ON THE LINEAR SEPARABILITY OF DIAGNOSTIC MODELS

John W. Sheppard
Department of Computer Science
The Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218
jsheppa2@jhu.edu

Stephyn G. W. Butcher
Department of Computer Science
The Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218
stephyn@jhu.edu

Abstract—As new approaches and algorithms are developed for system diagnosis, it is important to reflect on existing approaches to determine their strengths and weaknesses. Of concern is identifying potential reasons for false pulls during maintenance. Within the aerospace community, one approach to system diagnosis—based on the *D*-matrix derived from test dependency modeling—is used widely, yet little has been done to perform any theoretical assessment of the merits of the approach. Past assessments have been limited, largely, to empirical analysis and case studies. In this paper, we provide a theoretical assessment of the representation power of the *D*-matrix and suggest algorithms and model types for which the *D*-matrix is appropriate. Finally, we relate the processing of the *D*-matrix with several diagnostic approaches and suggest how to extend the power of the *D*-matrix to take advantage of the power of those approaches.

INTRODUCTION

Within the aerospace community and similar communities producing large, complex systems (e.g., the Department of Defense), considerable attention has been given to developing diagnostic systems based on a specific modeling paradigm—dependency modeling. Many available tools map their models into the so-called "*D*-matrix" (from "dependency" matrix) and derive diagnostic strategies from this matrix. Recent research has even demonstrated a functional "equivalence" between a variety of graphical diagnostic models such as the behavioral Petri net, the bipartite Bayesian network, and the multi-signal flow model [13]. The multi-signal flow model is of particular interest because it is one of those examples where tools have been developed that use the *D*-matrix [5]. Motivated by the widespread use of models based on the *D*-matrix, the derivative "diagnostic inference model" has been proposed by the IEEE as a standard representation of this kind of model in IEEE Std 1232-2002 [11], and this standard is a candidate for inclusion in the DoD's automatic test system framework [7] and the associated Automatic Test Markup Language (ATML) initiative [2].

Previously, Sheppard and Kaufman asserted that false alarms generally arise from multiple sources: human error, unpredictable or unmodeled environmental conditions, instrument uncertainty, or test design issues [20], and such false alarms can lead to false pulls and unnecessary maintenance actions. The test community's desire to identify causes for false pulls during system maintenance motivates the work in this paper. In addition to false alarms, false pulls can be attributed to ineffective diagnostics arising from incomplete models, inaccurate models, or erroneous reasoning. We focus on incomplete diagnostic models in this paper.

In the following, we will consider the diagnostic problem from the perspective of pattern classification [8] and prove a significant result on the representation power of models based on the *D*-matrix. We believe this result is well known in the pattern classification community, but for some reason, the result is not as well known within the automatic test community. Specifically, we will prove that a diagnostic model that is based upon the *D*-matrix instantiates a linearly separable classification problem. Given this characteristic, we will then assess a number of diagnostic inference algorithms that, when applied to the *D*-matrix, either indicate limitations in diagnostic power or suggest approaches to mitigate the limitations due to linear separability.

DIAGNOSTIC INFERENCE MODELS

A common form of model used in diagnostic systems is the dependency model. This model can be identified by alternative labels such as the signal flow model, information flow model, the causal model, and the bipartite graphical model. One of the more complete descriptions of the model can be found in [23], and we summarize that description here. In the following, we will use the IEEE-standard term—the Diagnostic Inference Model, or DIM.

A DIM is built upon two fundamental model objects—the test (and its associated outcomes) and the diagnosis (or diagnostic conclusion). Tests include any source of information that can provide an indication of the health state of the system (including symptoms, safe-to-turn-on tests, readiness tests, and diagnostic tests), and diagnoses include any diagnostic conclusion one wishes to draw about the system (including no fault found).

Inference relationships between tests and diagnoses as well as among tests are represented with a directed graph capturing information "flow" through the system. Specifically, let $\mathbf{V} = \mathbf{T} \cup \mathbf{D}$ be a set of vertices, where **T** represents the set of tests and **D** represents the set of diagnoses. We define the set of directed edges **E** to be dependence relationships between tests and diagnoses (d_i, t_j) indicating a logical relationship corresponding to $d_i \Rightarrow t_j$. In other words, if diagnoses d_i is true, then test t_j will also be true and thereby detect the diagnosis. The directed graph corresponding to the logical relationships between tests and diagnoses can be represented in a bit-wise adjacency matrix, and this matrix has been named the *D*-matrix. An example of such a matrix is given in Figure 1. In this matrix, a cell having a value of 1, indicates the corresponding logical relationship. For example, this matrix show $d_1 \Rightarrow t_1$ but $d_1 \not\Rightarrow t_2$.[1]

[1] Similar implications can be specified between tests. Due to a subsumption property defined within [23] that enables derivation of these relationships from the edges (d_i, t_j), we will ignore those relationships in this paper.

	t_1	t_2	t_3	t_4	t_5	t_6
d_1	1	0	0	1	1	0
d_2	0	1	0	1	0	0
d_3	0	1	1	0	1	1
d_4	0	0	0	1	0	1
d_5	0	0	1	0	1	1

Figure 1. Bit-wise Adjacency Matrix—*D*-Matrix.

