Near-Optimal Mapping of Network States using Probes

Bijaya Adhikari* Pavan Rangudu† B. Aditya Prakash* Anil Vullikanti*†

Abstract

In many applications, such as the Internet and infrastructure networks, nodes fail or get congested dynamically. We study the problem of inferring all the failed nodes, when only a sample of the failures is known, and there exist correlations between node failures/congestion in networks. We formalize this as the GraphStateInf problem, using the Minimum Description Length (MDL) principle. We propose the GraphMap algorithm for minimizing the MDL cost, and show that it gives an additive approximation, relative to the optimal. We evaluate our methods on synthetic and real datasets, which includes one from WAZE which gives traffic incident reports for the city of Boston. We find that our method gives promising results in recovering the missing failures.

1 Introduction

Most network applications assume the network state is static and is known ahead of time. This is not true in practice, and networks are inferred by indirect measurements, e.g., as in the case of the Internet router/AS level graphs, which are constructed using trace-routes, e.g., [5], or biological networks, which are inferred by experimental correlations, e.g., [15]. Further, network elements can fail dynamically, or their state may change with time. For instance, links in the Internet router network or the transportation network can get congested or fail. Reconstructing the network topology dynamically and inferring network states in such settings are fundamental problems. Such problems have been studied as part of the area of "network tomography", especially in communication networks, e.g., [8, 19, 10]. Such networks are not publicly accessible, and indirect probes are the only means of obtaining information; examples of probes include queries of the activity states of selected nodes and end-to-end measurements of delays between selected pairs of nodes. These become very challenging problems, and all prior work in this direction in network tomography has been focused on simple models of *independent* link failures and delays, e.g., with exponentially distributed probabilities [10].

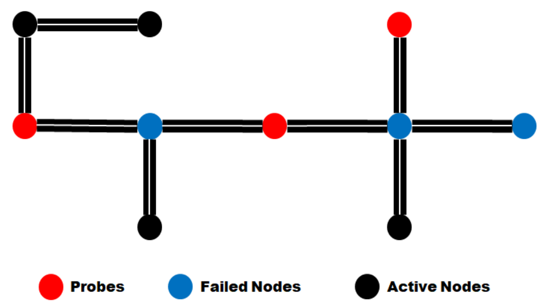


Figure 1: An example road network, where multiple intersections have failed. Given a partial probe of failed nodes (red), our method infers other nodes which have failed (blue).

In many settings, such as disaster events in infrastructure networks, however, failures might be spatially correlated, as in [3, 1, 14]. For instance, in the model considered in [1], where authors study vulnerability of networks, the probability that a node j fails decays with the distance from a source s. This motivates the problem of inferring the network states under such spatial correlations, which is the focus of our paper. For example in a toy road network shown in Figure 1, given a partial probe of failed nodes (shown in red), can we infer other nodes which have failed as well (shown in blue)?

A closely related topic is the inference of the source of an infection and other missing infections in the case of epidemic spread on networks—these are typically modeled as SI/SIR processes (see [7, 9] for an introduction to epidemic models), where the infection spreads from one node to its neighbors with some probability [11, 18, 17]. There has been some work in missing infection problems in this direction e.g. [11, 18] develop algorithms for the SI process. Intuitively, these link-based methods do not seem to directly work for our problem, due to the difference in propagation process of failures/infections and epidemiological processes. For example, if a node in a fully-connected clique is infected in an epidemic process, all the remaining nodes are highly likely to get infected; which is not necessarily the case in infrastructure net-

*Department of Computer Science, Virginia Tech. Email: {bijaya, badityap}@cs.vt.edu.

†NDSSL, Biocomplexity Institute, Virginia Tech. Email: {rangudu, vsakumar}@vt.edu.

works, say road networks, where failures are more spatially correlated.

Our main contributions are summarized below.

1. We develop a novel robust formulation for the GraphStateInf problem using the Minimum Description Length (MDL) principle [6, 12], which takes correlated failures into account and aims to find the missing failures which best explain the given data. We present GraphMap, an algorithm for inferring the missing failed nodes, given a sample probe of the failures. We prove that the MDL cost of the solution computed by GraphMap is within an additive approximation of the minimum cost MDL solution. Typically, approaches using MDL are based on heuristics and getting bounds is non trivial as MDL cost functions are not convex. To the best of our knowledge, our algorithm is the first to obtain rigorous bounds on the objective value among MDL based approaches for network inference.

2. We evaluate our results on different kinds of synthetic and real datasets, namely, one week's worth of traffic status and incident reports from WAZE for the city of Boston, and electric disturbance events in the power grid. We study the precision, recall and F1-score for GraphMap, compared with a baseline. We observe that our algorithm is quite effective in inferring unknown/missing failures in the network, and has lower MDL cost than the baseline.

The rest of the paper is organized in the following way: we formulate our problem in Section 2 and propose our methods in Section 3. In Section 4, we describe our datasets in detail and show experimental results. We then present the related work and conclusions in Sections 5 and 6 respectively.

2 Our Problem Formulation

We are given an undirected graph $G(V, E)$ representing an infrastructure network. We assume there is an initial failure at a node, referred to as the seed node, which causes other nodes to fail. Further, a subset $\mathcal{Q} \subseteq I$ of the actual failed nodes are assumed to be known. The objective in the GraphStateInf problem is to infer all the missing failures.

2.1 Failure Model Next, we will discuss the failure model we use to describe the failures in the given network G. This model is motivated by the geographically correlated failure model introduced by Agarwal et al [1], to capture failures in infrastructure networks due to large scale disasters. In such events, there is an initial localized failure, which causes other nodes to fail with some probability that decays with the distance from the source.

