Distributed Systems Diagnosis Using Belief Propagation

Irina Rish

IBM T.J. Watson Research
Hawthorne, NY 10532, USA
rish@us.ibm.com

Abstract

In this paper, we focus on diagnosis in distributed computer systems using end-to-end transactions, or probes. Diagnostic problem is formulated as a probabilistic inference in a bipartite noisy-OR Bayesian network. Due to general intractability of exact inference in such networks, we apply belief propagation (BP), a popular approximation technique proven successful in various applications, from image analysis to probabilistic decoding. Another attractive property of BP for our application is it natural parallelism that allows a distributed implementation of diagnosis in a distributed system to improve diagnostic speed and robustness. We derive lower bounds for diagnostic error in bipartite Bayesian networks, and particularly in noisy-OR networks, and provide promising empirical results for belief propagation on both randomly generated and realistic noisy-OR problems.

1 Introduction

One of the central problems in management of complex distributed systems is real-time diagnosis of various faults and performance degradations. However, there is always a trade-off between the quality of diagnosis (e.g., diagnostic accuracy and speed) and its cost, which involves both the cost of collecting various measurements and performing tests, as well as computational cost of diagnosis.

Particularly, in this paper, we will focus on fault diagnosis in distributed computer systems using *probes*. A probe is an end-to-end transaction (e.g., ping, webpage access, database query, an e-commerce transaction, etc.) sent through the system for the purposes of monitoring and testing. Usually, probes are sent from one or more designated machines called probing stations; probes ”go through” multiple system components, including both hardware (e.g. routers and servers) and software components (e.g. databases and various applications). A probe can be viewed as a disjunctive test over the components involved in the probe: indeed, a probe is OK if and only if all the involved components are OK, otherwise the probe fails. The problem of selecting minimal number of probes for diagnosis is closely related to the *group testing* problem [4] but is more complex due to constraints on probe construction, such as the network topology, and available application-level transactions. For more detail on existing probe selection approaches see [7].

In case of noisy probe outcomes, we address diagnosis as a probabilistic inference in a Bayesian network that represents the dependencies between the unobserved states of system components and observed probe outcomes; conditional probabilities for probe outcomes given the corresponding components are defined by the *noisy-OR* model which generalizes disjunctive tests to the case of noisy environment. However, exact inference in such networks is known to be NP-hard, and approximate approaches are required.

Motivated by recent success of *belief propagation* [9] in various applications, from probabilistic decoding [5] to image processing and medical diagnosis, we decided to investigate the applicability of this algorithm to our noisy-OR problems. Belief propagation is particularly attractive for distributed system's diagnosis since it allows a naturally distributed implementation, and thus a distributed monitoring/diagnosis architecture that eliminates the computational bottleneck associated with a central monitoring server.

We start with a theoretical analysis of diagnostic error and provide a lower bound on "bit-error rate", assuming most-likely diagnosis for each unobserved variable ("bit-wise decoding") provided by belief propagation[1]. Next, we demonstrate empirical results for belief propagation diagnosis both on randomly generated networks and on realistic Internet-like topologies simulated by INET generator [12]. The results are quite encouraging: belief propagation achieves a high-accuracy diagnosis, especially for realistic cases of low prior fault probability. There is also an interesting observation regarding the effect of the probe length on the error in randomly generated networks (lower error is achieved with shorter probes), which requires further theoretical investigation.

2 Background

We consider a system that contains n *components* that we are interested in diagnosing, each of which can be either be "OK" (functioning correctly) or "faulty" (functioning incorrectly). The *state* of the system is denoted by a vector $\mathbf{X} = (X_1, ..., X_n)$ of Boolean variables, where $X_i = 1$ denotes faulty state and $X_i = 0$ denotes OK state of the i-th component. Lowercase letters denote the values of the corresponding variables, e.g. $\mathbf{x} = (x_1, ..., x_n)$ denotes a particular assignment of node values.

The outcome of a *probe*, or *test*, T depends on the states of a subset of system components. A probe either succeeds or fails: if it succeeds (denoted $T = 0$), then every component in the subset is OK; it fails (denoted $T = 1$) if *any* of the components in the subset is faulty.