DIAGNOSTIC ALGORITHMS

A wide variety of inference algorithms have been proposed for fault diagnosis, many of which operate (or can operate) on the *D*-matrix. In the following, we will provide a brief overview of four such algorithms—rule based inference, set partitioning, Bayesian inference, and case based reasoning.

Rule Based Inference

Given the *D*-matrix representation for a diagnostic model, several types of algorithms have been used to process the models to perform diagnosis. When operating on the fundamental relationships where diagnosis-to-test and test-to-test relationships are specified, traditional rule-based inference methods have been used. In these cases, the rule $d_i \Rightarrow t_j$ is reversed using the logically equivalent form $\neg t_j \Rightarrow \neg d_i$, and algorithms such as forward chaining and backward chaining are applied [18]. If we let t_j denote test *j* failing and $\neg t_j$ denote test *j* passing then the challenge comes from considering the effect of multiple tests. Specifically, we find

$$\begin{aligned} d_i &\Rightarrow (t_j \wedge t_k \wedge \ldots) \\ \neg(t_j \wedge t_k \wedge \ldots) &\Rightarrow \neg d_m \\ (\neg t_j \vee \neg t_k \vee \ldots) &\Rightarrow \neg d_m . \end{aligned}$$

This becomes tricky for a chaining-type inference system and must be coupled with corresponding rules of the type $t_j \Rightarrow (d_i \vee d_k \vee \ldots)$, which fall out from the complete set of rules by disjuncting the rules with common consequents.

Set Partitioning

The most widely used algorithm for processing *D*-matrices is based on set partitioning and set intersection [23]. Usually, these algorithms impose a single fault assumption to reduce the computational complexity; however, recent tools (e.g., TEAMS and DSI eXpress). relax this requirement. Fundamentally, diagnosis operates by observing that tests indict or clear the diagnoses attached to them, as indicated by entries in the *D*-matrix. Sets of cleared and suspected diagnoses are maintained and updated as tests are performed. Whenever a test fails, the set of candidate diagnoses is updated as follows:

Let **S** be the set of suspect diagnoses already identified. Let **C** be the set of cleared diagnoses (identified when prior tests have passed). Let $\mathbf{I}_j$ be the set of diagnoses indicted by a test t_j failing. Then we update the set of suspect diagnoses using $\mathbf{S} \leftarrow \mathbf{S} \cap (\mathbf{I}_j - \mathbf{C})$. The set of cleared diagnoses is updated when test t_j passes as $\mathbf{C} \leftarrow \mathbf{C} \cup \mathbf{c}_j$. The process continues until some termination criterion is met such as reducing **S** to a set of diagnoses sufficient to apply a maintenance action.

Many model-based tools construct decision trees or paths based on test results. The most common approach to constructing such a tree is by choosing tests that maximize information gain [16], [23]. By considering the possible test outcomes, the set of possible diagnoses is partitioned, and a new subtree is constructed for each partition. This process continues recursively until some termination criterion is satisfied, and the result is a fault tree.

Recent approaches in constructing decision trees have also considered performing multiple tests at a particular node of the tree to reduce the size of the overall tree. Constructing such "oblique" trees also has advantages for building general decision trees because of the ability to consider tests that are correlated in some way [15].

Bayesian Inference

Recently, Bayesian methods have gained popularity, and a widely used Bayesian model is the bipartite network [20], [22]. Using this model, we assume the random variables in **D** (i.e., the diagnoses) are independent, as are the random variables in **T** (i.e., the tests). Now the characteristics of conditional independence allow for simple propagation of the probabilities from the tests to the diagnoses.

