

# IBM Research Report

## Entropy Approximation for Active Fault Diagnosis

**Alice X. Zheng**  
Department of EECS  
University of California at Berkeley  
Berkeley, CA 94720-1776

**Irina Rish, Alina Beygelzimer**  
IBM Research Division  
Thomas J. Watson Research Center  
P.O. Box 704  
Yorktown Heights, NY 10598



Research Division  
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

---

# Entropy Approximation for Active Fault Diagnosis

---

**Alice X. Zheng**

Department of EECS  
U. C. Berkeley  
Berkeley, CA 94720-1776  
alicez@eecs.berkeley.edu

**Irina Rish**

IBM T.J. Watson Research Center  
19 Skyline Drive  
Hawthorne, NY 10532  
rish@us.ibm.com

**Alina Beygelzimer**

IBM T.J. Watson Research Center  
19 Skyline Drive  
Hawthorne, NY 10532  
beygel@us.ibm.com

## Abstract

We address the problem of test selection for fault diagnosis on a Bayesian network, which requires several entropy terms whose exact computation is intractable. We propose an approximate approach that utilizes the loopy belief propagation infrastructure to simultaneously compute approximations of marginal and conditional entropies on multiple subsets of nodes. We apply the method to active probing for fault diagnosis in computer networks, and present promising empirical results on realistic Internet-like topology graphs.

## 1 Introduction

The problem of fault diagnosis appears in many places under various guises. Examples include medical diagnosis, computer system troubleshooting, and decoding messages sent through a noisy channel. Recently, researchers have formulated diagnosis as an inference problem on a Bayesian network, with the goal of assigning most-likely states to unobserved nodes based on outcome of certain “test” nodes.

An important issue in diagnosis is the trade-off between the cost of performing tests and the achieved accuracy of diagnosis. It is often too expensive or even impossible to perform all tests. In this paper, we concentrate on problem of *active* diagnosis, in which tests are selected sequentially to minimize the cost of testing (e.g., the number of test if their costs are equal). We use entropy as the cost function and select a set of tests providing maximum information (i.e. minimal conditional entropy) about the unknown variables.

However, exact computation of conditional entropies in a general Bayesian network can be intractable. While much existing research has addressed the prob-

lem of efficient and accurate probabilistic inference, other probabilistic quantities, such as conditional entropy and information gain, have not received nearly as much attention. There is a vast amount of literature on value of information and most-informative test selection [8, 3, 7, 17], but none of the previous work appears to focus on the computational complexity of most-informative test selection in a general Bayesian network setting.

It can be easily shown that, while entropy is decomposable (and thus tractable) on a Bayesian network, marginal conditional entropy (and thus information gain) is in general non-decomposable and requires performing possibly intractable probabilistic inference. We propose an approximation algorithm for computing marginal conditional entropy, which is based on loopy belief propagation, a successful approximate inference method. We illustrate the algorithm at work in the setting of active probing for fault diagnosis. Our method is general enough to apply to other applications of Bayesian networks that require the computation of information gain and conditional entropies of subsets of nodes. In our specific case, it can efficiently compute the information gain for all candidate tests simultaneously.

In section 2, we give the general problem setup for entropy approximation in a Bayesian network, followed by our solution in section 3. Section 4 gives an overview of the active probing application framework, while section 5 presents a detailed analysis of empirical results. We survey related work in section 6, and conclude in section 7.

## 2 Problem Setup

Let  $\mathbf{X} = \{X_1, X_2, \dots, X_N\}$  denote the set of  $N$  discrete random variables and  $\mathbf{x}$  a possible realization of  $\mathbf{X}$ . Assume that a joint probability distribution (PDF) over  $\mathbf{X}$  is encoded by a Bayesian network that yields

a product-form representation

$$P(\mathbf{x}) = \prod_{i=1}^n P(x_i | \mathbf{pa}_i), \quad (1)$$

where  $\mathbf{pa}_i$  is an assignment to the parents of  $X_i$  (nodes pointing to in the Bayesian network).

We adopt the *factor graph* representation [5], which was recently proposed as a convenient way of unifying both directed-graph (Bayesian networks) and undirected-graph (Markov networks) representations of joint PDFs. The PDF is assumed to be written in a factored form

$$P(\mathbf{x}) = \frac{1}{Z} \prod_a f_a(\mathbf{x}_a), \quad (2)$$

where the index  $a$  ranges over all *potential functions*, or *factors*  $f_a(\mathbf{x}_a)$ , defined on the corresponding subsets  $\mathbf{x}_a$  of  $\mathbf{X}$ .