In the presence of noisy probes and various probabilities of fault, it is convenient to use a probabilistic framework of Bayesian networks [9] that allows a compact representation of the joint probability distribution over the system states and test outcomes. A Bayesian network is a directed acyclic graph, where the nodes correspond to variables, and edges denote direct dependencies between the variables. Graphical structure encodes independence assumptions among the variables: namely, node Y is independent of its non-descendants in the graph given its *parents* $\mathbf{pa}(Y_i)$ (nodes pointing to Y). Each node Y_i is associated with a *conditional probabilities distribution (CPD)* $P(Y_i|\mathbf{pa}(Y_i))$, and the joint probability $P(\mathbf{Y})$ represented by the Bayesian network is written in factorized form as $P(\mathbf{Y}) = \prod_i P(Y_i|\mathbf{pa}(Y_i))$.

We assume that a probe outcome is independent on other probe outcomes given the states of probe's components, and that node failures are marginally independent. These assumptions are captured by a bipartite Bayesian network, such as one shown in Figure 1a. The network

[1]See [10] for analysis of block-error rate, or MAP diagnosis; note that in general, MAP diagnosis is more accurate than the bit-wise diagnosis since the MAP diagnosis aims at finding the optimal (most-likely) assignment to all X_i simultaneously.

represent a joint probability $P(\mathbf{x}, \mathbf{t})$:

$$P(\mathbf{x}, \mathbf{t}) = \prod_{i=1}^{n} P(x_i) \prod_{j=1}^{m} P(t_j | \mathbf{pa}(t_j)). \tag{1}$$

Since a probe succeeds if and only if all its components are OK, a probe outcome is a logical-OR function of its components, i.e. $T_i = X_{i_1} \vee ... \vee X_{i_k}$, where $\vee$ denotes logical OR, and $X_{i_1}, ..., X_{i_k}$ are all the nodes probe T_i goes through. In practice, however, this relationship may be disturbed by noise. For example, a probe can fail even though all the nodes it goes through are OK (e.g., due to another reason, or ”hidden cause”). Conversely, there is a chance that a probe succeeds even if a node on its path has failed (e.g., dynamic routing may result in the probe following a different path). Such uncertainties can be described by an *noisy-OR* model, where every probe T_j and component X_i on probe's path are associated with a *noise* parameter q_{ij}, also called *inhibition probability*, or *link probability* – a small probability that probe T_j succeeds even if node X_i on its path fails. There is also a parameter q_{leak} called the *leak probability* which corresponds to inhibition probability for the additional ”hidden cause”: if this additional cause of probe failure is not completely inhibited, i.e. $q_{leak} < 1$, then a probe may fail even when all the nodes on its path are OK. Finally, noisy-OR model assumes *causal independence*, i.e. it assumes that different causes (e.g., node failures) contribute independently to a common effect (probe failure). The the conditional probability distribution for each probe T_j can be written as

$$P(t_j = 0 | x_1, \ldots, x_k) = q_{leak} \prod_{i=1}^{n} q_{ij}^{x_i}$$

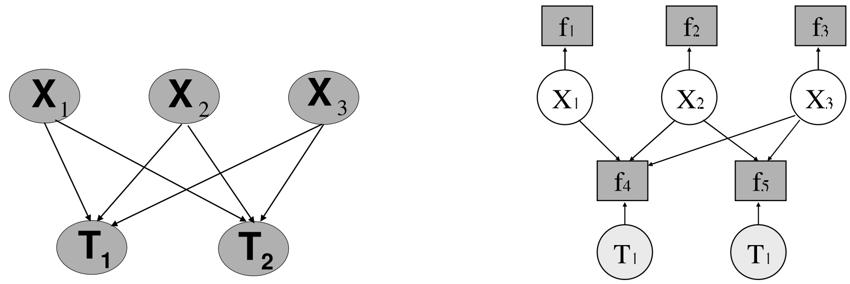
where $X_1, \ldots, X_k$ is the set of parents $\mathbf{pa_j}$ of T_j.

3 Probabilistic Diagnosis in Distributed Systems

Given the probe outcomes, we wish to find the most-likely assignment (called *maximum aposteriory probability, or MAP)* to all X_i nodes given the probe outcomes, i.e. $\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{t})$. Since $P(\mathbf{x}|\mathbf{t}) = \frac{P(\mathbf{x},\mathbf{t})}{P(\mathbf{t})}$, where $P(\mathbf{t})$ does not depend on $\mathbf{x}$, we get $\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x}, \mathbf{t})$. An alternative approach is to find the most likely value x_i^* of each node X_i separately, i.e. to find an assignment $\mathbf{x}' = (x'_1, ..., x'_n)$ where $x'_i = \arg\max_{x_i} P(x_i|\mathbf{t})$, $i = 1, ..., n$. We refer to the latter approach as *bit-wise diagnosis (bit-wise decoding)*, while the MAP approach can be viewed as a *block-wise diagnosis (block-wise decoding)*. Bit-wise diagnosis is more suited when using belief updating algorithms that compute posterior probability $P(X_i|\mathbf{T})$ for each X_i, rather than perform global optimization to find MAP, using either search or dynamic programming [3]. Clearly, bit-wise diagnosis may be suboptimal to MAP since in general, $x^* \neq \mathbf{x}'$. Nevertheless, it is often used in various practical applications, e.g., in LDPC decoding using belief propagation, where it seem to work quite well.