Following [1], we assume there is an initial single 'seed' node s and all the failures I in G are caused due to the influence of that seed node. We assume a discrete probability distribution function $p_s : V \to [0, 1]$ that gives the probability of each node $v \in V$ being a seed and conditional failure probability distribution function $F : V \times V \to [0, 1]$ that gives the failure probability of a node $v \in V$ given a seed node s. Note that the $p_s(v)$ is the probability of v being the only seed, i.e, $\sum_{v \in V} p_s(v) = 1$. These probability distributions are precomputed from historically observed failures. We assume that the conditional failure probabilities given by F are independent i.e., for-all $v_1, v_2 \in V$ and $v_1 \neq v_2$,

$$F(v_1 \cap v_2 \mid s) = F(v_1 \mid s)F(v_2 \mid s) \tag{2.1}$$

2.2 Probes Based on our model given above, we assume that some seed failed causing multiple correlated failures across the network G. The final set of true failures is represented by $I \subseteq V$. Further, we are also given a set of failed nodes represented by $\mathcal{Q} \subseteq I$, which we will refer to as probes in rest of this paper. In reality, the probes $\mathcal{Q}$ are failed nodes that are observed, and this is typically a random process. For ease of modeling, we assume that the probes are sampled uniformly at random from the true failure set I with probability γ.

2.3 MDL We formulate our problem using the Minimum Description Length (MDL) principle [6]. We will use two-part MDL, or the sender-receiver framework. Our goal here is to transmit the given set of probes $\mathcal{Q}$ from sender to receiver by assuming that both of them know the layout of the network G. We do this by identifying the model that best describes the given data in terms of a formal objective or cost function. This cost function consists of two parts:

1. Model cost that signifies the complexity of the selected model that explains the failures in the network; and

2. Data cost that represents the cost of observing the given probe data $\mathcal{Q}$ given the model.

More formally, given a set of models $\mathcal{M}$, MDL identifies the best model M^* as the model that minimizes $\mathcal{L}(M) + \mathcal{L}(\mathcal{D} \mid M)$, in which $\mathcal{L}(M)$ is the model-cost (length in bits to describe model M), and $\mathcal{L}(\mathcal{D}|M)$ is the data-cost (the length in bits to describe the data using M). Note that the data we need to describe in our situation is the probes set $\mathcal{Q}$ (and not the true failures set I). Next we describe the model space and the model and data cost, which we will optimize.

2.4 Model Space and Cost Model Space: The most natural model for our problem would have been $\mathcal{M} = (s, I)$ (the source $s \in V$ and the full failure set $I \subseteq V$), as it directly mimics the generative process of the failure model. However, this model has several disadvantages. Firstly, note that this model space is intuitively 'fragile': small changes in I or the source s can have vastly different costs. Hence due to data sparsity, we expect it would be very hard to learn the true source which generated the failures—indeed, in our experiments, we find that it was not robustly learning the true source. As a result, we also found that the solutions with minimum MDL cost were finding very few missing failed nodes (i.e. $I - Q$), leading to a very low recall. How to design a better model space for our problem? We observe that this model intuitively tries to explain 'more' than what is needed. Note that while our original goal was to map the missing *failures* only, this approach tries to explain the source as well as the set of failures. Hence we adopt a different approach, where we try to marginalize over the seeds, and focus only on the failures. This makes our model space more robust too. This motivates our proposed model, which consists of three components, namely, $\mathcal{M} = (|\mathcal{Q}|, |I|, I)$. In other words, we send the size of probes $\mathcal{Q}$, the size of true failure set I, and then identify the set itself. After sending the model, we will then identify the actual probes set Q as the data.

Model cost: The MDL model cost, $\mathcal{L}(|\mathcal{Q}|, |I|, I)$ has three components

$$\mathcal{L}(|\mathcal{Q}|, |I|, I) = \mathcal{L}(|\mathcal{Q}|) + \mathcal{L}\big(|I| \mid |\mathcal{Q}|\big) + \mathcal{L}\big(I \mid |\mathcal{Q}|, |I|\big).$$

We derive these below. We have $\mathcal{L}(|\mathcal{Q}|) = -\log\Big(Pr(|\mathcal{Q}|)\Big)$, by using the *Shannon-Fano* code to encode $|\mathcal{Q}|$. Similarly we have:

$$\begin{aligned} \mathcal{L}\big(|I| \mid |\mathcal{Q}|\big) &= -\log\Big(Pr(|I| \mid |\mathcal{Q}|)\Big) \\ &= -\log\Big(\frac{Pr\big(|\mathcal{Q}| \mid |I|\big)Pr(|I|)}{Pr(|\mathcal{Q}|)}\Big) \end{aligned} \tag{2.2}$$

From the sampling assumption for $\mathcal{Q}$, we can get:

$$Pr\big(|\mathcal{Q}| \mid |I|\big) = \binom{|I|}{|\mathcal{Q}|}\gamma^{|\mathcal{Q}|}(1-\gamma)^{|I \setminus \mathcal{Q}|} \tag{2.3}$$

Also observe that:

(2.4)
$$\begin{aligned} \mathcal{L}\big(I \mid |\mathcal{Q}|, |I|\big) &= -\log\Big(Pr\big(I \mid |\mathcal{Q}|, |I|\big)\Big) \\ &= -\log\Big(Pr\big(I \mid |I|\big)\Big) = -\log\Big(\frac{Pr(I)}{Pr(|I|)}\Big) \end{aligned}$$

