Unification of Leaky Noisy OR and Logistic Regression Models and Maximum *A Posteriori* Inference for Multiple Fault Diagnosis Using the Unified Model

Ali Abdollahi and **Krishna R. Pattipati**
University of Connecticut, Storrs, CT, USA
e-mail: {abdollahi, krishna}@engr.uconn.edu

Abstract

In this paper, we first discuss two widely-used models for fault diagnosis, namely, Detection-False Alarm (DFA) and Leaky Noisy OR (LNOR) models, and we prove that they are equivalent. Then, we discuss logistic test models from a broad-to-narrow perspective, first by considering a comprehensive, but combinatorial model and then simplifying it to obtain more tractable models. Two realization of logistic regression (LR) model are presented and their similarity to DFA and LNOR models are illustrated. Next, we present a unified model that subsumes the LNOR and the LR models. Finally, we present the *maximum a posteriori* (MAP) inference problem for the unified model and we derive its solution using Lagrangian relaxation method, and also its corresponding dual cost function. Comparisons of the LNOR and LR models are presented using an example. The results show that the LR model is more robust with respect to the structure of the training data.

1 Introduction

Fault Detection and Diagnosis (FDD) methods have mainly evolved upon three major paradigms, viz., model-based, data-driven and knowledge-based approaches. Model-based approaches for FDD require a mathematical representation of the system. Such methods are effective when the physics-based models of the system are available. However, these approaches are restricted to systems with relatively small number of inputs and outputs. Hence, for large-scale systems for which detailed analytical models of failure are not available, the analytic model-based methods are impractical. Traditional approaches for fault detection, as discussed in [Willsky, 1976], include "failure-sensitive" filters [Kerr, 1974], [Jones, 1973], voting systems (for systems with a high degree of redundancy in parallel hardware), multiple hypothesis filter-detectors [Lainiotis, 1971], [Athans *et al.*, 1975], [Clark *et al.*, 1975], jump process techniques [Boel *et al.*, 1975b], [Boel *et al.*, 1975a], and innovation-based detection systems [Mehra and Peschon, 1971], [Merrill, 1972], [Schweppe and Handschin, 1974], [Handschin *et al.*, 1975], [Willsky and Jones, 1976].

Data-driven approaches to FDD are used when the system models are unavailable, but adequate data monitoring is available. Such systems are frequently used when vendors of subsystems do not provide the details of the internal functioning of their products in order to protect their intellectual property. To circumvent this lack of product details, data-driven approaches utilize substantial monitoring data in order to train a model that satisfactorily simulates the black-box system. Neural networks [Frank and Köppen-Seliger, 1997], [Bishop, 1995] and machine learning methods [Theodoridis, 2015], [Rijmen, 2008] are among these data-driven techniques.

Knowledge-based approaches to FDD require qualitative models for process monitoring and are used when mathematical models are unavailable. Most knowledge-based techniques are based on causal analysis, expert systems, and/or ad hoc rules. Because of the qualitative nature of these models, knowledge-based approaches have been applied to many complex systems. Graphical models, such as Petri nets, multi-signal flow graphs and Bayesian networks [Theodoridis, 2015], are applied for diagnostic knowledge representation and inference in automotive systems. Bayesian Networks subsume the deterministic fault diagnosis models embodied in the Petri net and multi-signal models.

The model based, data-driven and knowledge-based approaches provide the "sand box" that test designers can use to experiment with, and systematically select relevant models or combinations thereof to satisfy the requirements on diagnostic accuracy, computational speed, memory, on-line versus off-line diagnosis, and so on. Ironically, no single technique alone can serve as the diagnostic approach for complex systems. Thus, an integrated diagnostic process that naturally employs data-driven techniques, graph-based dependency models and mathematical/physical models is necessary for fault diagnosis, thereby enabling efficient maintenance of these systems. The probabilistic graphical models provide such an integrating platform.

The probabilistic graphical model of a fault diagnosis system, as shown in Fig. 1, can be visualized as a tripartite digraph (directed graph) [Abdollahi *et al.*, 2016], [Shakeri *et al.*, 1998]. The first layer of the tripartite digraph consists of m failure sources. Failure sources can be *binary*, *multi-state*, or *continuous*. In a binary representation, a failure source x_i has either of the two values 0 and 1, respectively, corresponding to the *nonexistence* of failure (i.e., *normal state*) and the *existence* of failure, and to each is assigned a prior probability. The second layer consists of n tests, which, similar to failure sources, can be *binary*, *multi-state*, or *continuous*. In a binary description, a test t_j is either 0

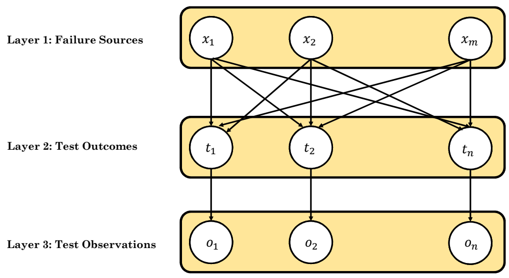

Figure 1: Fault diagnosis representation in tripartite digraph

or 1, respectively, representing the *pass* or *fail* outcome of the test. Failure sources affect the tests in a probabilistic way. The third layer of the tripartite graph comprises the observations of the test outcomes. The test observation o_j represents the measured outcome of test t_j. The difference between t_j and o_j may stem from errors in communication links, sensors, or sensor signal processing. If we assume that the test observations are perfect, that is, $o_j = t_j$ for $j = 1, \ldots, n$, then the tripartite digraph devolves to a bipartite digraph. In this paper, we assume that test observations are perfect and any uncertainty in the relationship between the states of failure sources and test outcomes is represented in the test models.

In this paper, we discuss two widely-used models for fault diagnosis, namely, Detection-False Alarm (DFA) and leaky noisy OR (LNOR) models, and we prove that they are equivalent. Two realizations of logistic regression (LR) are also discussed and their similarities with DFA and LNOR models are discussed. Then, we propose a unified model for LNOR and LR models, using which we define the fault diagnosis problem as one of MAP inference. The MAP problem is solved using the Lagrangian relaxation method and a dual cost function is derived. Comparisons of the LNOR and LR models are presented using a simple example.