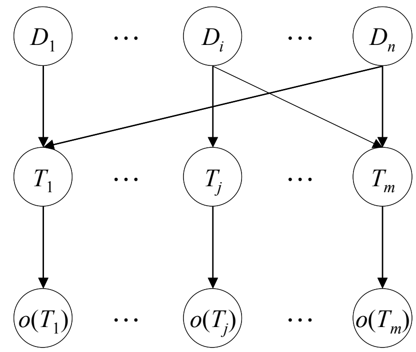

Figure 2. Bipartite Diagnostic Bayesian Network

Given the conditional independence of the diagnoses, one can compute the posterior probabilities of each of the diagnoses given the test results as follows. First, we will assume that we are using the network form presented in Figure 2 and partition the random variables into three sets: **D** (the diagnoses), **T** (the true test states), and **O** (the test observations). The evidence variables will be restricted to **O**.

$$\begin{aligned}\Pr(D_i \mid \mathbf{O}) &= \alpha \Pr(\mathbf{O} \mid D_i) \Pr(D_i) \\ &= \alpha \Pr(D_i) \sum_{T_j \in \mathbf{T}} \Pr(o(T_j) \mid T_j) \Pr(T_j \mid D_i)\end{aligned}$$

Here, α is a normalizer over the set **D**, equal to

$$\alpha = \sum_{D_i \in \mathbf{D}} \Pr(D_i) \sum_{T_j \in \mathbf{T}} \Pr(o(T_j) \mid T_j) \Pr(T_j \mid D_i) .$$

Assuming we are able to generate the probability distributions for nominal and faulty behavior, we consider the effects of locating the decision boundaries. For this discussion, we will draw on results from Bayes decision theory and its derivative, signal detection theory [8].

Observe that $\Pr(o(T_j) \mid D_i) \in \{0, 1\}$, so the members of the sum are restricted only to those tests that observe D_i. Because this corresponds exactly to the *D*-matrix, we only need to consider two things: $\Pr(D_i)$, which corresponds to the prior probability for D_i based on failure rate, and $\Pr(o(T_j) \mid T_j)$, which corresponds to the confidence value assigned to the observed test result. Using the Bayes' maximum *a posteriori* hypothesis, we determine the most likely diagnosis simply as

$$D_{MAP} = \arg\max_{D_i \in \mathbf{D}} \{\Pr(D_i \mid \mathbf{O})\}.$$

Case Based Reasoning

Case based reasoning (CBR) is a method of reasoning that combines elements of instance-based learning and data base query processing [1]. Test and diagnosis can use CBR in several ways. The simplest method involves defining a case as a collection of test results and attempting to determine an appropriate diagnosis given these results. The retrieval process is very simple. All of the cases are nothing more that feature vectors with an associated diagnosis $\langle t_1, t_2, \ldots, t_n; d_i \rangle$. The features in the feature vector correspond to test results and may be unknown. Retrieval then consists of "matching" the new case with all of the cases stored in the case base and selecting the most similar case.

When considering possible similarity metrics, numerical features are frequently compared using a member of the family of L_p norms. Let $\mathbf{x}'$ and $\mathbf{x}''$ be two feature vectors where any given x_i' corresponds to the *i*th feature in $\mathbf{x}'$ (similarly for $\mathbf{x}''$). These features could correspond, for example, to test results. An L_p norm is defined to be

$$L_p(\mathbf{x}', \mathbf{x}'') = \left(\sum_i (x_i' - x_i'')^p \right)^{1/p}.$$

The most common values for p are 1, 2, and ∞ and yield Manhattan distance, Euclidean distance, and max-norm distance respectively. Specifically, these metrics can be computed as:

$$L_1(\mathbf{x}', \mathbf{x}'') = \sum_i (x_i' - x_i'')$$
$$L_2(\mathbf{x}', \mathbf{x}'') = \sqrt{\sum_i (x_i' - x_i'')^2}$$
$$L_\infty(\mathbf{x}', \mathbf{x}'') = \sum_i \max\{x_i', x_i''\}$$

If we were using pass/fail results for testing, we would use either L_1 or L_2 (which would be equivalent). Note this is exactly what is done with fault dictionary-based diagnosis. With real values, we would most likely use L_2. Symbolic results are a bit more complicated and would require something like Stanfill and Waltz's "value difference metric" [24]. Regardless of the metric, retrieval would be done relative to test case *x* as

$$\mathbf{case} = \arg\min_{\forall c \in \mathbf{CASE\_BASE}} \{L_p(x, c)\}.$$

As alluded to above, the digital fault dictionary is an application of the case based approach that matches the *D*-matrix representation explicitly [21]. Fault dictionaries define a mapping from combinations of input vectors and output vectors to faults. Formally, this is represented as $FD: I \times O \rightarrow F$ where FD is the fault dictionary, *I* is the space of input vectors, *O* is the space of output vectors, and *F* is the space of faults. At a more basic level, this can be represented as $FD: \{0,1\}^n \times \{0,1\}^m \rightarrow F$.

We can convert the fault dictionary into a *D*-matrix by comparing test results in the presence of a fault to expected test results when the circuit is not faulty. We place a 1 in the corresponding cell of the matrix if these values are different and a 0 in the cell if they are the same. If the value in the cell is one, we claim the associated failure mode "causes" the given test to fail. If the value is zero, the presence of the associated failure mode will not be detected by the given test. Given a complete row in the matrix, we say that if the associated failure mode is present, then all of the tests associated with the failure mode (i.e., whose cells have a value of one) must fail. Conversely, if any of those tests pass, then the failure mode must not be present.