Combining all of the above, the complete model cost is:

$$\begin{aligned} &\mathcal{L}(|\mathcal{Q}|, |I|, I) \\ =&\mathcal{L}(|\mathcal{Q}|) + \mathcal{L}\big(|I| \mid |\mathcal{Q}|\big) + \mathcal{L}\big(I \mid |\mathcal{Q}|, |I|\big) \\ =&-\log\Big(Pr(|\mathcal{Q}|)\Big) - \log\Big(\frac{Pr\big(|\mathcal{Q}| \mid |I|\big)Pr(|I|)}{Pr(|\mathcal{Q}|)}\Big) \\ &-\log\Big(\frac{Pr(I)}{Pr(|I|)}\Big) \\ =&-\log\Big(Pr\big(|\mathcal{Q}| \mid |I|\big)\Big) - \log\Big(Pr(|I|)\Big) - \log\Big(\frac{Pr(I)}{Pr(|I|)}\Big) \\ =&-\log\Big(\binom{|I|}{|\mathcal{Q}|}\gamma^{|\mathcal{Q}|}(1-\gamma)^{|I\setminus\mathcal{Q}|}\Big) \\ &-\log\Big(\sum_{s\in V} Pr(I \mid s)p(s)\Big) \\ =&-\log\binom{|I|}{|\mathcal{Q}|} - |\mathcal{Q}|\log(\gamma) - (|I| - |\mathcal{Q}|)\log(1-\gamma) \end{aligned}$$

(2.5)
$$-\log\Big(\sum_{s\in V} p_s(s)\prod_{v\in I} F(v \mid s)\prod_{v'\notin I}\Big(1 - F(v' \mid s)\Big)\Big)$$

2.5 Data Cost Now, we need to describe the given input probes $\mathcal{Q}$ in terms of the model. Given model $\mathcal{M} = (|\mathcal{Q}|, |I|, I)$, describing $\mathcal{Q}$ is the same as specifying the adjustments that needs to be applied to the failure set I in the model to reach $\mathcal{Q}$, which can be done by describing the following sets:

1. Unobserved failures i.e., $\mathcal{Q}^+ = I \setminus \mathcal{Q}$
2. Observation errors i.e., $\mathcal{Q}^- = \mathcal{Q} \setminus I$

In this paper, we assume that there are no observation errors, i.e., $\mathcal{Q}^- = \emptyset$ (as $\mathcal{Q} \subseteq I$). According to the sampling assumption we have, $\mathcal{Q}$ is sampled uniformly at random from I with probability γ. This implies that $\mathcal{Q}^+ = I \setminus \mathcal{Q}$ is sampled from I with uniform probability $(1-\gamma)$. Hence we can compute the probability of seeing a set $\mathcal{Q}^+$ when sampled from I as follows

$$Pr(\mathcal{Q}^+|I) = \gamma^{|\mathcal{Q}|}(1-\gamma)^{|\mathcal{Q}^+|} \tag{2.6}$$

Now, using this probability distribution of observing the set $\mathcal{Q}^+$ given the failure set I we can compute the optimal number of bits required to transmit $\mathcal{Q}^+$ encoded in terms of model M as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{Q}^+|I) &= -\log\Big(\gamma^{|\mathcal{Q}|}(1-\gamma)^{|\mathcal{Q}^+|}\Big) \\ &= -|\mathcal{Q}|\log(\gamma) - (|I| - |\mathcal{Q}|)\log(1-\gamma) \end{aligned} \tag{2.7}$$

2.6 Our Formal Problem Putting it all together, we can state our formal problem GraphStateInf as following:

GraphStateInf: *Given an undirected graph $G(V, E)$, where node failures taken place in the network as per the model described in Section-2.1, and a set of observed failures $\mathcal{Q} \subseteq V$, which are sampled independently from the true failure set I^* with a uniform probability γ, find the complete set of failures $I \subseteq V$ by minimizing the MDL cost function $\mathcal{L}\big(|\mathcal{Q}|, |I|, I, \mathcal{Q}\big)$ given by*

$$
\begin{aligned}
&\mathcal{L}\big(|\mathcal{Q}|, |I|, I, \mathcal{Q}\big) \\
&= \mathcal{L}(|\mathcal{Q}|) + \mathcal{L}\big(|I| \mid |\mathcal{Q}|\big) + \mathcal{L}\big(I \mid |\mathcal{Q}|, |I|\big) + \mathcal{L}\big(\mathcal{Q} \mid |\mathcal{Q}|, |I|, I\big) \\
&= -\log\Big(\sum_{s \in V} p_s(s) \prod_{v \in I} F(v \mid s) \prod_{v' \notin I} \Big(1 - F(v' \mid s)\Big)\Big) \\
&\quad - 2|\mathcal{Q}|\log(\gamma) - 2(|I| - |\mathcal{Q}|)\log(1-\gamma) - \log\binom{|I|}{|\mathcal{Q}|}
\end{aligned} \tag{2.8}
$$

where $p_s(s)$ is the seed probability of s and $F(v \mid s)$ is the failure probability of node v given seed node s.

3 Proposed Methods

Clearly the search space for the problem is large, and there exists no trivial structure for fast search. Indeed, typically MDL-based optimization problems are very challenging. We now describe two approaches for finding solutions with low MDL cost. The first, LocalSearch, incrementally adds a node that gives the most reduction in MDL cost, till no further improvements occur. The second, GraphMap, guesses the size k of the optimal solution, and greedily picks the k nodes that would minimize the cost. We show that the cost of the solution produced by GraphMap is within an additive factor of the optimum.

3.1 Algorithm LocalSearch Here we discuss an algorithm based on a greedy local search approach which is popularly used in many MDL optimizations. Our algorithm LocalSearch works as follows: we initialize $\hat{I}$ to $\mathcal{Q}$. Then for each node v in $V \setminus \hat{I}$, we compute marginal change in the MDL cost caused by adding v to $\hat{I}$. We add the node u which results in maximum decrease in the MDL cost to $\hat{I}$. We repeat the process until the MDL cost cannot be reduced further. The complete pseudocode is given in Algorithm 1. Although intuitive and natural, it is hard to get provable guarantees for this algorithm.