The contribution of this paper are as follows:

- Proving the equivalence of two widely-used graphical models in fault diagnosis, namely, the DFA and the LNOR models.
- Presenting a unified model to for the LNOR and the LR models.
- Deriving the solution of the *maximum a posterior* (MAP) inference problem for the unified model, using Lagrangian relaxation method and deriving the corresponding dual cost function.
- Comparison of the LNOR and the LR models, which shows that the LR model is more robust with respect to the structure of the training data.

The structure of the paper is as follows. In section 2, we discuss two test models based on probabilistic graphical models, viz., Detection-False Alarm (DFA) and leaky noisy OR (LNOR), that are used for fault diagnosis, and we prove that for each DFA model, there exists a unique LNOR model. Then, in section 3, we discuss logistic representation of test models for fault diagnosis and present the logistic combinatorial, restricted logistic, and logistic regression (LR) models. We show that restricted and regression versions of logistic models are equivalent in a way that is reminiscent of DFA and LNOR models. In section 4, we present a unified model for the LNOR and LR models. In section 5, we present the *maximum a posteriori* (MAP) inference problem for the unified model, and derive its solution using the Lagrangian relaxation method, and then we derive the dual cost function for the MAP problem. In section 6, we present the simulation results, where we compare the performance of the LNOR and LR models and also discuss the dual cost function of the MAP inference problem for the unified model. Finally, in section 7, we conclude the paper.

2 Equivalence of Detection-False Alarm (DFA) and Leaky Noisy OR (LNOR) Models

In this section, we show the equivalence of two widely-used graphical models in fault diagnosis. In the binary case, the probabilistic relation between a failure source and a test outcome can be represented in terms of detection and false alarm probabilities [Shakeri *et al.*, 1998], [Abdollahi *et al.*, 2016]. The detection probability (Pd_{ij}) is the probability that test t_j has a *fail* outcome, given that the failure source x_i is in a *failure* state; mathematically,

$$Pd_{ij} = \Pr\{t_j = 1 | x_i = 1\}. \tag{1}$$

False alarm probability (Pf_{ij}) is the probability that test t_j has a *fail* outcome, given that the failure source x_i is in a *non-failure* state; mathematically,

$$Pf_{ij} = \Pr\{t_j = 1 | x_i = 0\}. \tag{2}$$

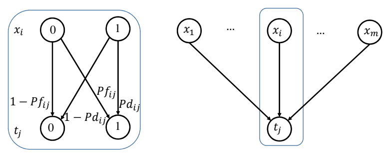

Figure 2: Detection-False Alarm (DFA) Model

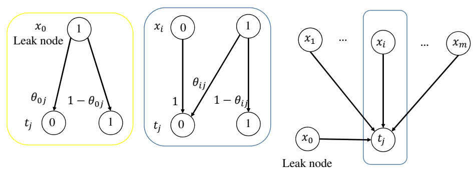

Figure 3: Leaky noisy OR

Figure 2 shows the Detection-False Alarm (DFA) model for a system with m failure sources and one test t_j. In this model, in order for the test t_j to *pass*, all failure sources should be in *no-failure* state. Based on conditional independence assumption of test outcomes given the states of failure sources, we can write:

$$\Pr\{t_j = 0 | \mathbf{x}\} = \prod_{i=1}^{m} (1 - Pf_{ij})^{1-x_i} (1 - Pd_{ij})^{x_i}. \tag{3}$$

Figure 3 shows the LNOR model. In this model, for failure sources x_1 to x_m a value of zero leads to test outcome t_j of *pass*; that is,

$$\Pr\{t_j = 0 | x_i = 0\} = 1. \tag{4}$$

However, when $x_i = 1$, the test outcome would be zero with probability θ_{ij} so that

$$\Pr\{t_j = 0|x_i = 1\} = \theta_{ij}. \tag{5}$$

Thus far, it is called noisy OR. In fact, if $\theta_{ij} = 0$ for $i = 1, 2, ..., m$, then it is a logical OR for which the output (t_j) is 1, if at least one of the inputs (x_i) is 1. However, when $\theta_{ij} \neq 0$, the output (t_j) could be 0 even when one or more inputs (x_i) are 1. Hence, it is called a noisy OR model. For this model, however, if all inputs are equal to 0, the output would be 0. To inject the possibility of the output being 1 even when all the inputs are 0, a leaky node is added (shown as x_0), whose value is always 1. The noisy OR model with a leaky node is called a leaky noisy OR (LNOR) model and the leaky node in fact represents the unmodeled dynamics in the system [Murphy, 2012]. For a LNOR model

$$\Pr\{t_j = 0|\mathbf{x}\} = \theta_{0j} \prod_{i=1}^{m} \theta_{ij}^{x_i}. \tag{6}$$

Proposition. *For a DFA model with m failure sources and n tests, and with detection probabilities Pd_{ij} and false alarm probabilities Pf_{ij}, there exist a unique LNOR model with the following parameters:*

$$\theta_{0j} = \prod_{i=1}^{m} (1 - Pf_{ij}), \tag{7}$$

$$\theta_{ij} = \frac{1 - Pd_{ij}}{1 - Pf_{ij}}, \quad i = 1, 2, \ldots, m. \tag{8}$$

Proof. The following relation proves the existence of a unique LNOR model for each DFA model.

$$\begin{aligned} \Pr\{t_j = 0|\mathbf{x}\} &= \prod_{i=1}^{m} (1 - Pf_{ij})^{1-x_i} (1 - Pd_{ij})^{x_i} \\ &= \prod_{i=1}^{m} (1 - Pf_{ij}) \left(\tfrac{1-Pd_{ij}}{1-Pf_{ij}}\right)^{x_i} \\ &= \left(\prod_{i=1}^{m} (1 - Pf_{ij})\right) \prod_{i=1}^{m} \left(\tfrac{1-Pd_{ij}}{1-Pf_{ij}}\right)^{x_i} = \theta_{0j} \prod_{i=1}^{m} \theta_{ij}^{x_i}. \end{aligned} \tag{9}$$

□

Remark. *The mapping from the LNOR parameters to the DFA parameters is not unique.*

Note that false alarms occur for two reasons: errors in sensor measurements and processing, or unmodeled parameters. The DFA model views the occurrence of false alarms from a sensor measurement and processing error perspective, while the LNOR model views them from an unmodeled dynamics perspective. However, in both models, once the parameters are learned from the fault injection-test output observations, both sensor measurement processing error and unmodeled parameter effects are reflected in the parameters of the respective models.

