Optimizing a Diagnostic Classifier with Sparse, Unsupervised Data

Abstract

This is exploratory work in building diagnostic models with sparse, unsupervised data that relies on a combination of conventional Bayes networks model structure elicitation, and optimization-based learning methods familiar in current machine learning. We have applied this to building a fault classification model for network failures that occur in the network connecting block storage units and servers that host virtual machines (VM) in a cloud data center.

We have developed a novel approach to learning the model parameters by use solely of the observed signal data. The optimization criterion is the error between the pair-wise conditional probability distribution of the observed variables and their distribution as predicted by the model. We have experimented with different combinations of error functions, more or less sparse bipartite structures, and algorithms for different optimization methods. Once optimization has converged we evaluate the model diagnostic accuracy by running an existing rule-based model on the same data. Gratifyingly the optimized model accuracy shows substantial improvement despite the lack of supervision in the training data.

We've demonstrated that this method on an example where an existing set of procedural rules serve as a gold standard to compare to the model inference. We can achieve high accuracy—this is largely due to the local independence properties of the diagnostic Bayes network—it's possible to optimize faults individually. This work is still initial stages; in this paper we share our insights and speculate how this process may be improved upon.

1 INTRODUCTION

The challenge of building classification models when data is limited and lacks classification labels is not new, and has been the subject of research going back to the origins AI. Bayesian methods promise the ability to combine diverse sources of knowledge when empirical data is limited, yet standard methods to exploit this ability are not widely applied. This is exploratory work in building diagnostic models with sparse, unsupervised data that relies on a combination of conventional Bayes networks model structure elicitation with optimization-based learning methods familiar in current machine learning. We have applied this to building a fault classification model for network failures that occur in the network connecting block storage units and servers that host virtual machines (VM) in a cloud data center. We have demonstrated a method to improve elicited Bayes network models by use of sparse, unlabelled data.

Our Bayes network diagnostic model is generated in part from a data-set of observed monitoring signals in combination with a bipartite network structure elicited from a mix of engineering domain knowledge and connectivity constraints derived from the network connectivity. Observed signal variables make up the lower level of the bipartite graph, the upper level being unobserved root-cause variables, each associated with a subset of signals indicated by arcs from the root-cause nodes to the signal nodes. The model parameters consist of the noisy-or reliability and "leak" entries in the signals' conditional probability tables (CPTs), and are initially set qualitatively to values that respect the sign of the causal relationships.

We have developed a novel approach to learning the signal node CPT parameters by use solely of the observed signal data. Given a bipartite network structure suggested by the domain architecture, we optimize the signal node CPT parameters to best approximate the distribution of the observed signal data. The optimization criterion is the error between the pair-wise conditional probability distribution of the observed variables and their distribution as predicted

Submitted to the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021). **To be used for reviewing only**.

by the model. We have experimented with different combinations of error functions, with more or less sparse bipartite structures, and with algorithms for different optimization methods. Once optimization has converged we evaluate the model diagnostic accuracy by comparing it against an existing rule-based model on the same data. Gratifyingly the optimized model accuracy shows substantial improvement despite the lack of supervision, i.e., labels for the root-cause nodes, in the training data.

Building bipartite diagnostic networks with unlabelled data is a type of latent-variable problem. Dependencies observed among observations suggest unobserved common causes that condition the observations. Previous researchers in this field have recognized the benefit of the additional constraints offered by assuming a Bayes network structure. Halpern and Sontag [2013] present a polynomial-time method that learns network parameters for a suitably sparse bipartite network structure. Taking this idea one step further, independencies among observable variables suggest the structural properties, such as the absence of arcs in the Bayes network. Šingliar and Hauskrecht [2006] develop a variational optimization method, assuming initially a fully connected bipartite network that recovers not only the model parameters, but also the network structure. We borrow notions from both authors, but focus on understanding how local properties of the optimization problem contribute to accuracy improvements, in a case where data-sets are not only missing labels for root-cause variables, but are sparse.

1.1 A THREE NODE EXAMPLE

The assumed sparse bipartite network partially factors into subsets of one root-cause node and the observable nodes that are its direct descendants. The simplest example consists of 3 binary-valued nodes; the root-cause C, and two observations, B and D, as shown in Figure 1. We use this network fragment to show how the optimization method applies locally. In the fragment we can observe the conditional probabilities of the observables on each other, which can be compared to the empirical values of these probabilities computed from the data. The optimization problem is to minimize an error function of these conditional probabilities in the network versus the empirical values, by adjusting the parameters in the network.