Algorithm 1 Algorithm LocalSearch

Input: Instance $(V, \mathcal{Q}, p, F, \gamma)$
Output: Solution $\hat{I}$ that minimizes $\mathcal{L}(|\mathcal{Q}|, |\hat{I}|, \hat{I}, \mathcal{Q})$
$\hat{I} \leftarrow \mathcal{Q}$
while $\exists\, v \in V \setminus \hat{I} : \mathcal{L}(|\mathcal{Q}|, |\hat{I}|, \hat{I}, \mathcal{Q}) - \mathcal{L}(|\mathcal{Q}|, |\hat{I}| + 1, \hat{I} \cup \{v\}, \mathcal{Q}) > 0$ **do**
  $u \leftarrow \arg\max_{v \in V \setminus \hat{I}} \mathcal{L}(|\mathcal{Q}|, |\hat{I}|, \hat{I}, \mathcal{Q}) - \mathcal{L}(|\mathcal{Q}|, |\hat{I}| + 1, I \cup \{v\}, \mathcal{Q})$
  $\hat{I} \leftarrow \hat{I} \cup \{u\}$
end while
Return $\hat{I}$

3.2 Algorithm GraphMap In this section we propose an efficient algorithm for finding a failure set I which also provides an additive approximation guarantee on the MDL cost of the solution, thereby ensuring that our nodes-set is of high-quality.

First, let $A = -\log\binom{|I|}{|\mathcal{Q}|} - 2|\mathcal{Q}|\log(\gamma)$. We rewrite the MDL cost function in the following manner:

$$
\begin{aligned}
&\mathcal{L}\big(|\mathcal{Q}|, |I|, I, \mathcal{Q}\big) \\
=& A - \log\Big(\sum_{s \in V} p_s(s) \prod_{v \in I} F(v \mid s) \prod_{v' \notin I} \Big(1 - F(v' \mid s)\Big)\Big) \\
&- 2(|I| - |\mathcal{Q}|)\log(1-\gamma) \\
=& A - \log\Big(\sum_{s \in V} p_s(s) \prod_{v \in V} \Big(1 - F(v' \mid s)\Big) \\
&\qquad \prod_{v \in I} \frac{F(v \mid s)}{(1 - F(v \mid s))}\Big) - \log{(1-\gamma)}^{2(|I| - |\mathcal{Q}|)} \\
=& A - \log\Big(\sum_{s \in V} p_s(s) \prod_{v \in V} \Big(1 - F(v' \mid s)\Big)(1-\gamma)^{-2|\mathcal{Q}|} \\
&\qquad \prod_{v \in I} \frac{F(v \mid s)(1-\gamma)^{2|I|}}{(1 - F(v \mid s))}\Big) \\
=& A - \log\Big(\sum_{s \in V} g(s) \prod_{v \in I} f(s, v)\Big),
\end{aligned} \tag{3.9}
$$

where $g(s) = (1-\gamma)^{-2|\mathcal{Q}|} p_s(s) \prod_{v \in V} \Big(1 - F(v \mid s)\Big)$ and $f(s, v) = \frac{F(v \mid s)(1-\gamma)^2}{1 - F(v \mid s)}$. Therefore, the problem reduces to finding a set $\hat{I}$ such that

$$
\begin{aligned}
\hat{I} = \arg\min_I \Big\{ &-\log\binom{|I|}{|\mathcal{Q}|} + 2|\mathcal{Q}|\lambda_1 \\
&- \log\Big(\sum_{s \in V} g(s) \prod_{v \in I} f(s, v)\Big)\Big\}.
\end{aligned} \tag{3.10}
$$

In GraphMap, we first compute quantities $f(s, v)$ for each $s, v \in V$ and $g(s)$ for each $s \in V$. Then for each $s \in V$, we sort nodes $v \in V$ by $f(s, v)$. The main idea in our algorithm is to use the quantity $f(s, v) = \frac{F(v \mid s)(1-\gamma)^2}{1 - F(v \mid s)}$ defined above as the 'weight' for each pair (s, v). We guess the size of the solution $|I_s|$, if the source were to be s. Then we compute, $\phi(s, I_s)$. For each possible size k of the failure set, we compute the cost $h(k)$. Based on pre-computed $\phi(s, I_s)$ and $h(k)$, we compute $\alpha(s, k)$. Finally, we pick the set of $|I_s|$ nodes which maximizes $\alpha(s, k)$. The complete pseudocode is presented in Algorithm 2 and analyzed in Theorem 3.1.

Algorithm 2 Algorithm GraphMap

Input: Instance $(V, \mathcal{Q}, p, F, \gamma)$
Output: Solution $\hat{I}$ that minimizes $\mathcal{L}(|\mathcal{Q}|, |\hat{I}|, \hat{I}, \mathcal{Q})$

1: For each $s, v \in V$, compute $f(s, v)$
2: For each $s \in V$, compute $g(s)$
3: **for** each $s \in V$ **do**
4: Order the nodes $v_1^s, \ldots, v_{n-|\mathcal{Q}|}^s$ such that $f(s, v_1^s) \geq f(s, v_2^s) \geq \ldots$
5: {The set $I_s(k) = \{v_1, \ldots, v_k\} \cup \mathcal{Q}$ will be considered}
6: Compute $\phi(s, \emptyset) = g(s) \prod_{v \in \mathcal{Q}} f(s, v)$
7: **for** $k = 1$ to $n - |\mathcal{Q}|$ **do**
8: Compute $\phi(s, I_s(k)) = \phi(s, I_s(k-1)) f(s, v_k)$
9: { $\phi(s, I_s(k)) = g(s) \prod_{v \in I_s(k)} f(s, v)$}
10: **end for**
11: **end for**
12: $h(0) = 1$
13: **for** $k = 1$ to $|V| - |\mathcal{Q}|$ **do**
14: $h(k) = h(k-1)(k + |\mathcal{Q}|)/k$
15: **end for**
16: **for** $s \in V$ **do**
17: **for** $k = 1$ to $|V| - |\mathcal{Q}|$ **do**
18: Compute $\alpha(s, k) = -\log \phi(s, I_s(k)) - \log h(k)$
19: **end for**
20: **end for**
21: Return $I_s(k)$ which maximizes $\alpha(s, k)$