Let  $\mathbf{E}$  be a possibly empty set of evidence nodes for which observation is available, and let  $a$  be the index of an arbitrary factor. We concentrate on the problem of estimating the conditional marginal entropy of  $\mathbf{X}_a$  given a set of observations  $\mathbf{e}$ . It is easy to show that the entropy  $H(\mathbf{X})$  is decomposable over the set of families and thus yields tractable computation exponential only in the largest family size. Unfortunately, this is not true for computing conditional marginal entropies:

$$H(\mathbf{X}_a | \mathbf{e}) = \sum_{\mathbf{x}_a} P(\mathbf{x}_a | \mathbf{e}) \log P(\mathbf{x}_a | \mathbf{e}) \quad (3)$$

$$\text{where } P(\mathbf{x}_a | \mathbf{e}) = \sum_{\mathbf{x} \setminus \mathbf{x}_a} P(\mathbf{x} | \mathbf{e}). \quad (4)$$

This computation is not decomposable and requires calculating  $P(\mathbf{x}_a | \mathbf{e})$ , which is an NP-complete probabilistic inference problem [1]; commonly used exact inference algorithms require time and space exponential in the *treewidth*[16] of the graph, which corresponds to the size of largest clique induced by inference.

## 2.1 Drawing from ideas of belief propagation

Although probabilistic inference can be intractable in general case, there exists a simple linear-time message-passing algorithm, known as belief propagation (BP) [16], which is provably correct on polytrees (i.e. Bayesian networks with no undirected cycles), 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 BP is said to diverge.

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 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 [11]:

$$n_{i \rightarrow a} := \prod_{c \in N(i) \setminus a} m_{c \rightarrow i}(x_i), \quad (5)$$

and

$$m_{a \rightarrow i}(x_i) := \sum_{\mathbf{x}_a \setminus x_i} f_a(\mathbf{x}_a) \prod_{j \in N(a) \setminus i} n_{j \rightarrow a}(x_j) \quad (6)$$

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

$$b_i(x_i) \propto \prod_{a \in N(i)} m_{a \rightarrow i}(x_i) \quad (7)$$

$$b_a(\mathbf{x}_a) \propto f_a(\mathbf{x}_a) \prod_{i \in N(a)} n_{i \rightarrow a}(x_i). \quad (8)$$

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

One might imagine directly approximating the entropy, modifying the messages to update entropy as opposed to probabilities. More generally, one might imagine such an algorithm for calculating general functions of the probability. However, our initial forays along this line are met with firm resistance from the recursive characteristics of message passing. That is, previous messages would be incorporated into the current estimate of the entropy, which would be used to formulate new messages. It is difficult to control and normalize such recursive updates for complicated functions of the probability.

We settle for a direct usage of beliefs resulting from the original belief propagation mechanism. The efficiency of belief propagation enables us to calculate marginal conditional entropies for multiple subsets of the Bayesian network in one single iteration.

### 3 Belief Propagation for Entropy Approximation (BPEA)

Recall our initial problem of approximating the marginal conditional entropy in Eqn. (3). The trick is to replace the marginal posterior  $P(\mathbf{x}_a|\mathbf{e})$  with its factorized BP approximation, and make use of the BP message passing mechanism to perform the summation over  $\mathbf{x}_a$ . We call this process Belief Propagation for Entropy Approximation (BPEA, pronounced “bee-pee”).

First, pick a node  $X_0$  from the family  $\mathbf{X}_a$  to be the designated root node. We modify the final message passed to  $X_0$  (i.e., the message passed on the last iteration) as follows:

$$m'_{a \rightarrow 0}(x_0) := - \sum_{\mathbf{x}_a \setminus x_0} \tilde{b}_a(\mathbf{x}_a) \log \tilde{b}_a(\mathbf{x}_a), \quad (9)$$

where

$$\tilde{b}_a(\mathbf{x}_a) := f_a(\mathbf{x}_a) \prod_{c \in N(0) \setminus a} m_{c \rightarrow 0}(x_0) \prod_{j \in N(a) \setminus 0} n_{j \rightarrow a}(x_j). \quad (10)$$

$\tilde{b}_a(\mathbf{x}_a)$  is essentially the unnormalized belief of  $\mathbf{X}_a$ , in which we have included the messages gathered at the edges of all the nodes in the family  $a$ .  $m'_{a \rightarrow 0}(x_0)$  includes a partial sum of the entropy term. To get the entire entropy, it still remains to sum over the root node  $X_0$ :

$$\tilde{h}(\mathbf{X}_a|\mathbf{e}) := \sum_{x_0} m'_{a \rightarrow 0}(x_0). \quad (11)$$