3 Logistic Representation of Test Models

The problem of fault diagnosis in the binary case can be formulated in the following form using the logistic function.

$$\Pr(t_j = 0|\mathbf{x}) = \frac{\exp(f_j(\mathbf{x}))}{1 + \exp(f_j(\mathbf{x}))}, \tag{10}$$

Since the sum of probabilities of all possible outcomes must add up to 1, it is required that:

$$\Pr(t_j = 1|\mathbf{x}) = \frac{1}{1 + \exp(f_j(\mathbf{x}))}, \tag{11}$$

By dividing equations (10) and (11), and taking the logarithm, we obtain:

$$f_j(\mathbf{x}) = \ln\left(\frac{\Pr(t_j = 0|\mathbf{x})}{\Pr(t_j = 1|\mathbf{x})}\right) \tag{12}$$

3.1 Logistic Combinatorial Test Models

The simplest way of modeling $f_j(\mathbf{x})$ is to assign a weight ω_j for each configuration of failure states as[1]:

$$\Pr(t_j = 0|\mathbf{x}) = \frac{\exp(\omega_j(x_m, \ldots, x_1))}{1 + \exp(\omega_j(x_m, \ldots, x_1))}, \tag{13}$$

That is

$$f_j(\mathbf{x}) = \omega_j(x_m, \ldots, x_1) \tag{14}$$

Another way of constructing a logistic combinatorial model is to use a polynomial of degree m for $f_j(\mathbf{x})$, as follows:

$$f_j(\mathbf{x}) = \sum_{k_m=0}^{1} \cdots \sum_{k_1=0}^{1} \nu_{k_m \cdots k_1 j} x_m^{k_m} \cdots x_1^{k_1} \tag{15}$$

3.2 Restricted Logistic Model

A method to reduce the number of parameters is to assign a weighting to each failure source and *restricting* its dependency on the state of that failure source; that is,

$$\Pr(t_j = 0|\mathbf{x}) = \frac{\exp\left(\sum_{i=1}^{m} \omega_{ij}(x_i)\right)}{1 + \exp\left(\sum_{i=1}^{m} \omega_{ij}(x_i)\right)} \tag{16}$$

As an example, consider two failure sources x_1, x_2 and one test t_j. The state configuration and the corresponding $f_j(\mathbf{x})$ are shown in Table 2.

Table 1: Input and $f_j(\mathbf{x})$ for two failure sources and a single output

x_2	x_1	$f_j(x_2, x_1)$
0	0	$\omega_{2j}(0) + \omega_{1j}(0)$
0	1	$\omega_{2j}(0) + \omega_{1j}(1)$
1	0	$\omega_{2j}(1) + \omega_{1j}(0)$
1	1	$\omega_{2j}(1) + \omega_{1j}(1)$

Note that the logistic model of (16) is analogous to the DFA model in the sense that in both models a zero state of any of the failure sources affects the probability of t_j, via Pf_{ij} in the DFA model and via $\omega_{ij}(0)$ in the logistic model.

[1]Indexing in reverse order is intentional for decimal representation of the configuration states later.

3.3 Logistic Regression Model

Similar to the derivation of LNOR from DFA we can derive the LR model from (16). We showed that the LNOR can aggregate the effect of the zero-states of the failure sources by introducing a leaky node and that the LNOR is equivalent to the DFA model. This suggests us to consider the effect of $\omega_{ij}(0)$ for all $i = 1, 2, \dots, m$ into one parameter. To illustrate this point, we consider a system with two failure sources x_1 and x_2 and one test t_j, all binary. For the new logistic model we neglect the effect of each failure source when its state is zero. Therefore, corresponding to $\omega_{1j}(0)$ and $\omega_{2j}(0)$, we do not have any parameter in the new model, but we assign a parameter ξ_{0j} that is applied in any combination of the failure states. Therefore, we have the following relations between the ω parameters of model (16) and the ξ parameters of the new logistic model.

$$\begin{array}{l} \omega_{2j}(0) + \omega_{1j}(0) = \xi_{0j} \\ \omega_{2j}(0) + \omega_{1j}(1) = \xi_{0j} + \xi_{1j} \\ \omega_{2j}(1) + \omega_{1j}(0) = \xi_{0j} + \xi_{2j} \\ \omega_{2j}(1) + \omega_{1j}(1) = \xi_{0j} + \xi_{2j} + \xi_{1j} \end{array} \tag{17}$$

Solving the above equations for the ξ parameters, yields:

$$\begin{array}{l} \xi_{0j} = \omega_{2j}(0) + \omega_{1j}(0) \\ \xi_{1j} = \omega_{1j}(1) - \omega_{1j}(0) \\ \xi_{2j} = \omega_{2j}(1) - \omega_{2j}(0) \end{array} \tag{18}$$

Remark. *In general, for m failure sources $x_1, x_2, \dots, x_m$ and one test t_j, all binary, we can write:*

$$\begin{array}{l} \xi_{0j} = \sum\limits_{i=1}^{m} \omega_{ij}(0) \\ \xi_{ij} = \omega_{ij}(1) - \omega_{ij}(0), \quad i = 1, \dots, m \end{array} \tag{19}$$