Theorem 3.1. *Let I^* be the set minimizing the MDL cost, and let I denote the solution computed by Algorithm* GraphMap. *Then, $\mathcal{L}(|\mathcal{Q}|, |I|, I, \mathcal{Q}) \leq \mathcal{L}(|\mathcal{Q}|, |I^*|, I^*, \mathcal{Q}) + \log(n)$, where n is the number of seed nodes.*

Proof. Recall the definitions of $g(s)$, $f(s, v)$ and A above. Then,

$$\mathcal{L}(|\mathcal{Q}|, k, I, \mathcal{Q}) = -\log\Big(\sum_{s \in V} \phi(s, I)\Big) + A \quad ,$$

where $\phi(s, I) = g(s) \prod_{v \in I} f(s, v)$. Note that $\phi(s, I)$ is maximized for the set $I_s(k)$ defined in Algorithm GraphMap, since this consists of the set of top $k - |\mathcal{Q}|$ nodes in $V \setminus \mathcal{Q}$, with respect to the quantity $f(s, v)$, along with all nodes in $\mathcal{Q}$. Therefore, we have

$$\phi(s, I_s(k)) \geq \phi(s, I^*)$$

Adding over all possible seed nodes, we have

$$\sum_{s \in V} \phi(s, I_s(|I^*|)) \geq \sum_{s \in V} \phi(s, I^*)$$

which implies for some seed $\hat{s}$, we have

$$\phi(\hat{s}, I_{\hat{s}}(k)) \geq \frac{1}{n} \sum_{s \in V} \phi(s, I^*)$$

$$\begin{aligned} \Rightarrow -\log\Big(\phi(\hat{s}, I_{\hat{s}}(k))\Big) &\leq -\log\Big(\frac{1}{n} \sum_{s \in V} \phi(s, I^*)\Big) \\ &= -\log\Big(\sum_{s \in V} \phi(s, I^*)\Big) + \log(n) \end{aligned}$$

This, in turn, implies

$$\begin{aligned} \mathcal{L}(|\mathcal{Q}|, k, I_{\hat{s}}(k), \mathcal{Q}) &= -\log\Big(\sum_{s \in V} \phi(s, I_{\hat{s}}(k))\Big) + A \\ &\leq -\log\Big(\phi(\hat{s}, I_{\hat{s}}(k))\Big) + A \\ &\leq -\log\Big(\sum_{s \in V} \phi(s, I^*)\Big) + \log(n) + A \\ &\leq \mathcal{L}(|\mathcal{Q}|, k, I^*, \mathcal{Q}) + \log(n), \end{aligned}$$

where the first inequality follows because $\phi(s, I) \geq 0 \; \forall \; s, I$, so that $\sum_{s \in V} \phi(s, I_{\hat{s}}(k)) \geq \phi(\hat{s}, I_{\hat{s}}(k))$. Since Algorithm 2 searches over all possible solution sizes k, the theorem follows.

Lemma 3.1. *Algorithm* GraphMap *runs in $O(|V|^2 \log |V|)$ time.*

Proof. The quantities $g(s)$ and $f(s, v)$ defined earlier in (3.9) can be computed for all s, v in $O(|V|^2)$ time, which is done in lines 1 and 2. The sort step in line 4 takes $O(|V| \log |V|)$ time. The inner for loop in lines 7-10 computes $\phi(s, I_s(k))$ defined above, and takes $O(|V|)$ time. Therefore, the for loop in the lines 3–11 takes $O(|V|^2 \log |V|)$ time. The remainder of the steps take $O(|V|^2)$ time.

4 Experiments

4.1 Setup We briefly describe our setup next. All the experiments were conducted on a hybrid cluster with over 2500 nodes and 28 TB of RAM. GraphMap takes roughly 30 minutes to complete for any setting in a single node for our JAM dataset (See Section 4.2), which is very practical. Our code is publicly available for academic purposes[1].

4.2 Datasets We evaluate performance of our algorithms on various synthetic and real networks. We discuss our datasets in detail next.

Synthetic Dataset. We created a simple 60×60 grid where each cell is considered as a node in a road network, leading to 3600 nodes. We assumed an uniform

[1] Code and data at: `http://tiny.cc/GraphMap`

seed probability distribution across all nodes. We computed conditional failure probabilities (PlainCF) between pair of nodes (s, v) based on Geographically Correlated Failure (GCF) Model [1] i.e., if s is the seed node then

$$F(v \mid s) = 1 - d(s, v) \tag{4.11}$$

where $d(s, v)$ is a distance function $d : V \times V \to [0, 1]$. In our case, $d(\cdot, \cdot)$ is the Manhattan distance between the nodes normalized by the maximum distance. We will refer to this set of conditional failure probabilities as GCF.

Real Datasets. We created three datasets from real world node failure logs in transportation and power-grid networks. We use failure logs in road networks from WAZE alerts data, which is publicly available on the City of Boston's website[2]. These alerts have been reported by users of WAZE[3] between Monday 23$^{\text{rd}}$ February, 2015 and Sunday 1$^{\text{st}}$ March, 2015. WAZE is a popular crowd sourced application commonly used for navigation. The alerts in the dataset are spatially distributed across Boston, Cambridge, and Brook-line regions of Massachusetts. The alerts include different types failures such as traffic jam, extreme weather, accidents, and road closures. Additionally, the latitude and longitude of the affected locations, and start and end time of the alert is also given. From these alerts, we created two datasets based on traffic jam (JAM) and extreme weather (WEATHER).

Similarly, we use a list of Electric disturbance events from Energy.gov[4]—this list includes reported events of electric emergencies and disturbance in power supply from 2002 to 2015. Each event log contains information regarding date and time of the beginning and restoration of the event, geographical areas affected by the event, number of customer affected, and so on. We created POWER-GRID dataset from the log of electric emergencies and disturbances.