As an example, consider the simple circuit in Figure 3. The corresponding fault dictionary is in Table 1. In this case, the fault dictionary is simply the transpose of the *D*-matrix.

LINEAR SEPARABILITY

As we will show, diagnosis can be posed as a classification problem. A common concern arises when constructing any classifier of whether the underlying concept to be learned is linearly separable. If the underlying concept is linearly separable, then a variety of "simple" classifiers can be constructed to learn the concept, ranging from naïve Bayes classifiers [12] to single-layer perceptrons [14]. On the other hand, if the underlying concept is *not* linearly separable, then

Table 1. Fault Dictionary for Sample Circuit.

Test	**Fault Signatures**														
	a_0	a_1	b_1	c_1	d_1	f_0	f_1	g_1	i_0	i_1	j_1	k_0	k_1	m_0	nf
t_1	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0
t_2	1	0	0	1	0	0	1	1	0	0	1	0	1	0	0
t_3	1	0	1	0	0	0	1	1	0	0	1	0	1	0	0
t_4	0	1	1	1	1	0	1	1	1	1	1	0	1	0	1
t_5	1	0	0	1	1	1	0	1	0	1	1	1	1	0	1
t_6	1	0	1	1	1	1	0	1	0	1	1	1	1	0	1
t_7	1	0	1	0	1	1	0	1	0	1	1	1	1	0	1
t_8	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0

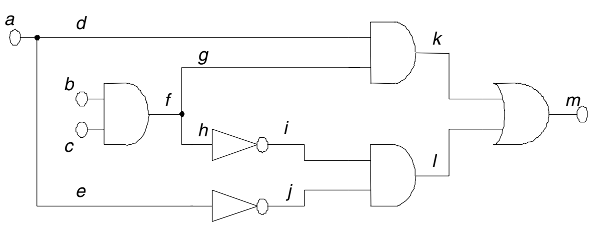

Figure 3. Sample Combinational Circuit.

more complex classifiers must be used (e.g., decision trees [16], augmented Bayes classifiers [9], [19], or multi-layer perceptrons [17]) to learn the concept.

In the following, we will discuss the relationship between linear separability and diagnostic systems based on the *D*-matrix. First, we will define formally what a *D*-matrix is.

Definition 1: Let **D** be a set of diagnoses. Let **T** be a set of tests. Assume each $d_i \in \mathbf{D}$ is a Boolean variable such that *eval*(d_i) $\in$ {0,1}. Assume each $t_j \in \mathbf{T}$ is also a Boolean variable such that *eval*(t_j) $\in$ {0,1}. Then a *diagnostic signature* is defined to be the vector

$$\mathbf{d}_i = [eval(t_1), \ldots, eval(t_{|\mathbf{T}|})] \, ,$$

where

$$eval(t_j) = \begin{cases} 1 & \text{if } t_j \text{ detects } d_i \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2: A *D-matrix* is defined to be the set of diagnostic signatures $\mathbf{d}_i$ for all $d_i \in \mathbf{D}$.

Note that the diagnostic problem can be posed as a classification problem. Formally, we define a Boolean classifier as follows.

Definition 3: Let **C** be a set of concepts (or classes). Let **A** be a set of attributes (or features) of some object or concept. Assume each $c_i \in \mathbf{C}$ is a Boolean variable such that *eval*(c_i) $\in$ {0,1}. Assume each $a_j \in \mathbf{A}$ is also a Boolean variable such that *eval*(a_j) $\in$ {0,1}. Then a *feature vector* is defined to be the vector

$$\mathbf{c}_i = [eval(a_1), \ldots, eval(a_{|\mathbf{A}|})] \, ,$$

where

$$eval(a_j) = \begin{cases} 1 & \text{if } a_j \text{ is an attribute of } c_i \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4: A *Boolean classifier* is a mapping $f : \mathbf{A} \rightarrow \mathbf{C}$.

By combining the definitions above with the following assignments

A = **T**, **C** = **D**, and $\mathbf{c}_i = \mathbf{d}_i$

the diagnosis problem is shown to be equivalent to a classification problem.

There are some very specific restrictions in the above definition, however. First, we are assuming all attributes are Boolean (i.e., we are not permitting nominal or real-valued attributes at this point). Second, we are assuming there is a single feature vector sufficient to characterize each class.