As is a common convention in diagnostic models, only one state, `detect`, is instantiated. Since there are no missing values in the data, without loss of generality, when an observation is `normal`, the node is left uninstantiated. Computationally this reduces the degrees of freedom in the problem, and implies the observables' conditional probabilities form a p by p matrix with $p(p-1)$ degrees of freedom from the observables $S_i \quad \text{for} \quad i = 1 \cdots p$. In this case the conditional probability matrix of $\mathsf{P}(S_i = \texttt{detect} \,|\, S_j = \texttt{detect})$ is just 2 by 2:

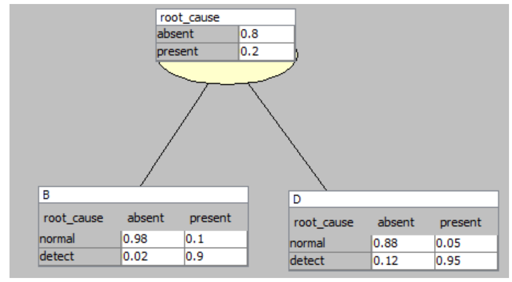

Figure 1: An Example of a Network Fragment.

$$\begin{array}{c} \\ B \\ D \end{array} \begin{array}{c} \begin{array}{cc} B & \quad D \end{array} \\ \begin{pmatrix} 1 & 0.617 \\ 0.889 & 1 \end{pmatrix} \end{array}$$

Note that this matrix, with 1s along the diagonal is not symmetric. The problem, as stated is under-determined, since the error is a function of just these 2 terms, and the network has 5 free parameters. To make the problem tractable, the root-cause prior is assumed constant, and only one parameter for each observation is varied—the "reliability" of the `detect` state—the "leak" assumed to be small and constant. These assumptions are reasonable for this domain.

In this network fragment is it possible to solve for the off-diagonal elements of the observables' conditional probability matrix as a function of the node parameters. Let $\vec{\lambda}$ be the vector $\mathsf{P}(B \,|\, D = \texttt{detect})$, and $\vec{c}$ be the root-cause prior $\mathsf{P}(C = \texttt{absent} \,|\, C = \texttt{present})$, then the π-message incident on the CPT_D matrix is just their dot product, $\vec{\pi} = \vec{\lambda} \bullet \vec{c}$, hence the posterior on D results from this matrix multiplication:

$$\mathsf{P}(D \,|\, B = \texttt{detect}) = CPT_D \, \vec{\pi}$$

.

Despite having restricted the number of parameters, this equation raises an additional concern. One can see that for such a network fragment treated in isolation, each element in the conditional probability matrix is a linear function of the adjustable network parameters, (or more precisely is monotonic once normalized). So considered individually computed posteriors do not have an inflection point, and hence may not have an internal solution in the interval (0.1). Because of this there is the undesirable possibility that the optimization routine may settle on the interval's extreme values and set the parameter on the interval boundary. The ramifications of this will be discussed in the paper's final section.

2 THE DATA CENTER LOSS OF STORAGE CONNECTIVITY (LSC) PROBLEM DOMAIN

Network diagnostics are characterized by multiple layers of abstraction, from the physical to the application layer, which generate a flood of monitoring data. We model an architecture where computing equipment, e.g. servers and disk storage, are separated in different racks in the data center. Each is networked by a“top of rack" (TOR) switch that in turn connects to higher level network switching layers. The model focuses on localizing lack-of-connectivity faults that occur either in a component of the server host, or at the TOR that connects the host to the rest of the network, or on the virtual machines (VMs) running on the host. Despite the high-rate data feeds, there is uncertainty about the layer where a fault occurs, it possibly being in the data monitoring layer itself.

As part of the system, the diagnostic model is embedded in the data center monitoring pipeline from which it receives the current signal values. When the VM senses a data I/O failure—a persistent loss of connectivity—the VM re-boots itself. This reboot event, which is intended to reset the system, triggers the model diagnostics to localize the fault, and return a list of the possible causes ranked by posterior probability of failure.