$\tilde{h}(\cdot)$  is almost the approximate entropy we seek, except for one thing. Recall that  $\tilde{b}_a(\mathbf{x}_a)$  is the *unnormalized* version of the belief, i.e.,  $\tilde{b}_a(\mathbf{x}_a) = \sigma b_a(\mathbf{x}_a)$  where  $\sigma = \sum_{\mathbf{x}_a} \tilde{b}_a(\mathbf{x}_a)$  is the normalization constant that makes  $\sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) = 1$ . We need to normalize  $\tilde{h}$  in order to obtain the correct entropy.

$$h(\mathbf{X}_a|\mathbf{e}) := \frac{\tilde{h}(\mathbf{X}_a|\mathbf{e})}{\sigma} + \log \sigma \quad (12)$$

**Lemma 1.** *BPEA is exact whenever BP is exact.*

**Proof.** The lemma follows trivially once we show that  $\tilde{b}_a(\mathbf{x}_a)$  is the unnormalized belief  $b_a(\mathbf{x}_a)$  and that Eqn. (12) achieves the correct normalization.

Plugging Eqn. (5) into Eqn. (8) and separating the terms involving the root node from the rest, we obtain

$$\begin{aligned} b_a(\mathbf{x}_a) &\propto f_a(\mathbf{x}_a) \prod_{i \in N(a)} \prod_{c \in N(i) \setminus a} m_{c \rightarrow i}(x_i) & (13) \\ &= f_a(\mathbf{x}_a) \prod_{c \in N(0) \setminus a} m_{c \rightarrow 0}(x_0) \prod_{j \in N(a) \setminus 0} n_{j \rightarrow a}(x_j) & (14) \\ &= \tilde{b}_a(\mathbf{x}_a) & (15) \end{aligned}$$

To prove the correctness of the normalization in Eqn. (12), note that, for any function  $g(x)$  and any constant  $\omega$ ,  $-\sum_x \omega g(x) \log \omega g(x) = -\omega \sum_x g(x) \log g(x) - \omega \log \omega \sum_x g(x)$ . In our case,  $g(\mathbf{x}) = b_a(\mathbf{x}_a)$ , which sums to one. Eqn. (12) thus follows.  $\square$

#### 3.1 Computational efficiency

The normalization constant  $\sigma$  is already computed during normal BP iterations. The computation of  $\tilde{b}_a(\cdot)$ ,  $m'_{a \rightarrow i}$ , and  $\tilde{h}(\cdot)$  can all be piggy-backed onto the same BP infrastructure, and does not impact its overall complexity. What’s more, due to the local and parallel message update procedure in BP, we can compute the marginal posterior entropies of multiple families in the graph in one single sweep. This will turn out to be important in our application to the *active probing* problem considered below.

## 4 Fault Diagnosis and Active Probing

Modern day Internet and company-wide intranets have expanded to sizes beyond the capabilities of real-time manual supervision. For a typical quality of service agreement of 99.9% uptime, the computer system can be down for no more than 10 minutes a week. This calls for some form of automatic, computer-aided failure detection and diagnosis. In previous work, Rish et.al. [9][13] formulate automatic fault diagnosis as an inference problem on a QMR-DT-like[12] Bayesian network.

Suppose we wish to monitor a system of networked computers. Let  $\mathbf{S} = \{S_1, S_2, \dots, S_N\}$  denote a set of binary random variables representing the state of  $N$  network elements.  $S_i = 0$  indicates that the element is in normal operation mode, and  $S_i = 1$  indicates that the element is faulty.  $S_i$  could represent hardware, software, or network aspects of the system, or any system’s component whose state can be measured using a suite of tests. Each component can be in one of two states, either 0 (ok) or 1 (faulty). In a large system, it is often impossible to test each individual component directly. A common solution is to test a *subset* of components with a single probe. Such a test reveals either that all components in the subset are ok (in which case its outcome is 0) or that at least one of the components is faulty (if the outcome is 1), but not how many or which are faulty.

We assume there are machines designated as *probe stations*, that are instrumented to send out *probes* (test transactions) to test the response of the network elements represented by  $\mathbf{S}$ . Let  $\mathbf{T} = \{T_1, T_2, \dots, T_M\}$  denote the available set of probes. A probe can be as

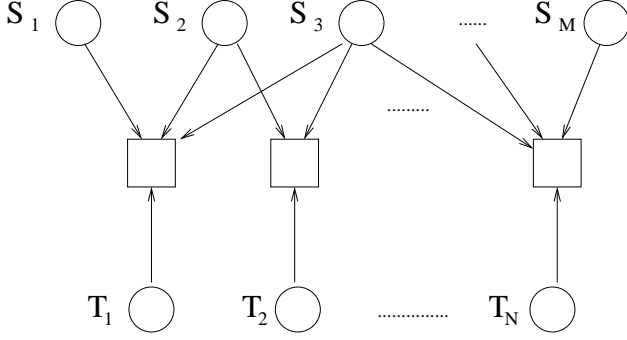


Figure 1: The fault diagnostic Bayes net.

simple as a *ping* request, which detects network availability, or a more sophisticated test such as an e-mail message and a webpage-access request.

