most interesting and challenging case is when the random sickness and epidemic are of similar size, and hence simple counting cannot help in distinguishing one process from the other. Therefore in the sequel, we assume that p is set so that the expected sizes of both processes are equal. We use S to denote the set of all infected nodes, regardless of which process caused the infection. Note that the larger S is, the more network information gets washed out; so for instance, if S contains every node in G, then the two processes would be identical, and it would then be impossible to distinguish them. Likewise, if S is too small, for example if it contains only a single node, then the processes cannot be distinguished. We are interested in the intermediate range, when $t^{(n)}$ is large enough that a reasonable portion of the graph is infected.

B. The Reporting Process

The infection proceeds for time $t^{(n)}$ as in the previous section, either as a random sickness or an epidemic. Then the infected nodes are partially revealed. The reporting by any infected node is assumed to be independent (reporting probability is the same across infected nodes). We define this probability as $q^{(n)}$, where n is the number of nodes in G. The smaller this value $q^{(n)}$, the more difficult the problem. We typically take this probability $q^{(n)}$ to be constant or decreasing asymptotically in n.

C. False Positives

In all cases, we use the aforementioned reporting process. However, we also consider several other limitations to the knowledge about the infection process. First, along with the false negatives, we also consider the case when there are false positives. Not only do some infected nodes not report their infection, some uninfected nodes falsely report that they are infected. These false positives may be scattered randomly over the population, in the same way as a random sickness, obscuring a possible epidemic.

We model these false positives by fixing the ratio between the number of reporting infected nodes and the number of false positives. For a constant $f \geq 0$ and $|S_{\text{rep}}|$ truly reporting nodes, we set the number of false positives to be (approximately) $f\,|S_{\text{rep}}|$. For each of the $\lfloor f\,|S_{\text{rep}}| \rfloor$ false positives, we independently choose a random node from the entire graph and that node reports an infection, where repeats are allowed. We allow the chosen nodes to be in the set of infected nodes S to reduce dependency on S. Then the number of nodes that falsely report an infection may be slightly less than the specified amount, though for smaller infections, this effect is negligible. For larger infections, this model is nearly equivalent to slightly increasing the true reporting probability (since some of the "false positives" include nodes in S) and limiting the false positives to the uninfected nodes $V \setminus S$. However, the model we use allows for a simpler analysis due to the independence between the locations of the false positives and S. In addition, it ensures the reporting nodes in S have a higher density than that of the reporting nodes outside the infected set, regardless of the parameters. The set of all reporting nodes (including false positives) is denoted $\bar{S}_{\text{rep}} \supseteq S_{\text{rep}}$. Note that for $f \approx 1$, the number of truly reporting nodes and the number of false positives are almost equal, and as $f \to \infty$, nearly all the reporting nodes are false positives.

D. Unknown Edges

Knowledge of the underlying graph of an epidemic can be difficult to obtain. However, a substantial amount of the graph must be known to distinguish a random sickness from an epidemic, since it is necessary to be able to determine whether nodes are nearby or distant in order to evaluate how "clustered" the set of reporting nodes are.

In order to model the situation when the graph is not completely known, we suppose some number of edges from the true graph are missing from the graph known by the algorithm. Define the graph with the known edges (which is provided to the algorithm) as $\bar{G}$, so $E_{\bar{G}} \subseteq E_G$. There are two distinct types of edges that may be unknown. First, the edge may be "short" (in a sense to be formalized shortly), connecting two nodes that are already close to each other. For these edges, the structure of the graph is not significantly impacted by its removal. As one might expect, it is possible to tolerate a large number of unknown short edges.

On the other hand, when edges are "long", the neighborhoods of nearby nodes can change significantly. If one were to examine an epidemic spreading across $\bar{G}$ with long edges missing, the epidemic would appear to suddenly jump across the graph whenever the infection spread across these edges. The set of infected nodes would appear as multiple clusters, possibly of varying size. It is difficult to tolerate a

large number of these missing edges, since a large number of these clusters, each with possibly only a few reporting nodes, can easily appear like a random sickness. We define the length of a missing edge as the following. For a removed edge e connecting nodes i and j, we say the length of e is $\mathrm{dist}_{\bar{G}}(i,j)$, the distance between i and j on the graph with missing (unknown) edges. For a constant J, removed edges are considered short if their length is at most J. Otherwise, they are called long edges.