Unfortunately, both MAP inference and belief updating are known to be NP-hard [1], and the complexity of best-known inference techniques is exponential in the graph parameter known as *treewidth*, or *induced width* [2], which reflects the size of a largest clique in the graph (and thus the largest dependency) created by an inference algorithm. Treewidth depends on the graph structure of a Bayesian network: e.g., it is small (just 1) for trees, but tends to grow with size and density of the network; in our diagnostic applications, it was often observed to be intractable so that exact inference could not be used for diagnosis[2].

[2]There exist an alternative exact inference algorithm, called *Quickscore*[6], which is specifically derived for noisy-OR networks; its computational complexity does not depend on the treewidth but is exponential in the



(a) (b)

Figure 1: (a) A two-layer Bayesian network structure for a set $\mathbf{X} = (X_1, X_2, X_3)$ of system's components and a set of probes $\mathbf{T} = (T_1, T_2)$, and (b) a corresponding factor graph.

3.1 Belief Propagation

In order to cope with the complexity of inference, approximation algorithms are commonly used. For example, *belief propagation* algorithm [9], designed for networks without undirected cycles (polytrees), was successfully applied as an approximation to cyclic networks, e.g. to decoding LDPC codes (which are intractable for exact decoding schemes), and recently became the state-of-art approach that outperforms many previously known decoding techniques; surprisingly accurate decoding using LDPC codes with belief propagation decoder was even called a ”revolution” in coding theory [5][3].

Belief propagation (BP) is a simple linear-time message-passing algorithm that is provably correct on polytrees and can be used as an approximation on general networks. Belief propagation passes probabilistic messages between the nodes and can be iterated until convergence (guaranteed only for polytrees); otherwise, it can be stopped at certain number of iterations. The algorithm computes approximate beliefs (posterior probability distributions given observations) for each node.

We describe the algorithm in a more recent terminology of factor graphs rather than Bayesian network. A factor graph is a convenient representation generalizing directed (Bayesian networks) and undirected (Markov networks) probabilistic graphical models. It assumes that joint distribution $P(\mathbf{x})$ is represented as a product $P(\mathbf{x}) = \frac{1}{Z}\prod_a f_a(\mathbf{x}_a)$ where Z is a normalization constant called the *partition function*, and the index a ranges over all factors $f_a(\mathbf{x}_a)$, defined on the corresponding subsets $\mathbf{X}_a$ of $\mathbf{X}$. A factor graph is an undirected bipartite graph that contains factor nodes, shown as squares, and variable nodes, shown as circles; there is a link between variable node and factor node if and only if the variable participates in the corresponding. It is easy to convert a Bayesian network to a factor graph: every CPD and prior distribution will correspond to a factor node. For example, Figure 1b shows the factor graph for the Bayesian network in 1a.

The principle of belief propagation is simple and intuitive: each node sends messages to its neighbors about its belief regarding its own state. The messages are then multiplied by the local potential functions to update the neighbor's beliefs. The process is iterated until belief

number of faulty probes; this algorithm can work well when faults are rare and not many probes go through a faulty component. However, the number of simultaneous faults grows with the size of the system, especially if faults denote performance degradations rather than ”hard” faults; besides, realistic distributed systems tend to have ”hub” nodes which belong to multiple probes, e.g., in Internet-like network topologies, sometimes causing multiple probes to fail simultaneously.

[3]Belief propagation was also successfully applied to other domains, such as image processing and medical diagnosis, and recent progress in understanding belief propagation and its relation to free energy minimization lead to generalization of the algorithm to even better approximations [8].

fluctuations fall below a small threshold, or until patience runs out, at which point one declares divergence.