Figure 1 shows a factor graph representation of our Bayesian network. In the absence of noise a probe is a disjunctive test, i.e. it fails if any of the nodes along its path is down. A natural generalization is to use the well-known noisy-OR model [16]:

$$P(s_j) := (\alpha_j)^{s_j} (1 - \alpha_j)^{(1-s_j)}, \quad (16)$$

$$P(t_i = 0 | \mathbf{s}_{pa(i)}) := \rho_{i0} \prod_{j \in pa(i)} \rho_{ij}^{s_j}, \quad (17)$$

$$P(\mathbf{s}, \mathbf{t}) := \prod_i P(t_i | \mathbf{s}_{pa(i)}) \prod_j P(s_j). \quad (18)$$

Here,  $pa(i)$  denote the indices of the nodes in  $\mathbf{S}$  nodes covered by probe  $i$ ,  $\alpha_j := P(s_j = 1)$  is the prior fault probability,  $\rho_{ij}$  is the so-called inhibition probability, and  $(1 - \rho_{i0})$  is the leak probability of an unaccounted-for faulty element. The inhibition probability is a measurement of the amount of noise in the network; the higher the inhibition, the higher the chance that a probe will not observe an actual fault.

We model the problem of fault diagnosis as that of finding the most probable configuration of the nodes in  $\mathbf{S}$ , given probe outcomes. The quality of real-time fault diagnosis not only depends on its correctness – that is, whether or not we correctly identify the faulty elements – but also requires the diagnosis be made as quickly as possible, using the fewest number of potentially expensive probes. The situation is similar to that of medical diagnosis and problem determination.

To minimize the number of probes used, Rish et.al. [10][14] formulate the *active probing* framework. We sequentially select probes based on their information gain. Given the observed outcome  $\mathbf{t}'$  of previously selected probes, the next probe is taken to be

$$\operatorname{argmax}_T I(\mathbf{S}; T | \mathbf{t}'), \quad (19)$$

where  $I(\mathbf{S}; T | \mathbf{t}')$  is a conditional mutual information.

For probe selection purposes, we can distill Eqn. (19) down to only the elements involving  $T$ .

$$I(\mathbf{S}; T | \mathbf{t}') = H(\mathbf{S} | \mathbf{t}') - H(\mathbf{S} | T, \mathbf{t}') \quad (20)$$

$$= \text{const.} - H(\mathbf{S} | T, \mathbf{t}'), \quad (21)$$

where

$$H(\mathbf{S} | T, \mathbf{t}') = - \sum_{t, \mathbf{s}_{pa(T)}} P(\mathbf{s}_{pa(T)}, t | \mathbf{t}') \log P(t | \mathbf{s}_{pa(T)}) + \sum_t P(t | \mathbf{t}') \log P(t | \mathbf{t}') + \text{const.} \quad (22)$$

Eqn. (21) shows that maximizing the information gain is equivalent to minimizing the conditional entropy  $H(\mathbf{S} | T, \mathbf{t}')$ . The latter boils down to only two terms involving  $T$ . Let  $A(T, \mathbf{S}_{pa(T)} | \mathbf{t}')$  denote the first term in Eqn. (22); this is the cross entropy between the posterior probability of  $T$  and its parents, and the conditional probability of  $T$  given its parents. The second term in Eqn. (22) is simply a conditional entropy,  $H(T | \mathbf{t}')$ .

In previous work, a single-fault assumption was made, which effectively reduced  $\mathbf{S}$  to one random variable with  $N + 1$  possible states. In general, however, multiple faults could exist in the system simultaneously, which requires in the representation above. Since observations of probe outcome correlate the parent nodes, the exact computation of the posterior probabilities in both entropy terms is intractable.

We deal with the two terms separately. For  $H(T | \mathbf{t}')$ , we may use approximation methods such as BP or GBP to calculate the belief  $b(t | \mathbf{t}')$ , which can then be used to directly compute  $H(T | \mathbf{t}')$ . (Note that the summation over values of  $T$  is simple since  $T$  is binary-valued.) To calculate  $A(T, \mathbf{S}_{pa(T)} | \mathbf{t}')$ , we use the entropy approximation method (BPEA) as described in Section 3. Because BP message updates are done locally, we can compute  $A(T, \mathbf{S}_{pa(T)} | \mathbf{t}')$  for all unobserved  $T$  nodes during a single message update iteration of BP. Thus, picking the next probe requires only one run of the BPEA approximation algorithm.