E. Graphs

The graphs we consider can be thought of as drawn from families of graphs, where the family reflects the graph topology, such as grids or trees. We consider a series of graphs from the same family and increasing in size, and prove that as the graphs size increases, the probability of error of our algorithm tends to 0 under some conditions. We denote a graph family by $\mathcal{G} = \{\mathcal{G}^{(n)}\}$. The set $\mathcal{G}^{(n)}$ is a collection of weighted graphs, each with n nodes, that are included in our topology. In addition, there is a (possibly trivial) probability space $\left(\mathcal{G}^{(n)}, \sigma(\mathcal{G}^{(n)}), P^{(n)}\right)$ from which the graph of size n is chosen. For each n, we choose a graph from this distribution, and then randomly choose whether the infection is from an epidemic or a random sickness, each with the same probability. We allow the infection time $t^{(n)}$ and reporting probability $q^{(n)}$ to depend on n, but the edge weights must be uniformly bounded (and away from 0) for all graph sizes.

For graph G and arbitrary nodes i and j, define $\mathrm{len}(i,j)$ as the length, in hop count, between i and j. Similarly, define $\mathrm{dist}(i,j)$ as the minimum *weighted distance* between i and j. Our algorithm considers "balls" on these graphs to be all nodes within a certain distance (this distance is weighted) from a central node. For graph G, node i and radius r, define $\mathrm{Ball}(G,a,r) = \{j \in V : \mathrm{dist}(i,j) < r\}$. We cannot distinguish random sicknesses from epidemics in arbitrary graphs, e.g., on a complete graph, as there is no topological information. As first discussed in [19], [20], [21], two conditions – a speed condition and a spread condition – are important in characterizing graphs. These essentially say that an epidemic travels at a bounded maximum and minimum speed with high probability, and the neighborhood sizes are well behaved. The key property we use that implies these properties is fairly mild: all our graphs have a constant maximum degree D. We refer to [21] for further discussion, and here give the basic definitions. We call graphs that satisfy these conditions *acceptable graphs.*

Definition 1: Consider graph family $\mathcal{G}$. This family satisfies the speed condition for minimum speed $s_{(-)}$ and maximum speed $s_{(+)}$ if, for infection time t increases with n without bound, for graph G, infection S and infection source i,

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

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

The spread condition requires that the neighborhood sizes are well behaved. Most importantly, the neighborhoods cannot increase in size without bound as the graph size increases.

Definition 2: A graph family $\mathcal{G}$ satisfies the spread condition with invertible spreading functions $b_{(-)}(r)$ and $b_{(+)}(r)$, $0 < r$ if, for graph $G^{(n)}$ drawn from this family, with probability tending to 1 the following holds for each node i and radius $r \leq \mathrm{diam}(G^{(n)})$:

$$b_{(-)}(r) < |\mathrm{Ball}(G,i,r)| < b_{(+)}(r)$$

The constants in the previous conditions can be estimated using simulations. Alternatively, if the graph has maximum degree D, bounds (possibly loose) on the speed and neighborhood sizes can be obtained. For example, with maximum degree D, the speed is at most $1.1(D+1)$ (see [1]).

F. Algorithm

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

Our algorithm is called the Ball Density Algorithm. The algorithm takes parameters m and d. The algorithm searches through the graph, and determines whether any ball of radius m has a density of reporting nodes at least d. As before, a ball of radius m is defined as all nodes within some distance m of some central node. If there is a ball with sufficient density, the reporting nodes appear sufficiently clustered and the infection is labeled an 'epidemic.' Otherwise, it is labeled a 'random sickness.' Ideally, we want d to be close to the expected density in the infected set, q. However, q may not be known. In that case, we may use a modified form of the algorithm, called the Relative Ball Density Algorithm. In this case, rather than comparing the density within the ball to a constant d, we check whether the density within the ball exceeds the density outside the ball by a factor of at least $\beta > 1$. The Ball Density Algorithm requires as input the weighted graph G (including the weights), and the set of reporting sick nodes S_{rep}. We allows some edges to be missing from G, as described in greater detail later in the paper. This algorithm is efficient, as even a naive implementation requires checking only n balls.