Let a denote a factor node and i one of its variable nodes. $N(a)$ represents the neighbors of a, i.e., the set of variable nodes connected to that factor; $N(i)$ denotes the neighbors of i, i.e., the set of factors nodes to which variable node i belongs. The BP messages are defined as follows [8]:

$$n_{i\to a} := \prod_{c\in N(i)\setminus a} m_{c\to i}(x_i), \text{ and } m_{a\to i}(x_i) := \sum_{\mathbf{x}_a\setminus x_i} f_a(\mathbf{x}_a) \prod_{j\in N(a)\setminus i} n_{j\to a}(x_j) \tag{2}$$

Based on these messages, we can compute the beliefs about each node and about the probability potential for each factor, respectively:

$$b_i(x_i) \propto \prod_{a\in N(i)} m_{a\to i}(x_i) \text{ and } b_a(\mathbf{x}_a) \propto f_a(\mathbf{x}_a) \prod_{i\in N(a)} n_{i\to a}(x_i). \tag{3}$$

Observations are incorporated into the process via δ-functions as local potential for each node in **E**. When that is done, $b_i(x_i)$ becomes the approximation of the posterior probability $P(x_i|\mathbf{e})$.

One of the attractive features of belief propagation for our application is that the algorithm is naturally suited for parallel and/or distributed implementation, which can be quite helpful if we wish to perform diagnosis of a distributed system in a distributed way, e.g. to avoid computational bottleneck, as well as to improve the robustness of monitoring and diagnosis by avoiding single point of failure represented by a central monitoring server. Namely, we can designate several "diagnostic nodes" in the distributed computer system to perform inferences on subsets of probes (and thus associated with one or more factors in a factor graph) and exchange probabilistic messages between each other, effectively running belief propagation in a distributed way. For details on architecture and implementation of such approach, see [11].

4 Lower bound on Diagnostic Error

We will now derive a lower bound on diagnostic error when using bit-wise most-likely diagnosis (measured as the *bit error rate (BER)*), for bipartite Bayesian networks defined above, and particularly for noisy-OR bipartite networks.

The bit-error rate (BER) of diagnosis can be defined as $BER = \frac{\sum_{i=1}^{n} P(X_i \neq X_i'(\mathbf{T}))}{n}$, where $X_i'(\mathbf{T}) = \arg\max_x P(X_i = x|T)$ is the most-likely assignment to X_i given observed vector $\mathbf{T}$. Note that $X_i'(\mathbf{T})$ is a deterministic function if a deterministic tie-breaking rule is used for most-likely assignment (e.g., $X_i' = 0$ if $\mathrm{P}(X_i = 0|T) = 0.5$).

Theorem 1. *Given a bipartite Bayesian network that defines a joint distribution $P(\mathbf{x}, \mathbf{t})$ as specified by the equation 1, the bit error rate (BER) of bit-wise most-likely diagnosis is bounded from below as follows*

$$BER \geq L_{BER} = 1 - p_{max}(\alpha_0 + \alpha_1)^c, \tag{4}$$

where $c = \max_i |ch_i|$, $|ch_i$ being the number of X_i's children, $p_{max} = \max_i \max_{j\in\{0,1\}} P(X_i = j)$ is the maximum prior probability over all nodes, and $\alpha_k = \max_{j\in\{1,\dots,m\}} \max_{\mathbf{pa_j}(t_j)} P(t_j = k|\mathbf{pa_j}(t_j))$ is the maximum conditional probability of the test outcome $k \in \{0, 1\}$, over all test variables and over all assignments to their corresponding parent nodes.

Proof. Without loss of generality, we will first compute a bound on $BER(X_1) = P(X_1 \neq X'_1(\mathbf{T}))$, and then use it to compute the BER. Let $I(s)$ denote the *indicator function* for Boolean argument s, i.e. $I(s) = 1$ if s is $true$ and $I(s) = 0$ otherwise. Then

$$BER(X_1) = P(X_1 \neq X'_1(\mathbf{T})) = \sum_{\mathbf{x},\mathbf{t}} P(\mathbf{x},\mathbf{t}) I(x_1 \neq x'_1(\mathbf{t})) = 1 - \sum_{x_1,...,x_n} \sum_{\mathbf{t}} P(x'_1, x_2, ..., x_n, \mathbf{t}).$$