Network Creation. As discussed in Section 2.1, we need to define the seed probability and pair-wise conditional failure probability distributions over all nodes in the network. This is done in the following manner. For WAZE alert data, we have partitioned the complete geographical region occupied by these failures by using a 119×78 grid as shown in Figure 2a, where each cell is $0.00166°$ square and acts as a node in our virtual road network. For the POWER-GRID data, each location referred in the dataset acts as a node.

Seed Probability. Let $n_v = 1$ denote if node v is failed, 0 otherwise as discussed above, and let $N = \sum_v n_v$ denote the total number of failures across all nodes. We define the seed probabilities as:

$$p_s(v) = \frac{n_v}{N} \tag{4.12}$$

Conditional Failure Probabilities. We construct a Binary Failure State Time Series (BinTS) for a span of 7 days, by using the temporal information that is available from WAZE alerts data. This time series gives a binary (0 or 1) value for each time step which represents the failure state of the respective node i.e., $BinTS_v(t) = 1$ implies that there is at-least one failure in v at time t. Using BinTS we were able to compute the pair-wise conditional failure probabilities for our datasets in the following manner. For two nodes v_1 and v_2, we define the Plain Conditional Failure Probability (PlainCF) of v_1 given v_2 as the ratio between number of time steps in which both v_1 and v_2 are failed (i.e. with value 1 in BinTS) to the number of time steps in which only v_2 is failed.

$$PlainCF(v_1 \mid v_2) = \frac{|\{t|\forall t, BinTS_{v_1}(t) = 1 \& BinTS_{v_2}(t) = 1\}|}{|\{t|\forall t, BinTS_{v_2}(t) = 1\}|}$$

We study two more synthetic conditional failure probabilities, named URandCF and NRandCF, for each dataset; these are defined in the following manner: URandCF is an arbitrary sample from a uniform distribution and NRandCF is an arbitrary sample from a normal distribution $(0.1 \times \mathcal{N}(5, 1))$ over the values $[0, 1]$. We follow the same procedure to generate conditional probability failures for the POWER-GRID dataset.

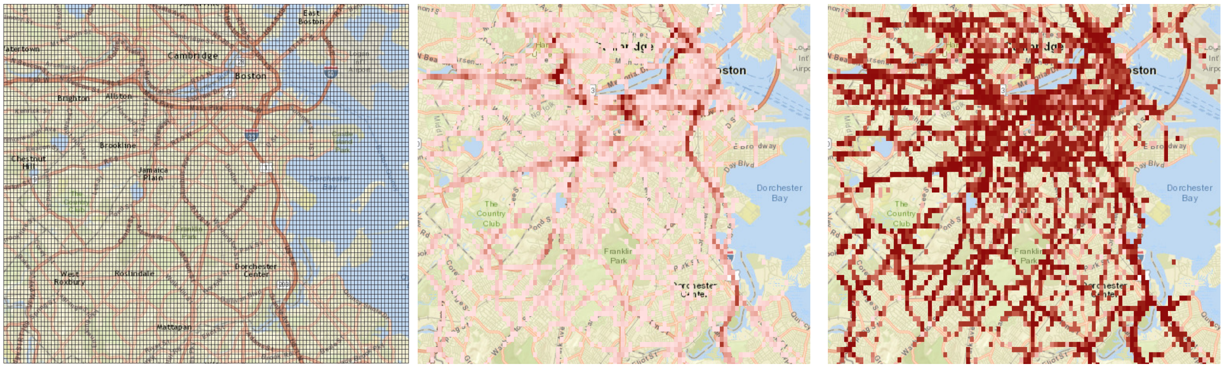
Descriptions. Following the above steps, we finally get the three different datasets below[1].

JAM: This is a dataset that we created from WAZE alerts data using data of failure type JAM. The resultant dataset consists of a road network with 2650 nodes along with seed probability distribution and conditional failure probabilities as discussed above. Figure 2b shows the spatial distribution of the seed probabilities and Figure 2c shows distribution of conditional failure probabilities for a randomly chosen seed.

WEATHER: Similar to the JAM dataset this dataset is created by using WEATHERHAZARD failure data from WAZE alerts data. The resultant dataset consists of a road network with 1520 nodes along with seed probability distribution and conditional failure probabilities computed as discussed above.

POWER-GRID: As mentioned earlier, we created this dataset from the log of electrical emergencies and disturbances. We filtered out the events in the log which were not related to loss of electric service. For this dataset, which consists of 24 nodes, we computed failure likelihood and conditional failure probabilities are before.

[2]https://data.cityofboston.gov/
[3]https://www.waze.com/
[4]https://www.oe.netl.doe.gov/



(a): Partitions of Boston region occupied by WAZE alerts. (b): Spatial distribution of seed probabilities. (c): Distribution of PlainCF for a random seed node.

Figure 2: JAM dataset created from WAZE alerts data of Boston, Cambridge, Brook-line regions. The darker regions indicate higher probability of failure. Note that the seed probability is higher closer to the city.

4.3 Performance evaluation In this section we discuss the performance of our algorithms against various datasets that are described earlier across various values of $\gamma \in [0.1, 1.0]$. We examine the precision, recall and F1-score for GraphMap, compared with LocalSearch. We observe that our MDL based approach does indeed allow us to infer unknown/missing failures in the network using the probes. The specific MDL formulation we consider in Section 2.6, which includes $|I|$ in the model seems to perform much better than other natural MDL formulations.

Comparison of GraphMap with LocalSearch. Figure 3 presents a comparison of the trends in performance of both algorithms on the JAM dataset with PlainCF probabilities across $\gamma \in [0.1, 1.0]$ and MDL cost of their respective solutions. For both algorithms, the performance varies with the sampling rate, γ. We find that the solution computed by GraphMap has lower MDL cost, compared to the baseline. One interesting observation from Figure 3b is that the recall for GraphMap decays with γ till 0.4 and then increases. GraphMap has higher F1-score compared to LocalSearch, for most values of γ. In the rest of our evaluation, we only consider GraphMap.