The general formulation for would be:

$$\Pr(t_j = 0|\mathbf{x}) = \frac{\exp\left(\xi_{0j} + \sum\limits_{i=1}^{m} \xi_{ij} x_i\right)}{1 + \exp\left(\xi_{0j} + \sum\limits_{i=1}^{m} \xi_{ij} x_i\right)} \tag{20}$$

Remark. *Note that ξ_{0j} is the sum of all $\omega_{ij}(0)$'s, analogous to θ_{0j} being the multiplication of all $(1 - Pf_{ij})$'s. This is because $\omega_{ij}(0)$ is related to the logarithm of the probability (see* (12)*). Also, note the similarity to a logistic neuron [Theodoridis, 2015].*

Remark. *Note that ξ_{ij} is the difference between $\omega_{ij}(1)$ and $\omega_{ij}(0)$, analogous to θ_{ij} being the ratio of $(1 - Pd_{ij})$ and $(1 - Pf_{ij})$.*

4 Unified Representation of Leaky Noisy OR and Logistic Regression Models

In this section we present a unified representation for the LNOR and LR models. Both models can be represented as follows:

$$z_j(\mathbf{x}) = \theta_{0j} \prod_{i=1}^{m} \theta_{ij}^{x_i}. \tag{21}$$

with

$$z_j(\mathbf{x}) = \Pr(t_j = 0|\mathbf{x}) \Big(1 - \Pr(t_j = 0|\mathbf{x})\Big)^{-d} \tag{22}$$

where,

$$d = \begin{cases} 0 & \text{Leaky noisy OR} \\ 1 & \text{Logistic regression} \end{cases} \tag{23}$$

Note that in the LNOR model, $z_j(\mathbf{x})$ is the probability of test t_j being zero given $\mathbf{x}$, that is, $z_j(\mathbf{x}) = \Pr(t_j = 0|\mathbf{x})$, and in the LR model, $z_j(\mathbf{x})$ is the odds of test t_j being zero given $\mathbf{x}$, that is $z_j(\mathbf{x}) = \frac{\Pr(t_j=0|\mathbf{x})}{1-\Pr(t_j=0|\mathbf{x})}$. The parameters $\theta_{ij}, i = 0, 1 \dots, m$ for the LNOR are as they were defined in section 2, and for the LR, they are as follows:

$$\theta_{0j} = \exp(\xi_{0j}) \tag{24}$$

$$\theta_{ij} = \exp(\xi_{ij}), \quad i = 1, \dots, m \tag{25}$$

The unified representation can be written as follows, as well:

$$\ln(z_j(\mathbf{x})) = \beta_{0j} + \sum_{i=1}^{m} \beta_{ij} x_i \tag{26}$$

where for the LNOR, $\beta_{ij} = \ln(\theta_{ij})$, and for the LR model, $\beta_{0j} = \xi_{0j}$, and $\beta_{ij} = \xi_{ij}$ for $i = 1, \dots, m.$

Remark. *Note that* (26) *states that for the LNOR, the logarithm of probability of test t_j being zero is an affine linear function of the failure source states (linearity in probability) while, for the LR the logarithm of the odds of t_j being zero is an affine linear function of the failure source states (linearity in odds).*

Remark. *For the same observed data, the parameters of β_{ij}, $i = 0, 1 \dots, m$ for the LNOR model are different from those of the LR model.*

Remark. *For the LNOR, the parameters β_{ij} for $i = 0, 1, \dots, m$ are non-positive, because θ_{ij} for $i = 0, 1, \dots, m$, being probabilities, are restricted to be in the interval* $[0, 1]$.

5 Maximum A Posteriori (MAP) Inference for the Unified Model

In this section, we present the fault diagnosis problem as the *maximum a posterior* (MAP) inference problem for the unified model, and derive its solution using the Lagrangian relaxation method. Then, we derive the dual cost function for this MAP inference problem.

Let $\mathbf{x}$ denote the vector of all failure sources, and T be the test outcomes. Given T, the inference problem can be expressed as follows:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \Pr(\mathbf{x}|T). \tag{27}$$

The MAP estimation for the above inference problem can be written as follows:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \ln(\Pr(T|\mathbf{x}) \Pr(\mathbf{x})). \tag{28}$$

By categorizing the observed test outcomes T into two disjoint sets of passed test outcomes T_p, and failed test outcomes T_f, we can write the MAP problem (28), after taking logarithm, as follows:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \{\ln(\Pr(T_f|\mathbf{x})) + \ln(\Pr(T_p|\mathbf{x})) + \ln(\Pr(\mathbf{x}))\} \tag{29}$$

Assuming each failure source x_i has the prior probability p_{s_i} of being in the fail state, we can write:

$$\Pr(x_i) = (1 - p_{s_i})\left(\frac{p_{s_i}}{1 - p_{s_i}}\right)^{x_i} \tag{30}$$

Validity of (30) can be verified by checking $\Pr(x_i)$ for the two possible values $x_i = 0$ and $x_i = 1$. Using the conditional independence of failure sources, we can write:

$$\Pr(\mathbf{x}) = \prod_{i=1}^{m} \Pr(x_i) \tag{31}$$

Taking logarithm of (31), and using (30), we can write:

$$\ln(\Pr(\mathbf{x})) = \gamma_0 + \sum_{i=1}^{m} \gamma_i x_i \tag{32}$$

where,

$$\gamma_0 = \sum_{i=1}^{m} \ln(1 - p_{s_i}) \tag{33}$$

$$\gamma_i = \ln\left(\frac{p_{s_i}}{1 - p_{s_i}}\right) \tag{34}$$

To calculate $\ln(\Pr(T_p|\mathbf{x}))$, we start by equation (26) and (22), rewritten below for ready reference:

$$\ln(z_j(\mathbf{x})) = \beta_{0j} + \sum_{i=1}^{m} \beta_{ij} x_i \tag{35}$$

$$z_j(\mathbf{x}) = \Pr(t_j = 0|\mathbf{x})\Big(1 - \Pr(t_j = 0|\mathbf{x})\Big)^{-d} \tag{36}$$