By definition, $P(x'_1, x_2, ..., x_n, \mathbf{t}) = \arg\max_{x_1} \left\{ P(x_1) \prod_{i=2}^{n} P(x_i) \prod_{j=1}^{m} P(t_j|\mathbf{pa_j}) \right\}$; grouping together factors that involve x_1, we get

$$\begin{aligned} P(x'_1, x_2, ..., x_n, \mathbf{t}) &= \arg\max_{x_1} \{ P(x_1) \prod_{t_j \in ch_1} P(t_j|\mathbf{pa_j}) \} [\prod_{i=2}^{n} P(x_i) \prod_{t_j \notin ch_1} P(t_j|\mathbf{pa_j})] \leq \\ &\leq p_{max} \alpha_0^r \alpha_1^{|ch_1|-r} [\prod_{i=2}^{n} P(x_i) \prod_{t_j \notin ch_1} P(t_j|\mathbf{pa_j})], \end{aligned}$$

where r is the number of 0's in the assignment to a subset of tests $t_j \in ch_1$, and $|ch_1|$ is the number of children of X_1. Then

$$\begin{aligned} BER(X_1) &= 1 - \sum_{x_1,...,x_n} \sum_{\mathbf{t}} P(x'_1, x_2, ..., x_n, \mathbf{t}) \geq \\ &\geq 1 - \sum_{t_j \in ch_1} p_{max} \alpha_0^r \alpha_1^{|ch_1|-r} \sum_{i=2}^{n} P(x_i) \sum_{t_j \notin ch_1} [\prod_{i=2}^{n} P(x_i) \prod_{t_j \notin ch_1} P(t_j|\mathbf{pa_j})] = \\ &= 1 - p_{max} \sum_{r=0}^{|ch_1|} \binom{|ch_1|}{r} \alpha_0^r \alpha_1^{|ch_1|-r} = 1 - p_{max}(\alpha_0 + \alpha_1)^{|ch_1|}. \end{aligned}$$

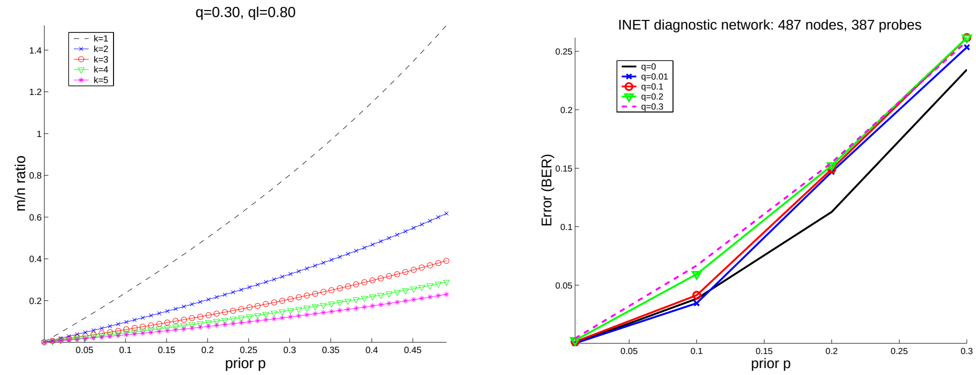
Similarly expression can be obtained for each $BER(X_i)$. Note that $1 \geq \alpha_0 + \alpha_1 \leq 2$, and thus $\min_i BER(X_i) \geq 1 - p_{max}(\alpha_0 + \alpha_1)^c$ where $c = max_i|ch_i|$. Therefore, $BER = \frac{\sum_{i=1}^{n} P(X_i \neq X'_i(\mathbf{T}))}{n} \geq \min_i BER(X_i) = 1 - p_{max}(\alpha_0 + \alpha_1)^c$. □

We can now derive a specific lower bound for noisy-OR networks. To simplify our analysis, let us assume a particular structure that we will call a *(k,c)-regular* bipartite graph, where each node in the lower layer has exactly k parents in the upper layer, and each node in the upper layer has $c = km/n$ children in the lower layer (recall that there are n nodes in the upper layer and m nodes in the lower layer).