As mentioned earlier, this algorithm has a different approach from [19], [20], [21] – instead of looking at covering balls, we now look at densities. We remark that the Ball Density algorithm is similar to the scan statistic in [2], [3], but with a different scaling (note that the Relative Ball Density algorithm does not compare with a fixed threshold; rather an 'inside vs outside' density ratio is used).

III. Fundamental Problem and Main Results

The fundamental case is when we have access to the entire graph G and the reporting nodes S_{rep} with no false positives.

Algorithm 1 Ball Density Algorithm

Input: Graph G; Set of reporting infected nodes S_{rep};
Parameters: Density d, Radius m
Output: Epidemic or Random Sickness

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

Later sections include the case when there are false positives, and when some graph edges are unknown. We demonstrate that the Ball Density Algorithm and Relative Ball Density Algorithm can succeed in determining the type of infection with asymptotic probability 1, and characterize the range of infection sizes for which this is possible.

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

Lemma 1: Suppose in a set U of nodes, each node reports an infection independently with probability q. Let U_r be the set of reporting nodes inside U. Then for any $\delta > 0$,

$$P\left(|U_r| \geq (1+\delta)q\,|U|\right) < \exp(-\delta^2 q\,|U|\,/3)$$

and

$$P\left(|U_r| \leq (1-\delta)q\,|U|\right) < \exp(-\delta^2 q\,|U|\,/2).$$

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

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

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

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

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

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

Theorem 3: Suppose G is an acceptable graph with size n, and let $\epsilon > 0$ be a small constant. Suppose that the expected number of infected nodes is at most $(1-\epsilon)n$ and $t > b_{(-)}^{-1}(\log^2 n)/s_{(-)}$. Using the Ball Density Algorithm with parameters m satisfying $b_{(-)}^{-1}(\log^2 n) < m < s_{(-)}t$ and density $d = (1-\epsilon/2)q$, the algorithm successfully distinguishes a random sickness and an epidemic with prob. tending to 1.

Proof: First, consider a random sickness. From Theorem 1, all balls of radius m have density less than d with probability approaching 1. In this case, the algorithm corrects label the infection a random sickness. Now consider an epidemic. From Theorem 2, there is a ball of radius m contained in the epidemic with density at least d with high probability. Again, the algorithm successfully labels it an epidemic. Therefore, both the Type I and Type II error probability tend to 0. ∎

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

Theorem 4: Let G be an acceptable graph of size n and $\epsilon > 0$ be a small constant. Let $\beta > 1$. Suppose that the expected number of infected nodes is at least $\log^2 n$, and that $t < s_{(+)}^{-1} b_{(+)}^{-1}(n/(\beta+\epsilon))$. Apply the Relative Ball Density

Algorithm with radius m satisfying $b_{(-)}^{-1}(\log^2 n) < m < s_{(-)}t$ and relative factor β. Then the algorithm correctly identifies the type of infection with probability approaching 1.

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

Next, suppose the infection is an epidemic. Let $\delta = \epsilon/(\beta+\epsilon)$. Using Theorem 2, the infection contains an m radius ball with density at least $(1-\delta)q$. From Lemma 1, the density of the entire infected set is at most $(1+\delta)q$. From the speed condition, we know with high probability, the epidemic is within a ball of radius $s_{(+)}t$, containing at most $n/(\beta+\epsilon)$ nodes by assumption. No nodes outside that ball report an infection. Therefore, the external density is at most $(1+\delta)q/(\beta+\epsilon)$. After some calculation, we find the ratio of the internal and external density $(1-\delta)(\beta+\epsilon)/(1+\delta)$ is at least β. Hence, the algorithm identifies it as an epidemic with probability tending to 1. ∎

We only prove the Relative Ball Density Algorithm succeeds for time such that the *maximum* epidemic spread covers nearly up to the network size, in contrast to the time when the expected epidemic size is nearly n for the original algorithm. There may be a constant factor between these times, depending on the network topology. That is, the algorithm may only be order-wise optimal in infection time, not infection size. For some graphs, such as grids, these are the same. However, for tree graphs, it means success is only guaranteed for infection sizes up to n^γ for some $\gamma < 1$. Nevertheless, we do not need knowledge of the reporting rate for this algorithm.