Inserting (36) into (35) yields:

$$\begin{aligned} \ln(\Pr(t_j = 0|\mathbf{x})) = &\Big(\beta_{0j} + \sum_{i=1}^{m} \beta_{ij} x_i\Big) \\ &+ d\ln\Big(1 - \Pr(t_j = 0|\mathbf{x})\Big) \end{aligned} \tag{37}$$

Since test outcomes are independent, we can write

$$\Pr(T_p|\mathbf{x}) = \prod_{t_j \in T_p} \Pr(t_j = 0|\mathbf{x}) \tag{38}$$

Taking logarithm of (38), and using (37), we have:

$$\ln(\Pr(T_p|\mathbf{x})) = \beta_0 + \sum_{i=1}^{m} \beta_i x_i + d \sum_{t_j \in T_p} \ln(1 - y_j) \tag{39}$$

where,

$$\beta_i = \sum_{t_j \in T_p} \beta_{ij}, \quad i = 0, 1, \ldots, m \tag{40}$$

$$y_j = \Pr(t_j = 0|\mathbf{x}) \tag{41}$$

To calculate $\ln(\Pr(T_f|\mathbf{x}))$, based on independence of test outcomes, we first note that:

$$\Pr(T_f|\mathbf{x}) = \prod_{t_j \in T_f} \Pr(t_j = 1|\mathbf{x}) \tag{42}$$

Taking logarithm of (42), and noting that probability of a fail outcome is the complement of probability of the pass outcome, we can write:

$$\ln(\Pr(T_f|\mathbf{x})) = \sum_{t_j \in T_f} \ln(1 - y_j) \tag{43}$$

Using equations (32), (39), and (43), and by discarding constants γ_0 and β_0, we can write the MAP inference problem (29) as follows:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \sum_{t_j \in T_z} \ln(1 - y_j) + \sum_{i=1}^{m} \alpha_i x_i \tag{44}$$

subject to:

$$\ln(y_j) - d\ln(1 - y_j) = \beta_{0j} + \sum_{i=1}^{m} \beta_{ij} x_i,\ \forall t_j \in T_z \tag{45}$$

where,

$$T_z = \begin{cases} T_f & \text{Leaky noisy OR} \\ T & \text{Logistic regression} \end{cases} \tag{46}$$

$$\alpha_i = \gamma_i + \beta_i, \quad i = 1, \ldots, m \tag{47}$$

Note that, for the LR model, we have used the relation $T_p \cup T_f = T$.

The problem in (44), in the ideal case ($\theta_{0j} = 1, \theta_{ij} = 0, i = 1, \ldots, m$) for the LNOR model, reduces to a set covering problem. Since the set covering problem is NP-hard [Miller, 2013], the general problem in (44) is NP-hard as well.

Note that, while in the second summation in (44), the effect of the failure sources are decoupled, the first summation through y_j couples *all* the failure source states nonlinearly. To relax this coupling, we use Lagrange relaxation method. The relaxed version of optimization problem becomes:

$$\begin{aligned} L = &\sum_{t_j \in T_z} \ln(1 - y_j) + \sum_{i=1}^{m} \alpha_i x_i \\ &+ \sum_{t_j \in T_z} \lambda_j \Big(\ln(y_j) - d\ln(1 - y_j) - \beta_{0j} - \sum_{i=1}^{m} \beta_{ij} x_i\Big) \end{aligned} \tag{48}$$

By taking the derivative of (48) with respect to y_j and equating it to zero, we get:

$$y_j = \frac{\lambda_j}{1 + (1 - d)\lambda_j}, \quad \forall t_j \in T_z \tag{49}$$

Inserting (49) into (48) and rearranging it, we have:

$$L(\boldsymbol{\lambda}, \mathbf{x}) = \sum_{t_j \in T_z} L_{1j}(\lambda_j) + \sum_{i=1}^{m} c_i(\boldsymbol{\lambda}) x_i \tag{50}$$

where

$$c_i(\boldsymbol{\lambda}) = \alpha_i - \sum_{t_j \in T_z} \lambda_j \beta_{ij} \tag{51}$$

$$\begin{aligned} L_{1j}(\lambda_j) = &\lambda_j \ln(\lambda_j) + (1 - d\lambda_j)\ln(1 - d\lambda_j) \\ &- (1 + (1 - d)\lambda_j)\ln(1 + (1 - d)\lambda_j) - \beta_{0j}\lambda_j \end{aligned} \tag{52}$$

Note that given $\boldsymbol{\lambda}$, the first summation in (50) is a constant and can be discarded, and the optimization in the second summation can be performed *for each failure source separately*. Therefore,

$$x_i^*(\boldsymbol{\lambda}) = u\left(c_i(\boldsymbol{\lambda})\right) = u\left(\alpha_i - \sum_{t_j \in T_z} \lambda_j \beta_{ij}\right) \tag{53}$$

where, $u(.)$ is the unit step function. The pure dual cost function then is as follows:

$$L(\boldsymbol{\lambda}) = \sum_{t_j \in T_z} L_{1j}(\lambda_j) + \sum_{i=1}^{m} L_{2i}(\boldsymbol{\lambda}) \tag{54}$$

where,

$$L_{2i}(\boldsymbol{\lambda}) = c_i(\boldsymbol{\lambda}) u\left(c_i(\boldsymbol{\lambda})\right) = \max\left(0, \;\; c_i(\boldsymbol{\lambda})\right) \tag{55}$$

and $L_{1j}(\lambda_j)$ is as in (52), which can also be written as in (56) and (57), respectively, for the LNOR and the LR models.

$$L_{1j}(\lambda_j) = \lambda_j \ln(\lambda_j) - (1+\lambda_j)\ln(1+\lambda_j) - \beta_{0j}\lambda_j \quad \forall t_j \in T_f \tag{56}$$

$$\begin{aligned} L_{1j}(\lambda_j) &= \lambda_j \ln(\lambda_j) + (1-\lambda_j)\ln(1-\lambda_j) - \beta_{0j}\lambda_j \\ &= -H(\lambda_j) - \beta_{0j}\lambda_j, \quad \forall t_j \in T \end{aligned} \tag{57}$$

where, $H(\lambda_j)$ is the binary entropy. Thus, one needs Lagrange multipliers for *failed tests only* in the case of LNOR, while the Lagrange multipliers are need for *all tests* in the case of LR.