Since the probe node  $T$  is the least connected member of its family, we designate it as the root node. Abusing notation slightly, we use  $t$  to index the factor representing the family of probe node  $T$  and its parents. The modified messages, in this case, reduces to:

$$\tilde{b}_t(t, \mathbf{s}_{pa(t)}) := P(t | \mathbf{s}_{pa(t)}) \prod_{j \in pa(t)} n_{j \rightarrow t}(s_j) \quad (23)$$

$A(T, \mathbf{S}_{pa(T)} | \mathbf{t}') = - \sum_{t, \mathbf{s}_{pa(T)}} P(\mathbf{s}_{pa(T)}, t | \mathbf{t}') \log P(t | \mathbf{s}_{pa(T)})$  is a cross entropy term. Hence we do not take the log of  $\tilde{b}$  during BPEA, but rather take the logarithm of

the known probabilities  $P(t|\mathbf{s}_{pa(T)})$ . This simplifies the normalization step described in Eqn. (12) to:

$$A(T, \mathbf{S}_{pa(T)}|\mathbf{t}') = \frac{\tilde{A}(T, \mathbf{S}_{pa(T)}|\mathbf{t}')}{\sigma},$$

where  $\sigma = \sum_{t, \mathbf{s}_{pa(T)}} \tilde{b}_t(t, \mathbf{s}_{pa(T)})$ .

## 5 Empirical Results

### 5.1 The dataset

We conduct our experiments on network topologies built by the INET generator[19], which simulates Internet-like topology at AS-level. Our dataset includes a set of networks of 485 nodes, where the number of probe stations varies from 1 to 50.

The parent structure of probe nodes  $\mathbf{T}$  are generated with two goals in mind: detection and diagnosis. A detection probe set needs to cover all network components, so that at least one probe has a positive probability of returning false when a component fails. A diagnosis probe set needs to not only cover all components, but be able to distinguish between faulty components. It is NP-hard to design the optimal probe set for either detection or diagnosis. For the datasets used here, we first use a greedy approach to obtain a probe set that covers all network components, then augment this set with additional probes in order to guarantee single-fault diagnosis. Detailed discussions of probe set design for diagnosis Bayesian networks may be found in [17, 18].

Table 1 lists the sizes of all of the networks used in our experiments. In general, fewer probes are needed for detection than for diagnosis. Also, adding more probe stations increases the diversity of probes, and therefore fewer probes are needed to cover the entire network.

Table 1: Number of  $T$  nodes in detection and diagnosis networks. (There are 485  $S$  nodes in all networks.)

#Probe Sts	1	10	20	30	40	50
detection	319	270	263	242	236	227
diagnosis	428	380	371	350	339	324

### 5.2 Parameters and measurements

There are four parameters of interest. The type of network (detection vs. diagnosis) and the number of probe stations both affect the tree-width of the graph, and are therefore interesting. We also measure the effects of the prior fault probability  $\alpha$  and inhibition

probability  $\rho$ .<sup>1</sup>

We measure the quality of approximation of the entropy terms, as well as the diagnostic quality of the selected probe set. For both measurements, we compare against ground truth obtained from the junction tree algorithm for exact inference. A third measurement is computational efficiency, measured in terms of CPU time, which we briefly summarize in subsection 5.5. Since all measurements depend on the particular set of probe outcomes, we repeat all our experiments on 10 different samples of the Bayes net.

We use the diagnostic quality of the probe set to determine when to stop the probe selection process: when the reduction in entropy (Eqn. (25)) for the past 5 iterations is no more than 0.00001, the process is deemed to converge. Otherwise the process runs until all probes have been picked.

### 5.3 Approximation accuracy

We first look at approximation accuracy. Recall that at each time step of the active probing process, we obtain a vector of approximate entropy values, one for each candidate probe  $T$ . We average the relative error between the approximation values and the exact value for unselected probes, and further average over all time steps and samples. Let  $M$  denote the total number of probes,  $n$  the number of selected probes,  $h_{ij}$  the approximate value for probe  $j$  at the  $i$ th stage of probe selection, and  $H_{ij}$  the corresponding exact values. We calculate

$$R(h, H) := \frac{1}{n} \sum_{i=0}^{n-1} \frac{1}{M-i} \sum_{j=1}^{M-i} \frac{|h_{ij} - H_{ij}|}{|H_{ij}|}. \quad (24)$$

First, we fix the network and examine the effects of prior fault and inhibition probabilities on  $R(h, H)$ . We use the detection network with 10 probe stations, and augment this probe set with single-node probes in order to alleviate the insufficiencies of a detection network. Fig. 2(a-b) contains plot of the average, the minimum, and the maximum approximation errors, taken over 10 samples of probe outcomes. Relative error values are shown separately for the two entropy terms in Eqn. (22).  $A(T, \mathbf{S}_{pa(T)}|\mathbf{t}')$  is calculated using BPEA, whereas  $H(T|\mathbf{t}')$  is obtained directly from the BP beliefs  $b(t|\mathbf{t}')$ . The approximation error is lower at lower levels of the prior fault probability. For both values of prior fault probability, and for all levels of inhibition probability, the error does not exceed 2% on average. At the maximum, the approximation error does not exceed 10% for  $A(T, \mathbf{S}_{pa(T)}|\mathbf{t}')$ , and 20% for  $H(T|\mathbf{t}')$ .