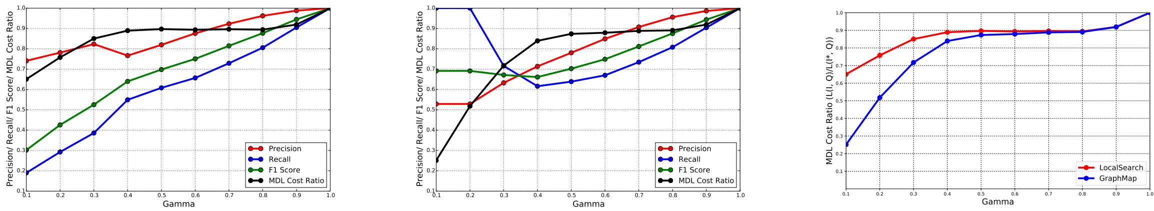
Performance of GraphMap for different datasets. Figures 4, 5, 6 and 7 present the performance of Algorithm GraphMap for all the datasets, and for the three different ways of defining conditional probabilities. Across all these results, on average we are able to find 80% of the failed nodes with an average precision of 79% across various values of $\gamma \in [0.1, 1.0]$. In other words, we are successful in inferring a reasonable fraction of unknown/missing failures in the network from partial set of observations with a reasonable precision.

5 Related Work

Some of the different areas related to our work include network and state inference in communication networks, reconstructing networks from cascades and inferring missing infections in the case of epidemics. We briefly discuss these below.

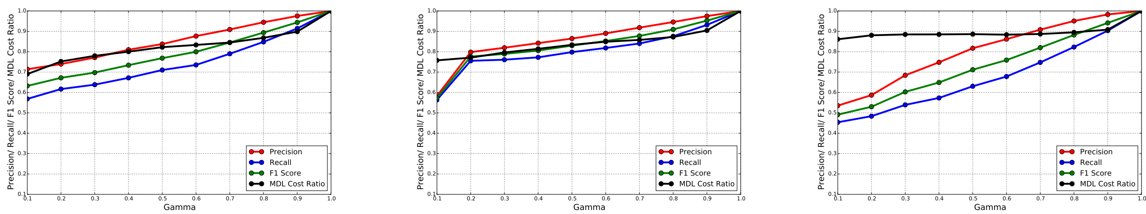
The area of network tomography involves inferring link states, such as delays and failures, in the Internet and other communication networks; see, e.g., [8, 19, 10, 2, 4]. Probes such as end-to-end delays are the only measurements that are available in such networks. At an abstract level, the problem here involves solving for the link delay vector $\mathbf{x}$, given the measured delays across the probes. This becomes a very challenging problem and it is typically assumed that link characteristics such as delays are modeled as *independent* random variables with known distributions, but potentially unknown parameters. Xia et al. [19] solve this assuming link delays are exponentially distributed. Ni et al. [10] study different kinds of probing models, including multicast probes which can give estimates on a tree, and develop methods for inferring the topology in dynamic networks. There has also been work on designing probes to infer part of the network structure, as in [2].

Another related topic is the inference of the source of an infection and other missing infections, in the case of epidemic spread on networks. Epidemics are modeled as stochastic processes, e.g., SI/SIR, in which the infection spreads from an infected node to its susceptible neighbors. Usually, only partial information about the infections is known, and some of the problems that have been studied include identifying the source of an infection and finding other missing nodes [11, 13, 18, 17, 16]. The Minimum Description Length (MDL) principle [6, 12] has been successfully used in [11, 18] for



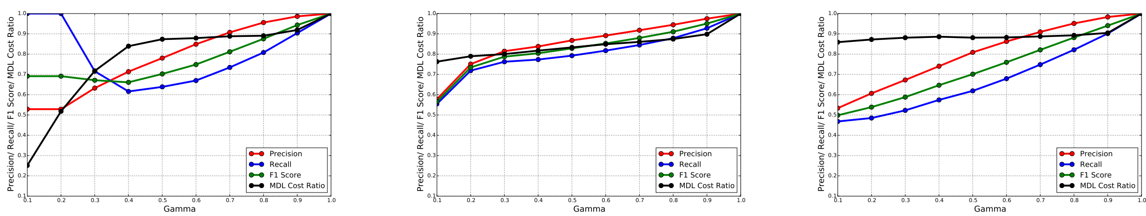
(a) (b) (c)

Figure 3: Performance of the (a) LocalSearch and (b) GraphMap on the JAM dataset with PlainCF probabilities. The MDL costs of the solutions is shown in (c).



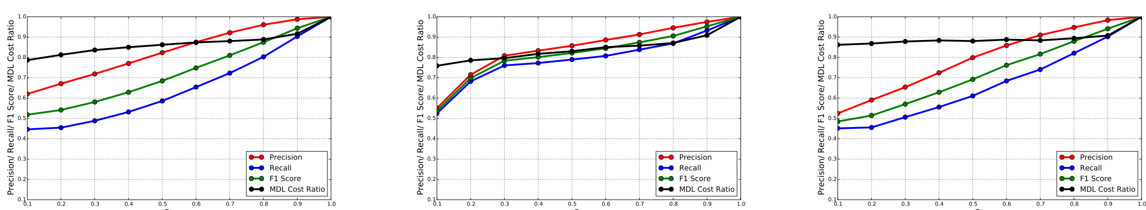
(a) (b) (c)

Figure 4: Performance of GraphMap on the Synthetic Grid Dataset with (a) GCF, (b) URandCF, and (c) NRandCF conditional probabilities.