IV. False Positives

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

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

Theorem 5: Consider an acceptable graph G of size n, and an infection on the graph, with false positive ratio f. Let ϵ be some small constant. Suppose the infection time is such that $t > b_{(-)}^{-1}(\log^2 n)/s_{(-)}$ and the expected infection size is less than $(1-\epsilon)n/(1+f)$. Then the Ball Density Algorithm, with parameters m in the range $b_{(-)}^{-1}(\log^2 n) < m < s_{(-)}t$ and density $d = (1-\epsilon/2)q$, determines the type of infection with probability that tends asymptotically to 1.

Proof: First, note that adding false positives only increases the density of nodes. Then clearly the Type II error probability decays to 0 as shown in Theorem 3. The remaining case is when the infection is a random sickness. As compared to the case without false positives, the density is increased by a factor of up to $(1+f)$, for an expected density of $q(1+f)E[|S|]/n$. As before, as long as d is greater than this quantity, the Type I error probability decays to 0. By assumption, $q(1+f)E[|S|] < q(1-\epsilon) < d$, so we are done. ∎

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

Theorem 6: Suppose G is a size n acceptable graph. Let $\epsilon > 0$ be a small constant, and let $\beta > 1$. Assume that the infection time t satisfies

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

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

Proof: For this theorem, the random sickness case is the easiest. The composition of false positives and the random sickness is similar to a random sickness with higher reporting rate. Just as in Theorem 4, the density inside and outside any ball is close to its expectation (and equal for both regions) and hence the Type I error probability tends to 0.

Now consider an epidemic on G. From the lower bound on t, the expected infection size is at least $\log^2 n$. Using the upper bound on t as in Theorem 4, the density of true reporting nodes over the network is at most $q(1+f)^{-1}(\beta+\epsilon)^{-1}$. Since the false positives increase this expected density by at most a factor of $(1+f)$, the outer density is at most $q/(\beta+\epsilon)$. As before, the expected density of the ball contained in the infection is q, plus additional density from the false positives. Hence, as desired, the ratio between the densities is at least β with probability tending to 1. ∎

V. Missing Edges

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

First we consider the case where there are many short edges. That is, suppose that for some constant J, each missing edge e_{ij} satisfies $\text{dist}_{\bar{G}}(i,j) \leq J$ as in Section II-D. Using this property, we find that the distance between any two nodes i and j on $\bar{G}$ increases by a factor of at most J

over the distance on G, since the length of each edge on the shortest path connecting the two nodes increases by at most that factor. Additionally, removing edges only lengthens the distance between nodes, never decreases it. By accounting for the possible increase in distance, we again show that the Ball Density Algorithm can distinguish the infection types.

Theorem 7: Let G be an acceptable graph with size n. Suppose the only unknown edges on G are short edges with length at most J. Let $\epsilon > 0$. Assume that the expected number of infected nodes is at most $(1-\epsilon)n$ and $t > b_{(-)}^{-1}(\log^2 n)/(Js_{(-)})$. For the Ball Density Algorithm, use parameters radius m and density d with $Jb_{(-)}^{-1}(\log^2 n) < m < s_{(-)}t$ and density $d = (1-\epsilon/2)q$. Then this algorithm correctly determines whether the infection is a random sickness or an epidemic with probability approaching 1.

Proof: As compared to Theorem 3, the lower bound on m is scaled up by a factor of J. The ball on $\bar{G}$ of radius m must contain at least $\log^2 n$ nodes, because it contains the ball on G of radius m/J, which by assumption contains at least $\log^2 n$ nodes. Hence, from Theorem 1, the density of a random sickness on all of these balls is no more than $(1-\epsilon/2)q$, an upper bound on the overall density. Therefore, the Type I error probability goes to 0.

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

From Theorem 7, we see that by simply increasing the minimum ball size to ensure we cover a sufficient portion of the network even with edges missing, the Ball Density Algorithm succeeds as before. Therefore, we conclude it is very tolerant of missing short edges. A similar result holds for the Relative Ball Density Algorithm.

Theorem 8: Consider an acceptable graph G of size n, and let $\epsilon > 0$ be a small constant. Set $\beta > 1$. In an infection, suppose that the number of infected nodes is at least $\log^2 n$, and that $t < s_{(+)}^{-1}b_{(+)}^{-1}(n/(\beta+\epsilon))$. Using the Relative Ball Density Algorithm with radius m in the range $Jb_{(-)}^{-1}(\log^2 n) < m < s_{(-)}t$ and relative factor β, the infection type is correctly determined with probability tending to 1.