<sup>1</sup> For the purpose of evaluating the quality of approximation and diagnosis, we assume one  $\alpha$  and one  $\rho$  values

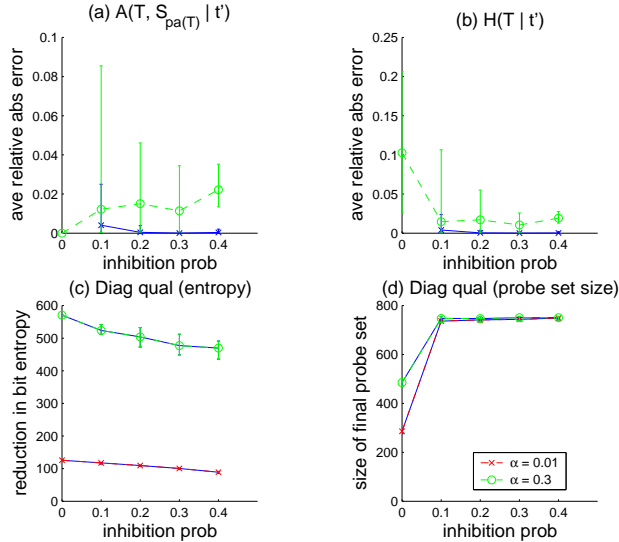


Figure 2: Approximation and diagnostic quality measurements on an augmented detection network, plotted against inhibition probability ( $\rho$ ), and shown at different prior fault probabilities ( $\alpha$ ).

Fig. 3(a-b) shows approximation error plots for detection networks with different numbers of probe stations. For this experiment, we fix  $\alpha = 0.1$  and  $\rho = 0.1$ . Here, again, the average approximation error never exceeds 2% for any number of probe stations, and the overall maximum error does not seem to be affected by the number of probe stations (and hence the coverage pattern and tree-width of the graph).

It would be interesting to perform the same approximation error analysis on the set of diagnosis networks in our dataset. However, due to increased tree-width, the exact method fails to run on all diagnosis networks with more than 1 probe station.

#### 5.4 Diagnostic quality

The quality of diagnosis is taken to be the reduction in conditional bit entropy of the state of the network elements. That is, if  $t'$  represents the observed outcomes of the final set of selected probes, we measure

$$H(\mathbf{S}) - H(\mathbf{S}|t') = - \sum_{\mathbf{s}} P(\mathbf{s}) \log_2 P(\mathbf{s}) + \sum_{\mathbf{s}} P(\mathbf{s}|t') \log_2 P(\mathbf{s}|t') \quad (25)$$

We first examine the effect of inhibition probability and prior fault probability on the quality of diagnosis. Fig. 2(c) plots the diagnostic quality of approximate

for the entire network.

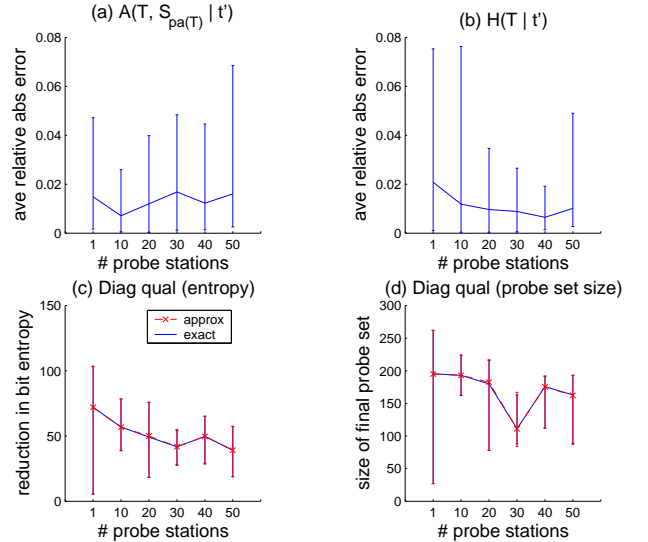


Figure 3: Approximation and diagnostic quality measurements for detection networks with varying numbers of probe stations. ( $\alpha = 0.1$  and  $\rho = 0.1$ )

and exact algorithms obtained on the previously mentioned augmented detection network. Note that, for both levels of prior fault probability and at all levels of inhibition, the two algorithms are practically indistinguishable in terms of diagnostic quality. Fig. 2(d) looks at the size of the final probe set (i.e., the number of probes selected when the active probing process is deemed to converge); here again, the two algorithms have identical behavior.