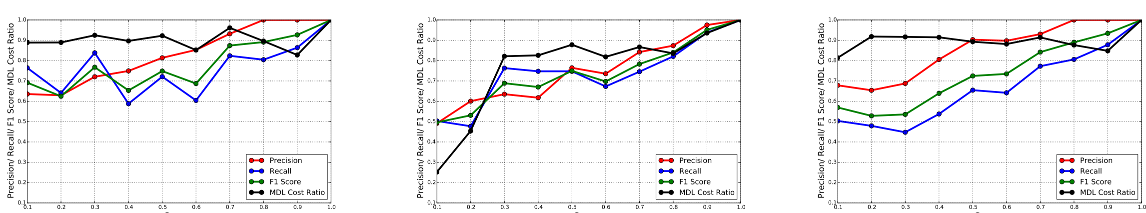
(a) (b) (c)

Figure 5: Performance of GraphMap on the JAM Dataset with (a) PlainCF, (b) URandCF, and (c) NRandCF conditional probabilities.



(a) (b) (c)

Figure 6: Performance of GraphMap on the WEATHER Dataset with (a) PlainCF, (b) URandCF, and (c) NRandCF conditional probabilities.



(a) (b) (c)

Figure 7: Performance of GraphMap on the Power Grid Dataset with (a) PlainCF, (b) URandCF, and (c) NRandCF conditional probabilities.

these problems, whereas [17] develop an MLE method. As discussed before, these methods do not give rigorous approximation algorithms or give it only for special graphs (like k-regular trees). In contrast, we give an additive approximation for our MDL formulation for any graph.

A class of failure models, different from the SI/SIR type of epidemics, has been studied extensively, motivated by settings such as disaster events, e.g., [3, 1, 14]. These studies assume an initial failure, and subsequent failures whose probability is correlated with the source. For instance, in [1], the probability $p(j|i)$ that node j fails, given that i is the source is a function of the distance from i to j, with the probabilities decaying with the distance. Our work is motivated by these models.

6 Conclusions

Our results show that an MDL based approach is quite useful in the problem of inferring missing failures in settings with correlated failures. This motivates its use in other inference problems with partial information. We have considered the simplest notion of a probe here—information about specific nodes which have failed. Extending our work to other kinds of probes (like connectivity queries) is an interesting and natural problem. Inferring the state of the network using such probes, and supporting additional queries are interesting problems. Further we have assumed there are no observational errors—designing robust and provable algorithms in face of errors is also interesting future work.

Acknowledgements This paper is based on work partially supported by NSF-CAREER, NEH HG-229283-15, ORNL Order 4000143330, the Maryland Procurement Office H98230-14-C-0127, a Facebook faculty gift, DTRA CNIMS HDTRA1-11-D-0016-0010, HDTRA1-17-0118, NSF IIS-1633028 and ACI-1443054.

References

[1] P. K. Agarwal, A. Efrat, S. K. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman. The resilience of wdm networks to probabilistic geographical failures. *IEEE/ACM Trans. Netw.*, 21(5):1525–1538, Oct. 2013.

[2] A. Anandkumar, A. Hassidim, and J. Kelner. Topology discovery of sparse random graphs with few participants. *SIGMETRICS Perform. Eval. Rev.*, 39(1):253–264, June 2011.

[3] Y. Berezin, A. Bashan, M. M. Danziger, D. Li, and S. Havlin. Localized attacks on spatially embedded networks with dependencies. *Scientific Reports*, 2015.

[4] N. G. Duffield, J. Horowitz, F. L. Presti, and D. Towsley. Multicast topology inference from measured end-to-end loss. *IEEE Trans. Inf. Theory*, 2002.

[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, volume 29, pages 251–262, 1999.

[6] P. Grünwald. A tutorial introduction to the minimum description length principle. In *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2004.

[7] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Rev.*, 42(4):599–653, 2000. http://www.math.rutgers.edu/ leenheer/hethcote.pdf.

[8] X. Jin, W. Yiu, S. Chan, and Y. Wang. Network topology inference based on end-to-end measurements. *IEEE Journal on Selected areas in Communications*, 2006.

[9] M. Marathe and A. Vullikanti. Computational epidemiology. *Communications of the ACM*, 56(7):88–96, 2013.

[10] J. Ni, H. Xie, S. Tatikonda, and Y. Yang. Efficient and dynamic routing topology inference from end-to-end measurements. *IEEE/ACM Transactions on Networking*, 2010.

[11] B. A. Prakash, J. Vreeken, and C. Faloutsos. Spotting culprits in epidemics: How many and which ones? In *ICDM*, 2012.

[12] J. Rissanen. Minimum description length principle. In S. Kotz and N. L. Johnson, editors, *Encyclopedia of Statistical Sciences*, volume V, pages 523–527. John Wiley and Sons, New York, 1985.

[13] P. Rozenshtein, A. Gionis, B. A. Prakash, and J. Vreeken. Reconstructing an epidemic over time. In *KDD*, pages 1835–1844, 2016.

[14] H. Saito. Analysis of geometric disaster evaluation model for physical networks. *IEEE/ACM Transactions on Networking*, 2015.

[15] D. J. Schwab, R. F. Bruinsma, J. L. Feldman, and A. J. Levine. Rhythmogenic neuronal networks, emergent leaders, and k-cores. *Physical Review E*, 82(5):051911, 2010.

[16] E. Sefer and C. Kingsford. Diffusion archeology for diffusion progression history reconstruction. *Knowledge and information systems*, 49:403–427, 2016.

[17] D. Shah and T. Zaman. Detecting sources of computer viruses in networks: theory and experiment. In *Proceedings of the ACM International Conference on Performance Evaluation (SIGMETRICS)*, pages 203–214, 2010.

[18] S. Sundareisan, J. Vreeken, and B. A. Prakash. Hidden hazards: Finding missing nodes in large graph epidemics. In *Proc. of SIAM International Conference on Data Mining (SDM)*, 2015.

[19] Y. Xia and D. Tse. Inference of link delay in communication networks. *IEEE Journal on Selected areas in Communications*, 2006.