Corollary 2. *Given a Bayesian network having the* (k,c)-regular *bipartite graph structure, where n is the number of hidden nodes, m is the number of tests, and where all conditional probabilities $P(t_j|\mathbf{pa(t_j)})$ are noisy-OR functions having the link probability at least q and the leak probability at most q_{leak}, the bit error rate (BER) of bit-wise most-likely diagnosis is bounded from below as follows*

$$BER \geq L_{BER}^{NOR} = 1 - p_{max}(1 + q_{leak}(1 - q^k))^{km/n}. \tag{5}$$

Proof. Note that $\alpha_0 = \max_{j \in \{1,...,m\}} \max_{\mathbf{pa_j}(t_j)} P(t_j = 0|\mathbf{pa_j}(t_j)) = q_{leak}$ is achieved when all parents $\mathbf{pa_j}$ are 0's, and $\alpha_1 = \max_{j \in \{1,...,m\}} \max_{\mathbf{pa_j}(t_j)} P(t_j = 1|\mathbf{pa_j}(t_j)) = 1 - q_{leak} q^k$ is achieved when all parents $\mathbf{pa_j}$ are 1's. Then the result easily follows from the previous theorem. □



(a) (b)

Figure 2: (a) A lower bound on rate m/n necessary for achieving zero-error diagnosis, plotted the versus fault prior p, for different probe length k. (b) Empirical results for belief propagation on realistic Bayesian network obtained for Internet-like topologies: the effect of prior fault probability and noise in probes on the diagnostic error.

Corollary 3. *Given a bipartite Bayesian network that defines a joint distribution $P(\mathbf{x}, \mathbf{t})$ as specified by the equation 1, a necessary condition for achieving error-free bit-wise diagnosis is*

$$L_{BER} \leq 0 \leftrightarrow c \geq \frac{\log(1/p_{max}}{\log(\alpha_0 + \alpha_1)}, \tag{6}$$

where c, α_0 and α_1 are defined as in Theorem 2. Particularly, for noisy-OR networks defined in Corollary 5, the necessary condition is

$$L^{NOR}_{BER} \leq 0 \leftrightarrow \frac{m}{n} \geq \frac{\log(1/p_{max}}{k \log(1 + q_{leak}(1 - q^k))}. \tag{7}$$

Assuming equal prior fault probabilities $p = P(X_i = 1)$, where $p < 0.5$ (typically, system's components are unlikely to be faulty), we get $\frac{m}{n} \geq \frac{\log(1/(1-p)}{k \log(1+q_{leak}(1-q^k))}$. In Figure 2a, we illustrate the growth of the lower bound on rate m/n with the increasing prior fault probability p, for different probe sizes k, and for a fixed noise parameters. As expected, higher probe to node ratio is necessary for higher fault probability p. Also, somewhat intuitively, longer probes (larger k) allow to reduce the required number of probes per node. However, as we will see in empirical section, this is often not the case in practice, which indicates that the bound is not tight, and indeed provides only necessary, but not sufficient, conditions[4].

5 Empirical Results

We performed initial experiments with belief propagation diagnosis on two classes of Bayesian networks: random bipartite graphs and realistic networks obtained by simulating end-to-end probing on Internet-like topologies built by INET generator [12].

[4]Ideally, one would like to provide an analog of Shannon limit for a constrained code that only permits disjunctive codes, and a particular type of channel defined by noisy-OR model. Namely, one would like to know if asymptotically error-free diagnosis is actually achievable at finite rate m/n, and under what conditions on prior p, noise parameters, and probe set construction. While there is a large amount of related work in the area of group testing (e.g., see [4]), this particular setting does not seem to be studied before. Moreover, taking into account constraints on probe construction (e.g., due to the network topology restrictions) makes the analysis much more complicated.