Experiments with varying the number of probe stations return similar results in terms of diagnostic quality and final probe set size. Fig. 3(c-d) contains the corresponding plots.  $\alpha$  and  $\rho$  are fixed to be 0.1, and the number of probe stations varies from 1 to 50. The approximate and exact methods return curves that practically coincide with each other.

Lastly, we look at quality measurements on the set of diagnosis networks. Since the exact method is not able to handle diagnosis networks with more than 1 probe station, we can only look at the diagnostic quality of the approximate method. Comparing Fig. 4(a) with Fig. 3(c), we see that the diagnosis networks indeed seem to offer slightly better diagnostic quality (in terms of reduction in bit entropy). But, as Fig. 4(b) and Fig. 3(d) shows, this improvement in diagnostic quality comes at the cost of retaining more probes in the active probing process.

#### 5.5 Implementation and speed

We use the junction tree inference engine in Kevin Murphy's Bayes Net Toolbox [15] for Matlab to ob-

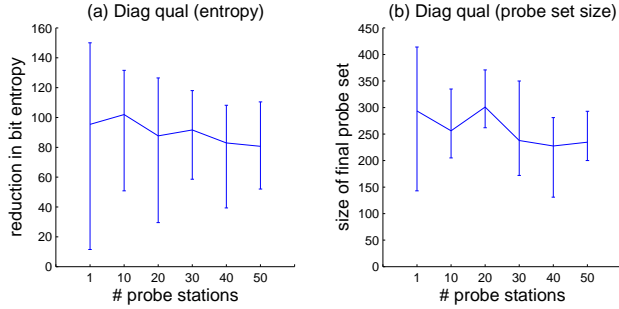


Figure 4: Diagnostic quality measurements for diagnosis networks with varying numbers of probe stations. ( $\alpha = 0.1$  and  $\rho = 0.1$ )

tain exact singleton posterior probabilities. The approximate method is implemented on top of the belief propagation C++/mex code developed by Yair Weiss and Talya Meltzer. We speed up the approximation active probing process by re-using beliefs from previous iterations of BP.

Fig. 5(a) plots the average, maximum, and minimum number of BP iterations that we save through re-using messages. The x-axis denotes the type of network used. The label `diag` represents the diagnosis network with 1 probe station, and the rest are detection networks with varying numbers of probe stations. Note that, on average, re-using messages shortens the BP convergence time by 40-50 iterations. This amounts to substantial savings in computation time over the entire active probing process.

Fig. 5(b) compares computation time of the approximate method to the exact method. On average, the approximate method turns out to be slower. Closer examination of the results show that, for most probe selection steps, BP converges under 10 iterations, which puts the approximate method ahead of the exact method. However, for a few of the probes, BP may take several hundred iterations to converge. Thus the average time requirement (per probe selection) of the approximate method is about 2 seconds longer than the exact method. However, keep in mind that, for networks with larger tree-width, the exact method is simply not feasible. Hence, in general, the approximate method is our only choice.

## 6 Related Work

The most-informative test selection was previously addressed in various work on diagnosis, decision analysis, feature selection in machine learning, and related areas. Given a cost function, a common decision-theoretic approach is to compute the *expected value-of-information* [8] of a candidate test, i.e. the expected

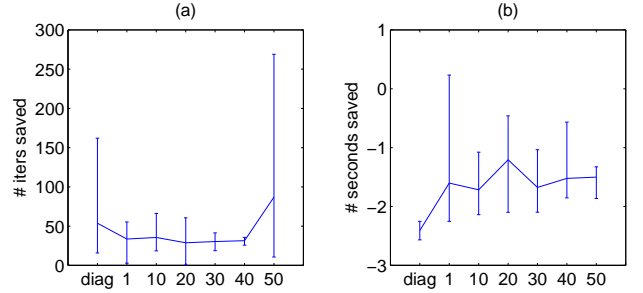


Figure 5: Efficiency of approximate method. (a) Average number of BP iterations saved by re-using messages; (b) Average speed-up compared to exact method.

cost of making a decision after observing the test outcome; using entropy as a cost function yields most-informative test selection. Value of information analysis (and particularly most-informative test selection) was considered in the context of model-based diagnosis [3], probabilistic diagnosis [16] and applied to many practical domains [17]. Previous research has addressed computational complexity of selecting a *set* of most-informative tests instead of a single test [7]. However, none of the previous approaches seem to address the efficiency of computing single-test information gain in a generic Bayesian network.