Proof: Just as in Theorem 7, a ball of radius m on $\bar{G}$ contains at least $\log^2 n$ nodes. In addition, such a ball around the source of an epidemic is contained within the epidemic with high probability as $m < s_{(-)}t$. These are the conditions necessary for the error probability to decay to 0 as shown in Theorem 4. ∎

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

Theorem 9: Let G be an acceptable graph with size n, and suppose all but C edges are known. Let $\epsilon > 0$ be a small constant. Consider an infection with expected size at most $(1-\epsilon)n$ and duration $t > 2b_{(-)}^{-1}((C+1)\log^2 n)/s_{(-)}$. Apply the Ball Density Algorithm, setting the parameters m so that $b_{(-)}^{-1}((C+1)\log^2 n) < m < s_{(-)}t/2$ and density $d = (1-\epsilon/2)q$, with the additional requirement that the number of nodes within any considered ball must be at least $\log^2 n$. Then a random sickness and an epidemic can be distinguished with probability approaching 1.

Proof: From our additional condition, we know the balls contain $\log^2 n$ nodes. As in previous theorems, we know from Theorem 1 that the random sickness density is less than d and the Type I error probability goes to 0. Next consider an epidemic. We know the ball on $\bar{G}$ is contained within the ball on G of the same radius. Split the infection into two phases, each of length $t/2$. From the speed condition, for each node within distance $s_{(-)}t/2$ from the infection origin, the ball of radius less than $s_{(-)}t/2$ around that node is contained in the infection. Applying Theorem 2, we see that, if any such ball has at least $\log^2 n$ nodes, it has the required density.

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

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

Theorem 10: Suppose G is an acceptable graph of size n, with at most C unknown edges. Let $\epsilon > 0$ and $\beta > 1$. Assume that the expected number of infected nodes is at least $\log^2 n$ and $t < s_{(+)}^{-1}b_{(+)}^{-1}(n/(\beta+\epsilon))$. Use the Relative Ball Density Algorithm with radius m in range $b_{(-)}^{-1}((C+1)\log^2 n) < m < s_{(-)}t/2$ and relative factor β, with the additional requirement that we consider only balls containing at least $\log^2 n$ nodes. This algorithm accurately distinguishes whether the infection is a random sickness or an epidemic with probability going to 1.

Proof: From the additional algorithm condition, the ball contains at least $\log^2 n$ nodes, so in the same way as Theorem 4, we see that the Type I error probability goes to 0. For the epidemic, using the result from Theorem 9, we know there is a ball contained within the infection of radius m on $\bar{G}$ with at

least $\log^2 n$ nodes. This ball satisfies the necessary conditions for the same approach as in Theorem 4 to work. Then the Type II error probability tends to 0. ■

VI. Simulations

We now provide simulation results that confirm our analytic results. In addition, these simulations provide additional insight into how the probability of error changes with variations in the parameters. First, we compare the performance of the Ball Density Algorithm and the relative version with other algorithms. In the next section, we illustrate the effect that changing the weights of the graph has on the probability of error. Finally, we show the probability of error for various numbers of missing edges. For these simulations, we consider a grid graph where all the horizontal edges have one weight, and the vertical edges have another. Note that structure is desired in these weights. If the weights were simply random, then the infection behavior would be nearly the same as an unweighted infection with a modified edge traversal time distribution. We use graph size $n = 4900$. The reporting probability is $q = 0.25$, and no false positives or missing edges are used unless specified. The ball radius parameter is set to be the optimum value as determined empirically. For the Ball Density Algorithm, we set the density threshold to $d = 0.245$, close to q. For the Relative Ball Density Algorithm, we use a relative ratio of $\beta = 2$. After 1000 trials, the overall probability of error is determined by the average of the error probabilities of both the random sickness and epidemic cases. Other problem parameters are stated in each section below.