Current diagnostics are provided by a rule-based system that is straightforward, easy to understand, but requires extensive effort and time to build, also it is hard to extend to other events due to its explicit tree structure and lack of an explicit causal representation. Therefore, our data-driven Bayes network is intended to minimize labor effort and to replace a hard-to-maintain procedural rule-based diagnostic tool. The current rule-based decision tree method for the fault diagnosis is based on the experts' knowledge on the network architecture and the relative priorities of the root causes. We exploit the structure of the current rule-based decision tree from which we derive the set of root causes, and a rough idea of their relevance to different signals that suggests the connectivity of the Bayes network.

3 THE STRUCTURE OF THIS DIAGNOSTIC BAYES NETWORK

Mathematically a Bayes network is a sparse graph that factors the joint probability of a set of discrete variables. Well known, exact algorithms are available to run inference on such graphs. In this diagnostic bipartite Bayes network model we distinguish two classes of variables, the observed signals and the inferred (but unobserved) root-causes. Inference consists of computing the ranked probabilities of the root-causes given the observed state of the signals.

An advantage of Bayes networks we exploit is their ability to incorporate diverse sources of information into a consistent model. Statistical data, system architecture, engineering judgment, and incremental adjustments based on experience can be applied to learning a Bayes network. Here's how we take advantage of this. The network's variables are borrowed from the rule-based model. The branching structure of the decisions suggest which observables may relate to which root-causes. Furthermore the sequence that root-causes are addressed in the rule-based tree is based on the engineer's judgment of “priority" that we interpret as the relative prior probability of each fault. From the physical layout, the data path connectivity implies that all root-causes associated with components along the path form a noisy-or with the signal that checks connectivity. For computational convenience we unroll the noisy-or node into an equivalent chain of noisy-or nodes. Also, in many cases each root-cause has an observation paired with it. These hints, together with the observed dependencies in the signal data guides the construction of the structure of the diagnostic Bayes network.

3.1 AVAILABLE SIGNAL DATA

Our training data set consists of approximately 1500 samples of re-boot events collected over the course of a month from which eight binary feature values are derived from the raw monitoring time-series within an interval around the event. These eight signals ($Signal_1, \ldots Signal_8$) are collected from the monitoring data of the components that are possibly involved in the failures using the same thresholds applied for the rule-based system to indicate a detection. These the corresponding components (root-causes) to these signals are denoted as “Tor_rc1, Tor_rc2, NIC_rc1, NIC_rc2, NIC_rc3, NIC_rc4 and Host_rc1”, where TOR (Top Of Rank switch), NIC (Network Interface Card) and Host (Host machine) are three main sub-systems.

Typical of diagnostic domains, the fault prevalence is unbalanced, with the most common fault appearing several hundred times, and the least common occurring just in single digits. To evaluate the model accuracy after the model is learned, labels are added to the data set by running the rule-based model. Of course these labels are not seen in our model optimization step.

4 OPTIMIZATION TECHNIQUES FOR UNSUPERVISED PARAMETER LEARNING

Our premise is that the elicited Bayes network structure, when combined with the observed signal data is sufficient to learn the complete model. We formulate the problem as one of optimizing the parameters in the conditional probability tables (CPTs) for the observed nodes to be consistent with the dependencies between observed signals. For purposes of comparing the fit of the model to the data we consider

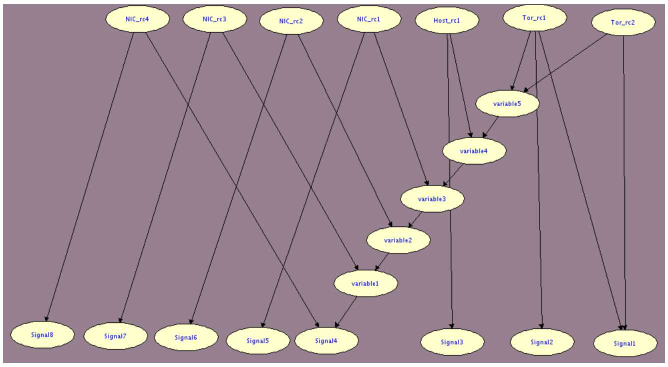

Figure 2: The Bayes network used in this example

a reduction of the joint probability of the observables by computing all pair-wise conditional probabilities between observable variables. An alternative would be to compute the full joint of the observables, but the sparseness of our elicited network structure suggests this may not be necessary. As described the pair-wise conditional probabilities form two matrici, named **D** for the data and **B** for the model.[1] The optimization error function measures the difference between them. These two matrices are:

- **D**: $p \times p$ conditional probabilities matrix, of which the ij^{th} element represents the conditional probability $P(\text{Signal}_i|\text{Signal}_j)$ calculated from the empirical data directly. It describes the ground truth of the signals' dependence in the data.
- **B**: $p \times p$ conditional probabilities matrix , of which the ij^{th} element represents the conditional probability $P(\text{Signal}_i|\text{Signal}_j)$ propagated by the estimated Bayesian Network. It describes the signals' dependence derived by the Bayesian Network with estimated parameters.

By minimizing this difference between these matrici, the solution could be regarded as a kind of semi-supervised learning, without the labeled root-cause data.

The options of the loss function to capture the difference between two matrices include:

- sum of square error, $\sum_{i \neq j} (b_{ij} - d_{ij})^2$
- chi-squared statistics
- K-L divergence

The options of the optimization algorithms include:

- Bisection line-search
- Grid-search
- Gradient free optimization: Nelder–Mead method

[1]The matrici D and B are not to be confused with the nodes D and B from the example.

Optimization by Line-search and Grid-search optimized the parameters one at a time, sequencing through the signal nodes, shuffling the order in different runs. For Nelder-Mead, the optimizer was given the full set of network parameters with their constraints; these runs took noticeably longer. Some runs terminated either when the optimization function converged, or in cases of early-stopping, when a set number of iterations completed. In most cases we saw a measurable improvement in the test accuracy after parameter optimization, but in some cases, the solution "collapsed", dedicating all cases to just a few of the most prevalent faults. We found similar results on more or less sparse versions of the model, and with greater or fewer root-cause nodes. In response we resorted to chi-squared and KL divergence error functions that should be more sensitive to small valued probabilities, and to global optimization, which should not get stuck in local optima found by individual line search, but neither of those alternative improved results, in general.

5 INFERENCE RESULTS

Over the combinations of optimization criteria and algorithms, after the optimization step the best results for the current model was found using Sum-of-squared error and Bisection line-search resulting in a decrease in the error between D matrix and B matrix from 0.084 to 0.047. One can see the details of the improvement in the error by side by side comparison of the observation pair-wise conditional probabilities computed from the data, with that of the Bayes network after parameter optimization.

The heatmap of the **D** matrix is shown in Fig 3, and that of **B** matrix in the heatmap of Fig 4. One can see that after the optimization, the magnitude of probabilities in **B** is close to the magnitude of probabilities in **D**. For example, the conditional probabilities of "HighCpu" on other signals are all close to 0 in both heatmaps, while the conditional probabilities of "Host2TorPingmeshDrops" on other signals are all relatively larger in both heatmaps. However, as constrained by the presumed network structure the final error after the optimization step is still appreciable. On closer examination, contrary to what one might expect, the tendency for rows in the **B** to take on uniform values implies that the algorithm has simplified the model by effectively zeroing-out some of the model's arcs.

5.1 INFERENCE ACCURACY – COMPARISON WITH RULE-BASED RESULTS

The improvement in accuracy gained by optimization that lead to more accurate inference by the Bayes network, as de-

	Tor_rc1	Tor_rc2	NIC_rc1	NIC_rc2	NIC_rc3	NIC_rc4	Host_rc1
Tor_rc1	673	23	0	0	0	0	1
Tor_rc2	0	136	0	0	0	0	25
NIC_rc1	0	14	38	0	0	20	26
NIC_rc2	0	0	0	0	0	2	26
NIC_rc3	0	3	0	0	4	0	7
NIC_rc4	1	1	0	0	0	3	0
Host_rc1	0	0	0	0	0	0	132

Table 1: Confusion matrix of the optimized Bayesian Network

Tor_rc1	precision: 0.999	recall: 0.966	F1: 0.982
Tor_rc2	precision: 0.768	recall: 0.845	F1: 0.805
Host_rc1	precision: 0.608	recall: 1.0	F1: 0.756
NIC_rc1	precision: 1.0	recall: 0.388	F1: 0.559
NIC_rc2	precision: 0.12	recall: 0.6	F1: 0.2
NIC_rc3	precision: 0.0	recall: 0.0	F1: 0.0
NIC_rc4	precision: 1.0	recall: 0.286	F1: 0.444