We will see that the restriction on feature vectors is a significant restriction that rarely, if ever, occurs in practice. In fact, the general classification problem does not assume a corresponding "*D*-matrix" but generates the classifier from a set of training instances where multiple, varying feature vectors can exist for each concept class. Nevertheless, most graphical model-based diagnostic systems (e.g., dependency models, multi-signal flow models, and fault dictionary models) are based on the *D*-matrix formalism.

From this point forward, we will use the language of diagnosis rather than classification unless we need to apply a result from classification theory. At that point, the association to diagnosis will be made explicit.

First, consider the case where we have only two diagnoses. Arguably, this is the simplest diagnostic problem since only one diagnosis would be trivially true (assuming a closed set).

Definition 5: Two concept classes are *linearly separable* if and only if there exists a linear function $\mathbf{w}^T\mathbf{a} - \theta = 0$ such that one concept is identified when $\mathbf{w}^T\mathbf{a} - \theta > 0$ and the other concept is identified when $\mathbf{w}^T\mathbf{a} - \theta \leq 0$, where $\mathbf{w}$ is a column vector of weights, $\mathbf{a}$ is a column vector of attributes, and θ is a threshold.

Theorem 1: Given two diagnoses d_1 and d_2 with distinct diagnostic signatures $\mathbf{d}_1$ and $\mathbf{d}_2$ (i.e., $\mathbf{d}_1 \neq \mathbf{d}_2$), then d_1 and d_2 are linearly separable.

Proof: The logical representation of the associated Boolean classifier is given as

$$d_i \Leftrightarrow \prod_j (eval(t_j) = \mathbf{d}[j])$$

where the product symbol, Π corresponds to logical AND. We can convert this into a linear expression as follows. First, let $\mathbf{w} = [1 \ \ldots \ 1]^T$ (which basically removes $\mathbf{w}$ from the function). Next, let

$$a_j = 1 - \left| eval(t_j) - \mathbf{d}[j] \right| + \varepsilon \ ,$$

where ε is any positive constant. Now

$$\prod_j (eval(t_j) = \mathbf{d}[j]) = \prod_j a_j \ ,$$

and this product is maximized if and only if the signature d_i is matched by the set of test outcomes $T = [eval(t_1), \ldots, eval(t_{|\mathbf{T}|})]$ for some set of actual test evaluations t_i. Then define the following:

$$C(T) = \log \prod_j a_j = \sum_j \log a_j > \theta \ .$$

We can use this linear function describing diagnosis d_i as a discriminant by setting $\theta = (|\mathbf{T}| - 1)\log(1+\varepsilon) + \log(\varepsilon)$. Specifically, when $C(T) > \theta$, the test vector T will be classified as diagnosing d_i. Thus, for a pair of diagnostic signatures, classification (i.e., diagnosis) is linearly separable. ❑

What happens when there are more than two distinct diagnostic signatures? As we will see, the result is the same.

Corollary 1: Given a set of diagnoses $\mathbf{D}$ with distinct diagnostic signatures $\mathbf{d}_1, \ldots, \mathbf{d}_{|\mathbf{D}|}$, then the set $\mathbf{D}$ is linearly separable.

Proof: This is evident from the fact that our linear discriminant $C(T) > \theta$ is satisfied whenever T matches the corresponding vector. Given the way θ was defined, even one mismatch will cause $C(T)$ to be less than θ. ❑

IMPLICATIONS FOR DIAGNOSTIC ALGORITHMS

The immediate conclusion to be drawn from the above analysis is that the *D*-matrix provides a simplistic view of the diagnostic problem. As defined, the *D*-matrix is a "conjunctive" model in that it specifies the logical-AND of the test results associated with a particular diagnosis. The advantage is that most diagnostic algorithms are able to process such models easily.

Now, if such a simple model can address real-world fault diagnosis requirements, then the underlying concept so modeled must be linearly separable. In such cases, a variety of "simple" classifiers can be constructed to learn the concept, ranging from naïve Bayes classifiers [12] to single-layer perceptrons [14]. On the other hand, if such a simple model cannot address real-world fault diagnosis requirements, then the underlying concept is probably *not* linearly separable. This means a more complex model and associated classifier must be used (e.g., decision tree [16], augmented Bayes classifier [9], [19], or multi-layer perceptron [17]) to represent the concept.