6 Simulation Results

6.1 Comparison of the Leaky Noisy OR and Logistic Regression Models

In this section, using an example we compare the LNOR and LR models. Consider a system with three failure sources x_1, x_2, and x_3 and two tests t_1 and t_2. Assume that test t_1 is affected by failures source x_1 and x_2, and test t_2 is affected by failure sources x_1 and x_3. From a DFA modeling perspective, the non-zero detection and false alarm probabilities are $Pd_{11} = 0.85$, $Pd_{21} = 0.90$, $Pd_{22} = 0.80$, $Pd_{32} = 0.95$, $Pf_{11} = 0.06$, $Pf_{21} = 0.03$, $Pf_{22} = 0.07$, $Pf_{32} = 0.08$. Using (7) and (8), we can calculate the parameters of its equivalent LNOR model. Then we can find the least squares estimate of LR model. Let $\Pr(t_j = 0|\mathbf{x})$for these models be, respectively, denoted as $\mathrm{P}_{\mathrm{OrgLNOR}}(\mathbf{x})$ and $\mathrm{P}_{\mathrm{OrgLR}}(\mathbf{x})$, where the subscripts "OrgLNOR" and "OrgLR" represent the best estimates for the *original* DFA model, respectively for the LNOR and the LR models. A wighted sum of the test probabilities of these two models is used as the training data.

$$\begin{aligned} &\Pr_{\mathrm{Obs}}(t_j = 0|\mathbf{x}) = \\ &w_{\mathrm{LNOR}} \mathrm{P}^j_{\mathrm{OrgLNOR}}(\mathbf{x}) + (1 - w_{\mathrm{LNOR}}) \mathrm{P}^j_{\mathrm{OrgLR}}(\mathbf{x}) \end{aligned} \tag{58}$$

where $0 \leq w_{\mathrm{LNOR}} \leq 1$ and $j = 1, 2$. When $w_{\mathrm{LNOR}} = 1$, the observations have a linear structure in the logarithm of probability of test pass, and when $w_{\mathrm{LNOR}} = 0$ they have a linear structure in the logarithm of the odds of test pass, and when $0 < w_{\mathrm{LNOR}} < 1$ the observations have neither linear structure in the logarithm of probability of test pass, nor in the logarithm of the odds of test pass.

We use $\Pr_{\mathrm{Obs}}(t_j = 0|\mathbf{x}), j = 1, 2$ to train the LNOR and the LR models. The results for probability of test pass given for two models and the training data are shown in Fig. 4 and Fig. 5, respectively for t_1 and t_2. Note that as $\Pr(t_1 = 0|\mathbf{x} = 1x_2x_1) = \Pr(t_1 = 0|\mathbf{x} = 0x_2x_1)$ and $\Pr(t_2 = 0|\mathbf{x} = x_3x_21) = \Pr(t_2 = 0|\mathbf{x} = x_3x_20)$, for each test only the combinations of the two contributing failure sources are shown. It is seen that for the extreme values of $w_{\mathrm{LNOR}} = 0$ and $w_{\mathrm{LNOR}} = 1$, as we expect, respectively, the LR and the LNOR models have perfect performance. As w_{LNOR} becomes less than 1 the performance of the LNOR degrades and, especially, for $\Pr(t_1 = 0|x_2x_1 = 00)$ and $\Pr(t_2 = 0|x_3x_2 = 00)$ the LNOR has poor performance for low values of w_{LNOR}. The LR model, however, provides robust performance as the structure of observed data varies from a log-probability-linearity to log-odds-linearity.

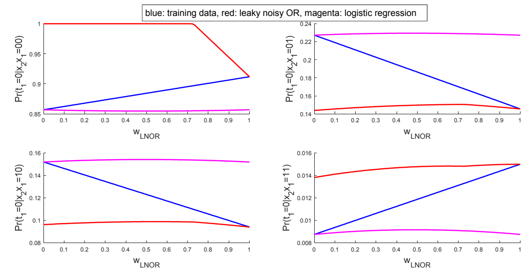

Figure 4: Comparison of probability of t_1 is passed given $\mathbf{x}$ for the LNOR and the LR models

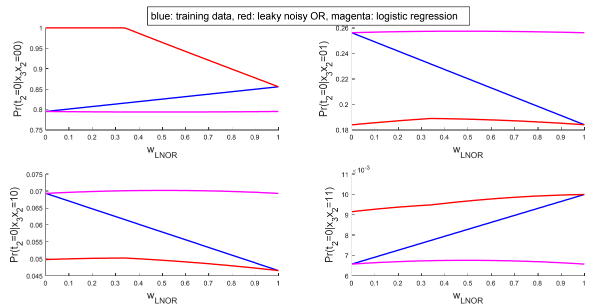

Figure 5: Comparison of probability of t_2 is passed given $\mathbf{x}$ for the LNOR and the LR models

Figures 6 and 7 show $\Pr(t_2t_1 = 11|\mathbf{x})\Pr(\mathbf{x})$ for all eight combinations of failure sources, for $p_{s_1} = 0.15, p_{s_2} = 0.10$, and $p_{s_3} = 0.05$. It is seen that while the LR model provides robust results, the LNOR results are off for low values of w_{LNOR} and for three combinations $x_3x_2x_1 = 000$, $x_3x_2x_1 = 001$, and $x_3x_2x_1 = 100$. The reason is that, for $x_3x_2x_1 = 100$, both inputs of test t_1 are zero, while for $x_3x_2x_1 = 001$, both inputs of test t_2 are zero, and for $x_3x_2x_1 = 000$, inputs of both tests t_1 and t_2 are zero, and we saw in Figures 4 and 5 that, the LNOR model poorly estimates $\Pr(t_j|00)$ when w_{LNOR} is low. In reality, as the number of failure sources become large and since tests are connected to some of them only, in many combinations of the failure sources, one or more tests are subjected to all-zero inputs, and hence in those combinations, the error would be large when w_{LNOR} is low. Figure 7 also shows that when both tests fail for any value of w_{LNOR},

$\Pr(t_2 t_1 = 11|\mathbf{x} \Pr(\mathbf{x}))$ has its highest value at $\mathbf{x} = 010$ and both models provide the same inference.