Table 2: Precision, Recall and F1 score of each category

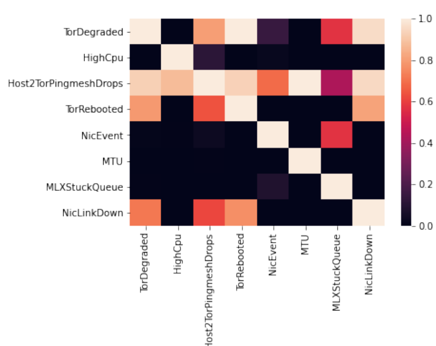

Figure 3: D matrix: empirical conditional probability matrix driven from the data

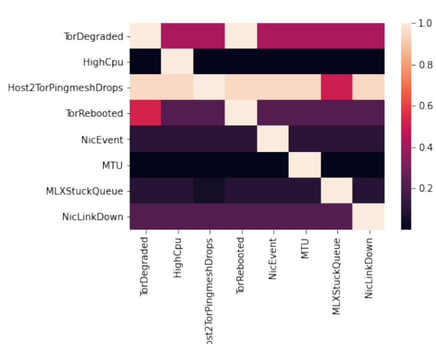

Figure 4: B matrix: estimated conditional probability inferred from the estimated Bayesian Network

termined by comparison with the rules-based model. **Correspondingly accuracy of the optimized model measured against the rule-based decision tree derived labels is** 86.87%. **compared to a pre-optimized model of roughly** 50%**.** The confusion matrix in Table 1 shows the results for all root-causes. The root-causes with higher prevalence have higher accuracy such as Tor_rc1, Tor_rc2 and Host_rc1. Accordingly, the rarer root causes are more likely to be classified incorrectly.

Besides the overall accuracy, the results of other measurement metrics (Precision, Recall and F1) in Table 2 also agree on our findings. All metrics are substantially higher for high-prevalence categories (Tor_rc1, Tor_rc2 and Host_rc1).

6 FUTURE WORK

By constructing an *a priori* Bayesian network structure with domain knowledge, and then learning the parameters by optimizing the loss function over observed data, we've demonstrated a promising approach that can correctly identify most of the root causes in the real scenario when compared with tediously generated rules-based model. We've tested the presumption that the achieved accuracy in light of sparse data is due to the local independence properties of the diagnostic Bayes network making it possible to factor the optimization among faults.

However, this method is still in early stages and has many limitations. During the optimization step, the optimal value of conditional probability table (CPT) parameters typically take on extreme values in the probability interval. Since as noted, for any one parameter, the value of the corresponding observation conditional probability is monotonic, there is a tendency for the optimization algorithm to get stuck at extremes of the intervals. This lack of robustness in the optimization problem formulation deserves further investigation. Another plausible explanation is that the data used for the rules-based model presumes a deterministic prediction (e.g. it predicts one outcome, effectively with probability one), and we just might be seeing the probabilistic model mimic this behavior. Additionally, since our model is largely singly connected, it may not be rich enough to find internal parameter values produced by parameter interactions. However experiments to date with multiply connected models have not shown improvement.

To improve the work there are challenges we are working on. First our current optimization methods tend to collapse infrequently observed signal combinations into more common cases. This may have a small effect on overall accuracy, but reduces the ability to distinguish rare faults. We are investigating better local optimization methods—selecting subsets of parameters to optimize at one time, based on network structure. Secondly we should be able to infer network sparsity based on zero entries in the **D** matrix. An initial approach, to derive independencies from the structure of a Bayes network learned just among the signals implied a copious number of arcs. In contrast, current experiments have found best results with very sparse structures, bordering on singly connected. The proper degree of sparsity suggested by the data remains an open question. Despite the limitations of the current method, the results show promise of practical methods for combined judgment-based and empirically data-driven Bayes network modeling.

References

Yonatan Halpern and David Sontag. Unsupervised learning of noisy-or bayesian networks. *Appears in Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*, 2013, August 2013. Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013. AUAI Press 2013.

T. Šingliar and Milos Hauskrecht. Noisy-or component analysis and its application to link analysis. *Journal of Machine Learning Research*, 7:2189–2213, Oct 2006.