A. Algorithm Comparison

In this paper, we present two algorithms to distinguish random sicknesses from epidemics: the Ball Density Algorithm with fixed density and the relative density of that algorithm. For this section, we denote these the 'Density' and 'Rel. Density' algorithms respectively. Though we show both of these algorithms succeed over similar ranges of infection sizes, we have not directly compared these algorithms analytically. To compare them, we have simulated both on a grid graph, with weights in $\{1, 10\}$. In addition, there are other algorithms to consider. Our algorithms use weighted balls, but it is also possible to use balls where the distance is measured in hop counts. We denote this variation of the Relative Ball Density Algorithm as 'Rel. Density with Hops.' Another possible algorithm is the weighted variant of the Ball Algorithm as presented in [19]. In this algorithm, the infection is labeled an epidemic if all the infected nodes can be contained within a (edge weighted) ball of a specified radius. Note that this algorithm is (nearly) equivalent to the Relative Ball Density Algorithm with infinite relative factor β. This algorithm is denoted 'Ball', and the version where hop counts are used for the distance is denoted 'Ball with Hops.'

The simulation results are presented in Figure 2 (the 'Ball with Hops' algorithm is omitted for clarity). There is a clear ordering of the algorithm performance. From best to worst, the algorithms are 'Rel. Density', 'Ball', 'Rel. Density with Hops', 'Ball with Hops' and finally 'Density.' For example, when around 89% of network is infected, the error probabilities are approximately 1%, 2%, 3%, 4%, and 5% respectively. Then we see that on this graph, the Relative Ball Density Algorithm performs better than the other algorithms, including the Ball Algorithm from prior work. We also see that including the effects of the weights in the graph is necessary for optimal performance. The regular Ball Density Algorithm lags behind, partially due to the inability to adapt as well to larger infection sizes, enabling a random sickness to more easily exceed the specified density threshold.

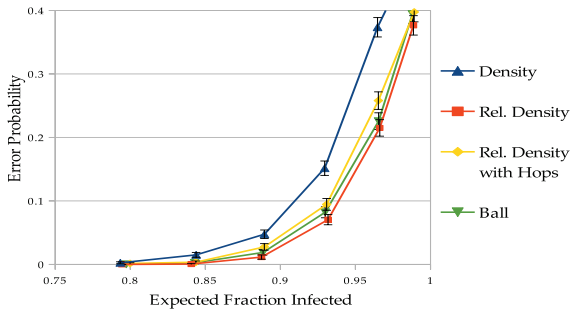

Fig. 2. This figure shows the overall error probability for a grid graph ($n = 4900$) with the weights on horizontal edges of 1, and on vertical edges of 10 over a range of infections sizes for different algorithms.

B. Weights

As the difference in edge weights increases, the more skewed the infection becomes towards the smaller edge weights. To examine how tolerant our algorithm is towards different edge weights, we simulated the Relative Ball Density Algorithm on a grid, fixing the weight of the horizontal edges at 1 and varying the weights of the other edges. The probability of error is shown in Figure 3. As the figure shows, though the error probability increases slightly as the weights increase, the performance of the algorithm is very similar regardless of edge weight distribution on this graph. Then we conclude the Relative Ball Density Algorithm appropriately adapts to the weight distribution in this case.

C. Unknown Edges

One key feature of our algorithm is that it is robust against unknown edges. We simulated the Relative Ball Density Algorithm for various numbers of missing edges to confirm this analytic result. The simulations use a grid graph with edge weights 1 and 10, but add a variable number of long distance edges between nodes chosen uniformly at random from the grid, each with weight 1. These edges are unknown to the algorithm, causing an epidemic to appear as multiple clusters. The probability of error for different numbers of these missing edges is shown in Figure 4. Note that due to this construction, the epidemic also spreads somewhat faster the more missing edges there are. As the figure shows, though

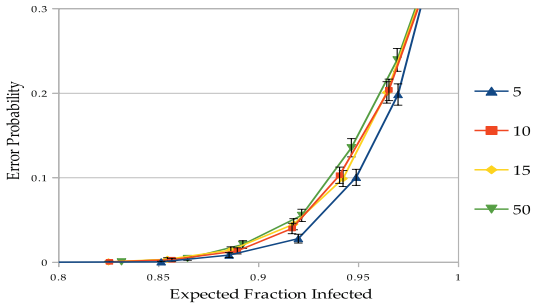

Fig. 3. This figure illustrates the overall error probability for the Relative Ball Density Algorithm on a grid of size $n = 4900$. The edge weights on the horizontal edges are 1, and the weights on the vertical edges are given in the legend.