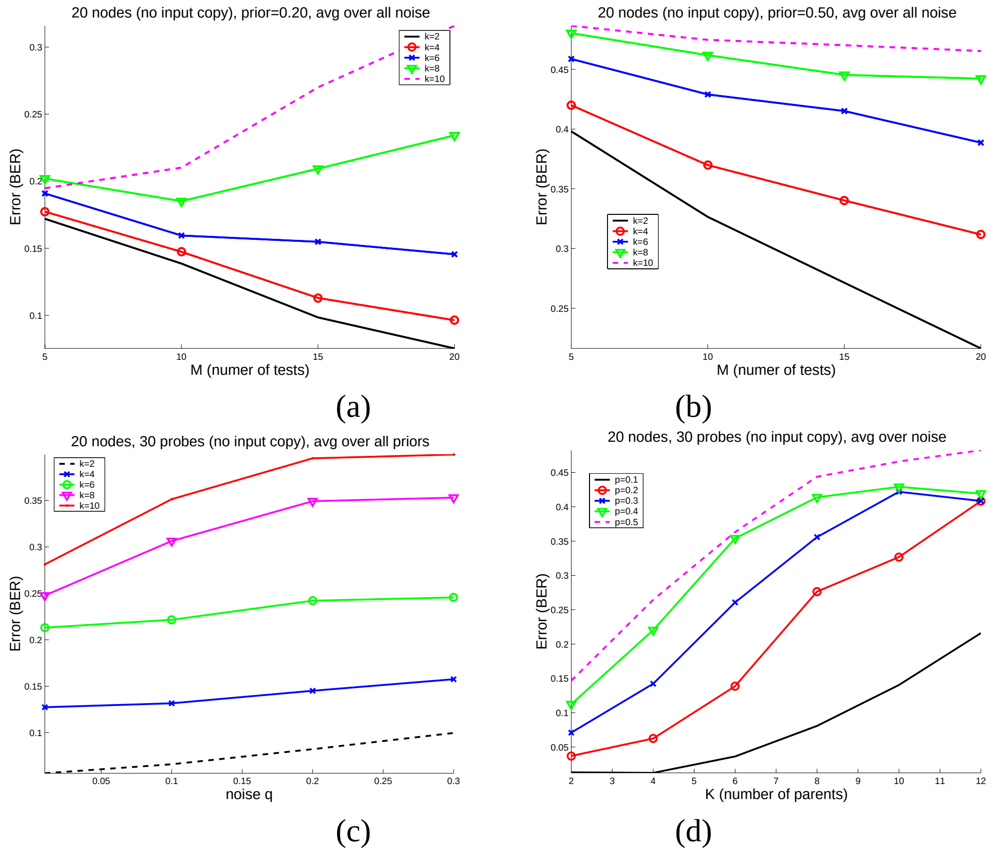


Figure 3: Diagnostic error of belief propagation on randomly generated bipartite networks with 20 nodes, and varying number of probes m, probe length k, and noise q.

Internet-like topologies. Realistic Bayesian network were created as follows. First, we generated computer network topology using INET generator [12], which simulates an arbitrary-size Internet-like topology (at the Autonomous Systems level) by enforcing a power-law node degree distribution that coincides with the one empirically observed for the Internet. Next, we were generating probe sets for fault detection and fault diagnosis, as described in [7]. A detection probe set needs to cover all network components, so that at least one probe has a positive probability of returning 1 when a component fails. A diagnosis probe set needs to further distinguish between faulty components. Optimal probe set design is NP-hard for either detection or diagnosis [7], but the greedy approaches based on selecting most-informative probe next appear to work quite well in practice, and were used here for creating probe sets. For each probe set, we construct the corresponding bipartite Bayesian network, and simulate different prior fault probabilities and different levels of noise.

Figure 2b presents the result of running belief propagation on a Bayesian network built on top of INET-topology with 487 nodes and 387 probes sufficient for single-fault diagnosis (single-fault assumptions are made in [7] to simplify the probe selection process). We plot the bit-error rate against the prior fault probability p at each node, for different levels of noise q. As expected, the error increases with growing probability of fault p and noise level q. Note that the error is quite low, especially for low values of p, which are more realistic (e.g., in a reliable system, we may not expect more than $p = 0.1$ probability of fault, which yields BER less than 0.05). Interestingly, the fault probability has much more impact on the error than the noise level.

Random bipartite networks. In order to investigate the effect of different network properties on the diagnostic error, we also experimented with randomly generated networks, where we had a full control over the parameters. Namely, we generated random bipartite graphs with parameters n, m, k, p, q and q_leak, where n is the number of components, or hidden nodes ("input bits"), m is the number of tests ("code bits"), k is the number of randomly selected parents of a test node[5], i.e. probe length, p is the prior probability of a fault, q and q_{leak} are the inhibition ("noise") and the "leak" (hidden cause inhibition) parameters, respectively, in the noisy-OR model.

Figure 3 plots the diagnostic accuracy of belief propagation, measured as the bit-error rate, on random bipartite networks with $n = 20$ nodes and varying m, k, and q; for these experiments, we assumed there are no additional hidden causes that can affect the probe outcome, thus setting the "leak" inhibition probability to $q_leak = 1$ (always inhibited hidden cause).

Figures 3a and 3b plot the average error, where averaging is performed over several values of noise (particularly, $q = 0.1, 0.2, 0.3, 0.4$ and 0.5), against the increasing number of probes m, for two particular values of prior fault probability, $p = 0.2$ (Figure 3a) and $p = 0.5$ (Figure 3b), and for different probe length $k = 2, 4, 6, 8, 10$. As expected, we observe that, the error tends to decrease with increasing number of probes, for all values of k at higher prior $p = 0.5$, and for almost all values of k when $p = 0.2$, except for cases of $k = 8$ and $k = 10$. Also, as expected, the errors are generally much lower for lower fault prior p. However, a much less intuitive result is that the error seem to be consistently lower for smaller values of k, i.e. for shorter probes.