To illustrate the limitations of linearly separable models, suppose we wish to construct a system-level model where each signature in the *D*-matrix corresponds to a single subsystem that might be faulty.[2] As a specific example, suppose we are attempting to fault isolate the stability augmentation system (SAS) of a helicopter. The SAS consists (at a minimum) of three gyros (one each for roll, pitch, and yaw), three servos (again, one each for roll, pitch, and yaw), and a mixing circuit. Fischer and Sivahop describe a SAS design consisting of redundant gyros feeding a mixing circuit to drive the servos [9]. The purpose of the mixing circuit is to compare digital and analog inputs and to determine the appropriate mix of roll, pitch, and yaw corrections. Consider the mixing circuit, which one could argue serves as a potential single-point failure for the SAS. The mixing circuit will determine differential corrections based on offset from a target orientation and apply, for example, *PID* control laws to signal the appropriate controls to the effectors.[3]

In keeping with the Fischer and Sivahop design, we have six gyros (due to redundancy in the three axes). Further, three of the gyros provide analog signals, and three provide digital signals to the mixing circuit. Finally, suppose that tests are constructed that only compare the aircraft orientation as inputs to the corrections generated by the servos. Three tests, one for roll, pitch, and yaw, are assumed. This, of course, yields an underspecified diagnostic model in that the three tests are only capable of differentiating at most eight diagnoses. Our SAS has 11 diagnoses (six gyros, three servos, the mixing circuit, and no-fault) and considerable ambiguity.

In spite of the fact the model is under-specified, let us focus on failure of the mixing circuit. Clearly, this mixing circuit embodies several possible failure modes; however, our assumptions have

Table 2. *D*-Matrix for Stability Augmentation System

	Roll Test	Pitch Test	Yaw Test
Roll Gyro 1	1	0	0
Roll Gyro 2	1	0	0
Pitch Gyro 1	0	1	0
Pitch Gyro 2	0	1	0
Yaw Gyro 1	0	0	1
Yaw Gyro 2	0	0	1
Roll Servo	1	0	0
Pitch Servo	0	1	0
Yaw Servo	0	0	1
Mixing Circuit	1	1	1
No Fault	0	0	0

reduced the model to consider a single, global failure mode of the mixing circuit as a whole. Consider the *D*-matrix in Table 3. There are two different approaches to using this matrix for performing fault diagnosis—1) perform all of the tests and compare the result to the fault signatures or 2) run the tests incrementally and reduce the space of possible faults as we go (analogous to the set-partitioning approach).

Consider the first approach. The problem with this matrix is that failure of the mixing circuit requires all three tests to detect that failure to correctly diagnose the problem. Unfortunately, it is possible that the mixing circuit may, as suggested above, fail in several ways that do not involve all three dimensions of control. Indeed, certain failure modes could result in canceling effects between the redundant gyros and the associated servo, thus leading to complex dependence relationships between the components of the SAS that are not linearly separable. In fact, if we only have two tests that fail, the best we could do is find the nearest match (except all other faults are equally distant), guess based on failure rate information, or declare an error. Notice also that removing any of the tests from the signature would end up missing failure modes involving the associated axis.

The other approach involves an incremental evaluation of the tests. This approach shows some promise. For example, suppose we run the Roll Test and the Pitch Test and they both fail. At this point, an incremental diagnostic system would halt and declare the mixing circuit as faulty, and this is probably correct. On the other hand, suppose the fault involves an incorrect mixing of

[2] The *D*-matrix has long been proposed as an excellent candidate for hierarchical, system-level modeling. The point of this discussion is to identify a necessary condition for this to proposal hold.

[3] *PID* control refers to a control system utilizing corrections “proportional” to the error with corrections dampened through a “derivative” term and stability maintained through an “integral” term. Thus, *PID*-control refers to “Proportional-Integral-Derivative” control.

roll and yaw corrections. This time, the Roll Test fails but the Pitch Test passes. At this point, the diagnostic system would halt and declare the roll section (one of the roll gyros or the roll servo) as being faulty. Thus a single diagnostic signature is insufficient to differentiate the mixing circuit from other diagnoses in the model.

Consider a seemingly simple extension to the model where we do not require the diagnoses to have one and only one signature. Without loss of generality, assume a particular diagnosis d_i. If we permit two (or more) signatures to appear in the *D*-matrix, each labeled with diagnosis d_i, we have

$$\mathbf{d}_i^1 = [eval^1(t_1), \ldots, eval^1(t_{|\mathbf{T}|})]$$

as well as

$$\mathbf{d}_i^2 = [eval^2(t_1), \ldots, eval^2(t_{|\mathbf{T}|})] .$$

This is equivalent to

$$\begin{aligned}\mathbf{d}_i &= [eval^1(t_1), \ldots, eval^1(t_{|\mathbf{T}|})] \\ &\vee [eval^2(t_1), \ldots, eval^2(t_{|\mathbf{T}|})].\end{aligned}$$

Thus we are now able to represent disjunctive concepts as well. Notice that by permitting "disjuncted" test signatures, any model we build will be able to represent a full disjunctive normal form (DNF). Since DNF can represent any propositional logic expression, this "simple" extension introduces considerable complexity into such models, including the ability to represent nonlinearly separable concepts (i.e., diagnoses).