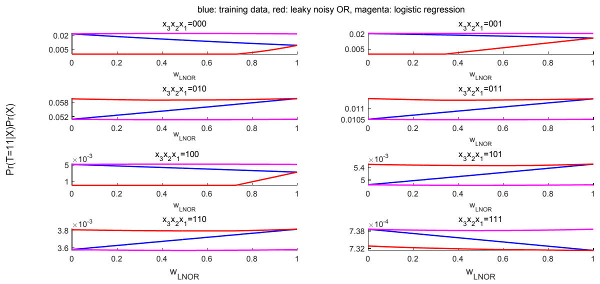

Figure 6: Comparison of $\Pr(T = 11|\mathbf{x})\Pr(\mathbf{x})$ for the LNOR and the LR models

6.2 Dual Cost Function and the MAP Estimate

Figures 8 and 9, respectively, show the dual cost function, i.e., equation (54), for the LNOR and the LR models. For the training of these models, we used $w_{\text{LNOR}} = 0.5$. Due to nonlinearities in (55), that is, $L_{2i}(\boldsymbol{\lambda}) = c_i(\boldsymbol{\lambda}) u\left(c_i(\boldsymbol{\lambda})\right)$, both dual cost functions have sharp corners making them non-differentiable. Note that for the LR model, based on (49), we have $\lambda_j = y_j$, and since y_j, as defined in (41), is a probability, the dual cost function for $\lambda_j > 1$ is undefined. However, for the LNOR model based on (49), we have $\lambda_j = \frac{y_j}{1-y_j}$, which is an odds function, and thus can take any nonnegative value.

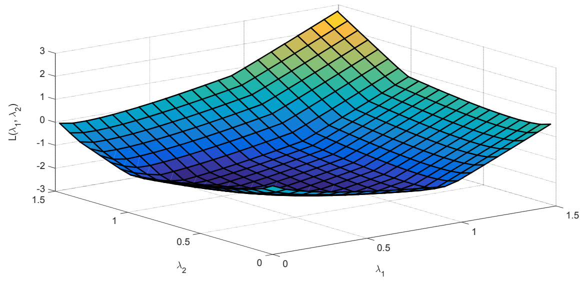

Figure 8: Dual cost function for the LNOR model

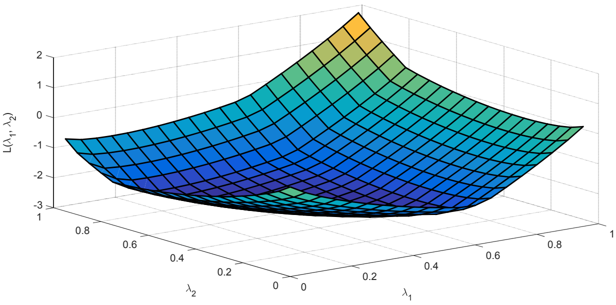

Figure 9: Dual cost function for the LR model

The minimum value of the dual cost function for the LNOR occurs at $(\lambda_1^*, \lambda_2^*) = (0.444, 0.716)$, and that for the LR occurs at $(\lambda_1^*, \lambda_2^*) = (0.343, 0.418)$. These result in $c_1(\boldsymbol{\lambda}^*)$ and $c_3(\boldsymbol{\lambda}^*)$ to be negative for both models, which means, based on (53), that $x_1^* = 0$ and $x_2^* = 0$. The value of $c_2(\boldsymbol{\lambda}^*)$ for both models is close to zero. When $c_i(\boldsymbol{\lambda}^*) = 0$, as the contribution of $c_i(\boldsymbol{\lambda}^*)\, u\left(c_i(\boldsymbol{\lambda}^*)\right)$ to the dual cost function is zero, irrespective of assigning 1 or 0 to $u\left(c_i(\boldsymbol{\lambda}^*)\right)$, decision based on (53) will be ambiguous. Therefore, for $c_i(\boldsymbol{\lambda}^*)$ close to zero, we should either check the primal cost function or use the set-covering ideas. As discussed in [Abdollahi *et al.*, 2016] for the DFA model, in practice, there are not many such cases. In the above example, comparing the primal cost function at $x_3 x_2 x_1 = 000$ and $x_3 x_2 x_1 = 010$ gives the optimal solution as $x_3^* x_2^* x_1^* = 010$.

7 Conclusion

In this paper, we discussed two graphical models, viz., DFA (Detection False Alarm) and leaky noisy OR (LNOR), and showed that the parameters of the LNOR model can be uniquely determined via the parameters of the DFA model. The reverse mapping, however, is not unique. The false alarms are viewed as errors in sensor measurements and processing in the DFA model, and as unmodeled parameters in the LNOR model. However, in both models, once the parameters are obtained via the fault injection-test output observations, both sensor measurement errors and unmodeled parameter effects are reflected in the respective parameters of the models. In logistic modeling, we presented two realization of the combinatorial case, one by assigning a weight to each failure combination and the other using a polynomial of degree m. Then, we showed a more tractable model by restricting the weights on the possible states of each failure source, rather than the combination of all failure states. This resulted in a logistic model similar to the DFA model in structure, in the sense that a zero state of any of the failure sources affects the probability of test outcomes in both models. Using the same approach that we used for showing the equivalence of the DFA and the LNOR models, we showed the equivalence of the restrictive model and the logistic regression (LR) model. Then, we devised a unified model that encompasses both the LNOR and the LR models. We presented the fault diagnosis problem as the *maximum a posterior* (MAP) inference problem for the unified model, and derived its solution using a Lagrangian relaxation method. Finally, we derived the dual cost function for this unified MAP inference problem. Using an example, we discussed the dual cost function. Simulation results show that the LR model is more robust with respect to the structure of the training data than the LNOR model.