Another view of the results, now for fixed number of probes $m = 30$, is shown in Figures 3c and 3d. Namely, Figure 3c plots the error averaged over different fault priors ($p = 0.10.2, 0.3, 0.4$ and 0.5), versus the noise q, for different probe length k, while Figure 3d plots the error averaged over different noise levels versus the number of probes k, for different priors. As expected, error increases with increasing noise level and prior fault probability. Again, we clearly observe the not so intuitive dependence of error on the probe length k: the error is clearly lower for smaller k. This behavior is not explained by the error bound we provided in the previous section, and requires further theoretical explanation.

6 Conclusions

Automated diagnosis is a challenging problem that arises in various application domains, from medicine to probabilistic decoding. In this paper, we consider the problem of cost-efficient diagnosis in distributed computer systems via end-to-end test transactions, or probes, which can be viewed as disjunctive tests over subsets of system components. In the presence of noise in the probe outcomes, our problem can be formulated as probabilistic diagnosis of unobserved variables in a noisy-OR Bayesian network. We derive a lower bound on diagnostic error in such networks, measured by bit-error-rate (BER), and also provide experimental evaluation on various noisy-OR networks. Due to computational complexity of exact inference in our networks, approximate techniques must be used. An appealing candidate is belief propagation (BP), a popular approximate inference algorithm successfully used in probabilistic decoding, e.g., for LDPC and Turbo-codes; moreover, BP allows a naturally distributed implementation since it is based on local message-passing; this is a particularly attractive feature that allows a distributed monitoring/diagnosis and eliminates the computational bottleneck associated with a central monitoring server. We provide experimental results for BP diagnosis on various

[5]We did not enforce the (k, c)-regularity condition defined before, i.e., the number of children varied for different hidden nodes X_i, however, it is easy to see that the average number of children in such random graph is still km/n, as in (k, c)-regular graphs.

noisy-OR networks, both randomly generated and realistic ones, derived from Internet-like topologies. The results for realistic networks are quite promising, showing low diagnostic error, especially when prior fault probabilities are sufficiently low. Experiments with randomly generated networks lead to an interesting observation that shorter probes lead to more accurate diagnosis, which remains to be explained theoretically.

Acknowledgements

I would like to thank Natalia Odintsova, Shang Guo, David Loewenstern and Vittorio Castelli for many insightful discussions that contributed to the ideas of this paper.

References

[1] G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2–3):393–405, 1990.

[2] R. Dechter. Bucket elimination: A unifying framework for probabilistic inference. In *Proc. Twelfth Conf. on Uncertainty in Artificial Intelligence*, pages 211–219, 1996.

[3] R. Dechter and I. Rish. Mini-buckets: A General Scheme for Approximating Inference. *J. of ACM*, (50(2)):107–153, 2003.

[4] D-Z. Du and F.K. Hwang. *Combinatorial Group Testing and Its Applications (2nd edition)*. World Scientific, 2000.

[5] B.J. Frey and D.J.C. MacKay. A revolution: Belief propagation in graphs with cycles. *Advances in Neural Information Processing Systems*, 10, 1998.

[6] D. Heckerman. A tractable inference algorithm for diagnosing multiple diseases. In *Proc. Fifth Conf. on Uncertainty in Artificial Intelligence*, pages 174–181, 1989.

[7] S. Ma N. Odintsova A. Beygelzimer G. Grabarnik K. Hernandez I. Rish, M. Brodie. Adaptive diagnosis in distributed systems. *To appear in the IEEE TNN Special Issue on Adaptive Learning Systems in Comm Networks*, 2005.

[8] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report TR-2004-040, MERL, May 2004.

[9] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Mateo, California, 1988.

[10] I. Rish, M. Brodie, and S. Ma. Accuracy vs. Efficiency Trade-offs in Probabilistic Diagnosis. In *Proceedings of AAAI-2002, Edmonton, Alberta, Canada*, pages 560–566, 2002.

[11] D. Loewenstern N. Odintsova S. Guo, I. Rish. Self-healing in large-scale systems: parallel and distributed diagnostic architectures. Technical report, IBM T.J. Watson Research Center, 2005.

[12] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. Technical Report CSE-TR-456-02, University of Michigan, 2002.