Returning to the example of the stability augmentation system, note that this extension is exactly what is required to refine the diagnosis. In this case, multiple signatures would be added to the *D*-matrix, all with the same class label. This is contrary to the traditional *D*-matrix, but addition of these signatures a) improves the "resolution" of diagnosis by modeling the relevant failure modes and b) enables that improved "resolution" to cover potential interdependencies between the failure modes. In other words, the *D*-matrix is no longer limited to modeling linearly separable classes.

Can the four inference algorithms we discussed previously handle this increased complexity? Looking at each in turn, we see that they can or are able to be suitably modified to do so.

<u>Rule-Based Inference:</u> Combinations of rules can be combined and converted into clause form (or even Horn clause form) for diagnosis. Diagnostic signatures are simply rules, so incorporating additional signatures simply adds rules to the rule base. In addition, rules covering multiple diagnoses might be able to be simplified for more efficient inference. Therefore, current rule-based systems and satisfiability solvers can adapt to cover the more complex rules [6].

<u>Set Partitioning:</u> Decision trees inherently form a partitioning of the set of diagnoses as tests are performed. Incorporating additional signatures into the *D*-matrix offers no increased difficulty in constructing decision trees and can, in fact, provide useful information for simplifying the structure of the tree. Alternative splitting criteria and the use of oblique and nonlinear splits also increase the power of the overall approach [15].

<u>Bayesian Inference:</u> The mapping of the *D*-matrix to a Bayesian model reduces to applying the naïve Bayes assumption to the dependencies. This is a natural fit for the *D*-matrix since naïve Bayesian inference is only able to solve linearly separable problems. Extending the Bayesian approach can be handled like the set partitioning approach; however, this is not efficient. An alternative approach is to augment the naïve Bayes network to capture resulting dependencies in the model [8], [19].

<u>Case Based Reasoning:</u> CBR naturally handles non-linearly separable concepts. The primary deficiency with the CBR approach relative to fault dictionaries or the *D*-matrix representation (even augmented with additional signatures) is the compactness assumption. Specifically, CBR requires the case base to approximate the underlying distribution of the data; otherwise, noisy or missing data will lead to misclassification [8].

In concluding this section, notice that by permitting "disjuncted" test signatures, we have permitted a full disjunctive normal form representation to be included in each of the models discussed above.[4] Since DNF can

[4] We could include these separate signatures with distinct diagnoses in the current *D*-matrix and then use a management application to differentiate them. At issue is the fact that we now need to consider all of the combinations with which a system might fail. The advantage to pattern classification-based approaches is their ability to "generalize" and not require all combinations.

represent any propositional logic expression, this "simple" extension introduces considerable complexity into the model, including the ability to represent nonlinearly separable concepts (i.e., diagnoses).

As a final comment, we should point out that the approach described for expanding the *D*-matrix seems to solve our problem, but potentially that solution is at a great cost. Specifically, we are suggesting that all we need to do is introduce another layer of specificity in the model where diagnosis d_i has more than one signature (corresponding to each of the failure modes of d_i). Unfortunately, it is possible that the more detailed failure mode may also have nonlinearities being rolled into the signature. This would suggest further refinement. In the limit, we would see a system level model being required to include all possible failure modes, thus eliminating any benefit attributed to the *D*-matrix for creating hierarchical models.

In each of the algorithms discussed above, with the possible exception of set partitioning, it is possible that models can be constructed based on available performance data to capture these nonlinearities in a more compact fashion. This, in fact, is the primary area of current and future research for the authors. See [3] and [19] for related work on this topic.

CONCLUSION

Throughout this paper, we considered the applicability of several diagnostic inference strategies to a common diagnostic model based on the *D*-matrix. We examined the theoretical properties of the *D*-matrix and proved that, under normal assumptions, the *D*-matrix inherently supports only linearly separable diagnostic signatures.

We are interested in identifying potential causes for test and diagnostic error in the maintenance process. Tools based on *D*-matrices are pervasive and provide the foundation for determining diagnostic strategies for maintenance manuals and test programs. Given our conclusions, if models have been constructed such that only a single conjunctive fault signature is provided for generating these strategies, even though associated inference techniques can support nonlinearly separable diagnoses, the model effectively cripples these techniques by hiding additional nonlinear dependencies in the system.

Significantly, the *D*-matrix can be extended quite easily to model nonlinearly separable diagnoses. We noted that such nonlinearities arise whenever there are multiple correct signatures that lead to the associated diagnosis. By permitting each of these signatures to be included in the model, the ability to find efficient diagnostic strategies is now much more difficult. Nevertheless, these diagnostic strategies can be constructed using existing heuristic techniques, and the result is we now have the ability to account for the deficiency of the traditional *D*-matrix.