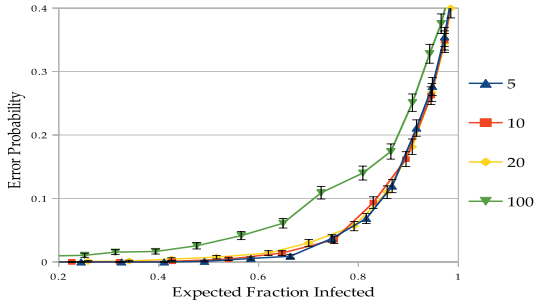

Fig. 4. This figure presents the overall error prob. when using the Relative Ball Density Algorithm (grid graph) with $n = 4900$ with horizontal and vertical edge weights of 1 and 10; additional unknown random edges of weight 1, for various numbers of missing edges.

the error probability increases significantly at smaller infection sizes compared to the case without missing edges, it is still low until a majority of the network is infected. In addition, the error probability increases very slowly as the number of missing edges increases.

VII. Conclusion

We develop the Ball Density Algorithm to distinguish between random sickness and graph-based spread in a variety of noisy network settings. We demonstrate that it succeeds with high probability for a large range of infection sizes, nearly up to when the entire network is infected. In addition, we show it is robust, able to handle large numbers of false positives and unknown edges. When the reporting probability is unknown, the Relative Ball Density Algorithm can be used, and identifies the infection mechanism under similar conditions.

VIII. Acknowledgments

This work was partially supported by NSF Grants CNS-1017525, CNS-0721380, CNS-1320175, EFRI-0735905, EECS-1056028, DTRA grant HDTRA 1-08-0029 and ARO Grants W911NF-11-1-0265 and W911NF-14-1-0387.

References

[1] I. Benjamini and Y. Peres. Tree-indexed random walks on groups and first passage percolation. *Prob. Theory and Rel. Fields*, 98:91–112, 1994.

[2] 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.

[3] 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.

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

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

[6] N. Demiris and P. D. O'Neill. Bayesian inference for stochastic multi-type epidemics in structured populations via random graphs. *Journal of the Royal Stat. Society Series B*, 67(5):731–745, 2005.

[7] W. Dong and W. Zhang and C. W. Tan. Rooting out the rumor culprit from suspects. *Proc. of IEEE Int. Symp. on Information Theory*, 2013.

[8] R. Durrett. *Random Graph Dynamics*. Cambridge Univ. Press, 2007.

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

[10] E. Meirom, C. Milling, C. Caramanis, S. Mannor, A. Orda, S. Shakkottai Localized epidemic detection in networks with overwhelming noise. arXiv Technical Report, arXiv:1402.1263, 2014.

[11] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM Trans. Knowl. Discov. Data*, 5(4):21:1–21:37, Feb. 2012.

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

[13] N. Karamchandani and M. Franceschetti. Rumor source detection under probabilistic sampling. *Proceedings of IEEE International Symposium on Information Theory*, 2013.

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

[15] A. Lokhov and M . Mezard and H. Ohta and L. Zdeborova. Inferring the origin of an epidemic with dynamic message-passing algorithm. arXiv:1303.5315, 2013.

[16] W. Luo and W. P. Tay. Identifying infection sources in large tree networks. In *9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 281–289, June 2012.

[17] W. Luo and W. P. Tay. Finding an infection source under the SIS model. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2013.

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

[19] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai. Network forensics: random infection vs spreading epidemic. *SIGMETRICS Perform. Eval. Rev.*, 40(1):223–234, June 2012. Longer version (titled "Distinguishing Infections on Different Graph Topologies") available as UT Austin Technical Report, 2014.

[20] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai. On identifying the causative network of an epidemic. In *Proc. of 50th Annual Allerton Conf. on Communication, Control, and Computing*, October 2012.

[21] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai. Detecting epidemics using highly noisy data. In *Proc. of the 14th ACM Int. Symposium on Mobile Ad Hoc Networking and Computing*, 2013.

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

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

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

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

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

[27] K. Zhu and L. Ying. Information Source Detection in the SIR Model: A Sample Path Based Approach arXiv:1206.5421, http://arxiv.org/abs/1206.5421.