The most-informative test selection problem is quite similar to the optimal coding problem[2]. There is, however, an important difference. In the coding domain, one may separate source coding (compressing  $\mathbf{S}$ ) from channel coding (adding redundancy to improve decoding accuracy). Fault diagnosis, on the other hand, has to deal with a combination of the two, which manifests itself in the nature of available tests (described by the conditional probabilities  $P(T_i|\mathbf{S}_{pa(i)})$ ). We may have no control over the source coding function, but we can still select the smallest, most informative subset of tests.

In the context of probing, i.e. disjunctive testing, optimal test selection is very similar to the *group testing* problem [4]. Given a set of Boolean variables representing objects that can be in two possible states (i.e. sick vs. healthy patients, failed vs. OK nodes), the objective of group testing is to find all 'failed' objects by using a sequence of disjunctive tests. Particularly, sequential test selection is known as *adaptive group testing* [4]. There is also a direct connection between adaptive group testing and Golomb codes [6]. Note that group testing assumes no constraints on the test selection (i.e., any subset of objects can be tested together), while in Bayesian networks the tests can be only selected from a fixed set. Even in a less restrictive case of probe selection we are still constrained by



the network topology. Constrained group testing (and coding in general) appears to be more complicated, particularly for theoretical analysis, than its unconstrained version.

## 7 Conclusions

We propose an entropy approximation method based on loopy belief propagation, and examine its behavior on the application of active probing for fault diagnosis in a networked computer system. The level of approximation error is found to vary with the level of noise. However, even with non-zero approximation errors, the diagnosis quality is practically identical to that obtained from the exact method. BPEA approximation takes slightly longer than the exact method on small networks. But it can handle much larger networks for which exact junction tree inference is infeasible. This highlights a promising direction for active probing and fault diagnosis, and for entropy approximation on Bayesian networks in general.

## Acknowledgements

The authors wish to thank Kevin Murphy, Yair Weiss, Talya Meltzer, and Natalia Odintsova for making their code available, and Alex Aiken for contributing computation resources. We would also like to thank Sheng Ma and Shang Guo for their suggestions and comments.

## References

- [1] G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2–3):393–405, 1990.
- [2] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley & Sons, 1991.
- [3] J. de Kleer and B.C. Williams. Diagnosing Multiple Faults. *Artificial Intelligence*, 32(1), 1987.
- [4] D-Z. Du and F.K. Hwang. *Combinatorial Group Testing and Its Applications (2nd edition)*. World Scientific, 2000.
- [5] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Info. Theory*, pages 498–519, 2001.
- [6] R. Gallager and D. Van Voorhis. Optimal source codes for geometrically distributed integer alphabets. *IEEE Trans. Information Theory*, IT-21:228–230, 1975.
- [7] D. E. Heckerman, E. J. Horvitz, and B. Middleton. An approximate nonmyopic computation for value of information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:292–298, 1993.
- [8] R. Howard. Information value theory. *IEEE Trans Syst Sci Cybern*, 2(1):22–26, 1966.
- [9] I. Rish, M. Brodie, and S. Ma. Accuracy vs. efficiency trade-offs in probabilistic diagnosis. In *AAAI*, 2002.
- [10] I. Rish, M. Brodie, N. Odintsova, S. Ma, G. Grabarnik. Real-time problem determination in distributed systems using active probing. In *NOMS*, 2004.
- [11] 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.
- [12] T. Jaakkola and M. Jordan. Variational probabilistic inference and the qmr-dt database. *Journal of Artificial Intelligence Research*, pages 291–322, 1999.
- [13] M. Brodie, I. Rish, and S. Ma. Intelligent probing: a cost-efficient approach to fault diagnosis in computer networks. *IBM Systems Journal*, (3):372–385.
- [14] M. Brodie, I. Rish, S. Ma, N. Odintsova. Active probing strategies for problem diagnosis in distributed systems. In *IJCAI*, 2003.
- [15] K. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 2001.
- [16] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Mateo, California, 1988.
- [17] I. Rish, M. Brodie, N. Odintsova, S. Ma, and G. Grabarnik. Real-time Problem Determination in Distributed Systems using Active Probing. In *Proceedings of 2004 IEEE/IFIP Network Operations and Management Symposium (NOMS 2004)*, Seoul, Korea, 2004.
- [18] Irina Rish, Mark Brodie, and Sheng Ma. Intelligent probing: a Cost-Efficient Approach to Fault Diagnosis in Computer Networks. *IBM Systems Journal*, 41(3):372–385, 2002.
- [19] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. Technical Report CSE-TR-456-02, University of Michigan, 2002.