ACKNOWLEDGMENTS

The authors would like to thank Mark Kaufman and Greg Bowman for reviewing an extended version of this paper.[5] The reviews by both Mark and Greg significantly tightened the discussion of both papers, and their efforts are greatly appreciated.

REFERENCES

[1] Aamodt, A. and Plaza, E., "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *AI Communications*, Vol. 7, No. 1, 1994, pp. 39–59.

[2] Automatic Test Markup Language Consortium website, http://www.atml.org.

[3] Butcher, S., Sheppard, J., Kaufman, M., and MacDougall, C., "Experiments in Bayesian Diagnostics with UID-Enabled Data," to appear in *AUTOTESTCON Conference Record*, New York: IEEE Press, September 2006.

[4] Cormen, T., Leiserson, C., Rivest, R., and Stein, C., *Introduction to Algorithms*, second edition, Cambridge, MA: The MIT Press, 2001.

[5] Deb, S., Pattitpati, K., Raghavan, V., Shakeri, M., and Shrestha, R., "Multi-Signal Flow Graphs: A Novel Approach for System Testability Analysis and Fault Diagnosis, *IEEE*

[5] The extended version has been submitted to the Journal of Electronic Testing: Theory and Application and includes an analysis of the complexity of building optimal fault trees using the linearly separable formalism.

Aerospace and Electronic Systems Magazine, Vol. 10, No. 5, May 1995, pp. 14–25.

[6] Dechter, R., *Constraint Processing*, San Francisco: Morgan Kaufman, 2003.

[7] *DoD Automatic Test System Framework Roadmap*, May 2005.

[8] Duda, R., Hart, P., and Stork, D., *Pattern Classification*, New York: John Wiley & Sons, 2001.

[9] Fischer, W. C. and Sivahop, A. *Tested, Dissimilar Stability Augmentation System*, United States Patent #4,313,201, January 26, 1982.

[10] Friedman, N., Geiger, D., and Goldszmidt, M., "Bayesian Network Classifiers," *Machine Learning*, 29:131–163, 1997.

[11] IEEE Std 1232-2002. *Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE)*, Piscataway, NJ: IEEE Standards Association Press, 2002.

[12] Ling, C. and Zhang, H. "The Representational Power of Discrete Bayesian Networks," *Journal of Machine Learning Research*, Vol. 3, 2002, pp. 709–721.

[13] Luo, J., Tu, H., Pattipati, K., Qiao, L., and Chigusa, S., "Graphical Models for Diagnosis Knowledge Representation and Inference," *IEEE Instrumentation and Measurement Magazine*, to appear, 2006.

[14] Minsky, M. and Papert, S., *Perceptrons*, Expanded Edition, Cambridge, MA: The MIT Press, 1988.

[15] Murthy, S., *On Growing Better Decision Trees from Data*, PhD Dissertation, Department of Computer Science, The Johns Hopkins University, 1996.

[16] Quinlan, J. R., "Induction of Decision Trees," *Machine Learning*, Vol. 1, No. 1, 1986, pp. 81–106.

[17] Rumelhart, D., Hinton, G., and Williams, R., "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing*, eds. D. Rumelhart and J. McClelland, Cambridge, MA: The MIT Press, 1986, pp. 318–362.

[18] Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach*, 2nd edition, Upper Saddle River, NJ: Prentice Hall, 2003.

[19] Sheppard, J., Butcher, S., Kaufman, M., and MacDougall, C., "Not-So-Naïve Bayesian Networks and Unique Identification in Developing Advanced Diagnostics," *Proceedings of the IEEE Aerospace Conference*, New York: IEEE Press, March 2006.

[20] Sheppard, J. and Kaufman, M., "A Bayesian Approach to Diagnosis and Prognosis Using Built In Test," *IEEE* Transactions *on Instrumentation and Measurement*, Special Section on Built-In Test, Vol. 54, No. 3, June 2005, pp. 1003–1018.

[21] Sheppard, J. and Simpson, W., "Improving the Accuracy of Diagnostics Provided by Fault Dictionaries," *Proceedings of the 14th IEEE VLSI Test Symposium*, Los Alamitos, CA: IEEE Computer Society Press, 1996, pp. 180–185.

[22] Shwe, M. and Cooper, G., "An Empirical Analysis of a Likelihood-Weighting Simulation on a Large, Multiply-Connected Medical Belief Network," *Computers and Biomedical Research*," 24, 453–475, 1991.

[23] Simpson, W. and Sheppard, J., *System Test and Diagnosis*, Norwell, MA: Kluwer Academic Publishers, 1994.

[24] Stanfill, C. and Waltz, D., "Toward Memory-Based Reasoning," *Communications of the ACM*, Vol. 29, No. 12, 1986, pp. 1213–1228.