Acknowledgments

The work reported in this paper was partially supported by NSF grants ECCS-0931956 (NSF CPS), ECCS-1001445 (NSF GOALI), CCF-1331850 (NSF CyberSEES). Pattipati's work was also supported by ONR grant N00014-10-1-0029, ONR grant N00014-12-1-0238, ONR grant N00014-16-1-2036, and QUBE grant N00173-12-2-C902. We thank NSF, ONR and QUBE for their support of this work. Any opinions expressed in this paper are solely those of the authors and do not represent those of the sponsors.

References

[Abdollahi *et al.*, 2016] Ali Abdollahi, Krishna R Pattipati, Anuradha Kodali, Satnam Singh, Shigang Zhang, and

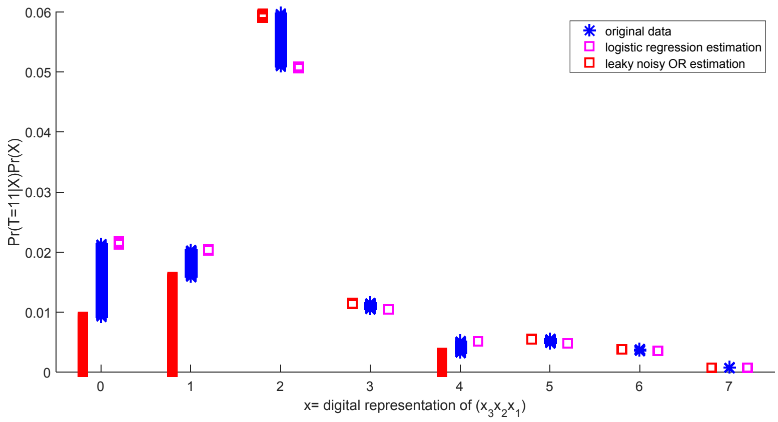

Figure 7: Comparison of $\Pr(T = 1|\mathbf{x})\Pr(\mathbf{x})$ for the LNOR and the LR models

Peter B Luh. Probabilistic graphical models for fault diagnosis in complex systems. In *Principles of Performance and Reliability Modeling and Evaluation*, pages 109–139. Springer, 2016.

[Athans *et al.*, 1975] M Athans, K-P Dunn, CS Greene, WH Lee, N Sandell, I Segall, and AS Willsky. The stochastic control of the f-8c aircraft using the multiple model adaptive control (mmac) method. In *Decision and Control including the 14th Symposium on Adaptive Processes, 1975 IEEE Conference on*, pages 217–228. IEEE, 1975.

[Bishop, 1995] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[Boel *et al.*, 1975a] R Boel, P Varaiya, and E Wong. Martingales on jump processes. ii: Applications. *SIAM Journal on Control*, 13(5):1022–1061, 1975.

[Boel *et al.*, 1975b] René Boel, Pravin Varaiya, and Eugene Wong. Martingales on jump processes. i: Representation results. *SIAM Journal on control*, 13(5):999–1021, 1975.

[Clark *et al.*, 1975] Robert N Clark, Douglas C Fosth, and Vincent M Walton. Detecting instrument malfunctions in control systems. *Aerospace and Electronic Systems, IEEE Transactions on*, (4):465–473, 1975.

[Frank and Köppen-Seliger, 1997] Paul M Frank and Birgit Köppen-Seliger. New developments using ai in fault diagnosis. *Engineering Applications of Artificial Intelligence*, 10(1):3–14, 1997.

[Handschin *et al.*, 1975] Edmund Handschin, Fred C Schweppe, Jurg Kohlas, and Armin Fiechter. Bad data analysis for power system state estimation. *Power Apparatus and Systems, IEEE Transactions on*, 94(2):329–337, 1975.

[Jones, 1973] Harold Lee Jones. *Failure detection in linear systems*. PhD thesis, Massachusetts Institute of Technology, 1973.

[Kerr, 1974] Thomas H Kerr. A two ellipsoid overlap test for real time failure detection and isolation by confidence regions. In *Decision and Control including the 13th Symposium on Adaptive Processes, 1974 IEEE Conference on*, pages 735–742. IEEE, 1974.

[Lainiotis, 1971] Demetrios G Lainiotis. Joint detection, estimation and system identification. *Information and control*, 19(1):75–92, 1971.

[Mehra and Peschon, 1971] Raman K Mehra and J Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7(5):637–640, 1971.

[Merrill, 1972] Hyde McCune Merrill. *Bad data suppression in state estimation, with applications to problems in power*. PhD thesis, Massachusetts Institute of Technology, 1972.

[Miller, 2013] R Miller. *Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations, Held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and Sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*. Springer Science & Business Media, 2013.

[Murphy, 2012] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[Rijmen, 2008] Frank Rijmen. Bayesian networks with a logistic regression model for the conditional probabilities. *International Journal of Approximate Reasoning*, 48(2):659–666, 2008.

[Schweppe and Handschin, 1974] Fred C Schweppe and Edmund J Handschin. Static state estimation in electric power systems. *Proceedings of the IEEE*, 62(7):972–982, 1974.

[Shakeri *et al.*, 1998] Mojdeh Shakeri, Vijaya Raghavan, Krishna R Pattipati, and Ann Patterson-Hine. Optimal and near-optimal algorithms for multiple fault diagnosis with unreliable tests. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(3):431–440, 1998.

[Theodoridis, 2015] Sergios Theodoridis. *Machine learning: a Bayesian and optimization perspective*. Academic Press, 2015.

[Willsky and Jones, 1976] Alan S Willsky and Harold L Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *Automatic Control, IEEE Transactions on*, 21(1):108–112, 1976.

[Willsky, 1976] Alan S Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12(6):601–611, 1976.