Not-So-Naïve Bayesian Networks and Unique Identification in Developing Advanced Diagnostics

John W. Sheppard and Stephyn G. W. Butcher
Department of Computer Science
The Johns Hopkins University
3400 North Charles Street
Baltimore, MD 21218
jsheppa2@jhu.edu and stephyn@apl.jhu.edu

Mark A. Kaufman and Craig MacDougall
NSWC Corona Division
US Navy
PO Box 5000
Corona, CA 91718
mark.kaufman@navy.mil and craig.macdougall@navy.mil

Abstract—Problems in accuracy and effectiveness in system diagnosis and prognosis arise from constructing models from design data that do not match implementation, failing to account for inherent uncertainty in test data, and failing to account for characteristics unique to specific units due to variations in usage, environment, or other factors. Large sums of money have been expended by owners of these systems, but little improvement in measures such as retest-OK rate and cannot duplicate rate has been reported. In fact, simply losing track of where specific units are located has resulted in substantial losses of money. In this paper, we study the problem of performing diagnosis and prognosis on systems and describe an approach to building models based on data collected about specific units. We rely on the emerging Department of Defense (DoD) Unique Identification (UID) program that is focusing on obtaining this data and apply Bayesian methods for constructing such diagnostic models. Specifically, we discuss an alternative class of Bayesian model that we call the "not-so-naive" Bayesian network (NBN). We also discuss the concept of the NBN in the context of the UID program as a means of tracking and deriving probabilities for creating the network. Finally, we focus on the specific problems encountered and lessons learned from working with a large, real-world database for the US Navy's STANDARD Missile.

TABLE OF CONTENTS

1. INTRODUCTION

Department of Defense inventories of weapon systems, support systems, parts, and supplies are massive. Much of the logistics process is devoted to planning for distribution and sparing of the assets contained in these inventories to maximize mission readiness. Unfortunately, the task of tracking each of these assets to ensure the right people are getting needed assets and the right people can locate those assets is monumental. The DoD has recognized the importance of improving approaches to asset management and has created Unique Identification numbers (UIDs) to improve the situation. UIDs will be used to track financial and contract records and obtain location and status information about parts in DoD inventory. UIDs will also support data collection for weapon systems from build, test, operations, maintenance, repair, and overhaul histories. In addition to improving the overall logistics process, UIDs offer an opportunity to utilize asset-specific data to improve system maintenance and support as well. This paper discusses the initial results of an Office of the Secretary of Defense (OSD) Pilot Project to implement UID on the STANDARD missile in the context of maintenance and support. In particular, it discusses the opportunities presented by the use of UID in the context of system diagnosis and prognosis.

Recent advances in applying Bayesian methods to system diagnosis have indicated the potential to extend these results to system prognosis as well. So called “dynamic Bayesian networks” (DBN) enable modeling system behavior by tracking faulty and fault-free conditions through time. The primary disadvantage to applying Bayesian techniques is the computational complexity associated with inferring results from evidence. For complex weapon systems, a simple Bayesian network can require several hundreds or even thousands of nodes. However, DBNs replicate these nodes over various time steps making inference much more complex. Even “simple” bipartite networks, such as those used in the “Quick Medical Reference-Decision Theoretic” (QMR-DT) system [1], require time that is exponential in the number of nodes to draw an exact inference.

0-7803-9546-8/06/$20.00 © 2006 IEEE
Paper 1572, Revision 4

One common approach to handling such computational complexity is through the application of naïve Bayesian classification. Naïve Bayesian classification constructs a network by assuming away joint dependencies in the model. The advantage of this approach is that the computational complexity of exact inference on general Bayesian networks (which is NP-hard, meaning no efficient algorithm is known to exist) reduces to polynomial time. The disadvantage is that one would expect such a strong independence assumption to limit performance. Surprisingly, this is often not the case. Several sources, including results reported herein, demonstrate the naïve Bayesian classifier to perform very well. Nevertheless, we contend potential accuracy is lost and can be recovered by identifying some of these dependencies.

One of the goals of our research is to investigate possible improvements to the naïve Bayesian approach by using an alternative class of Bayesian model we call the "not-so-naïve" Bayesian network (NBN). The NBN starts out as a naïve Bayesian network by assuming evidence nodes are conditionally independent given the class nodes. The NBN breaks with the naïve Bayes model by adding highly dependent relationships back with the goal of improving accuracy of classification. At the extremes, NBNs range from simple, canonical naïve Bayes classifiers to complete bipartite Bayes networks. Any particular NBN's position on that spectrum can be tuned according to the level of accuracy required and the computational cost that can be afforded. We discuss the application of the NBN approach in the context of the DoD's UID program, and the data available through that program, as a means of deriving probabilities for creating Bayesian networks for system diagnosis and prognosis.

2. DIAGNOSIS/PROGNOSIS AND UID

The DoD has a significant concern about the tracking and support of individual systems within their inventory. After Desert Storm, 35,000 shipping containers of supplies went unused and now needed to be redeployed. The task of returning these containers (and their contents) to the supply system or redeploying them to other theaters was monumental. The GAO concluded that visibility and accountability over the $3.4 billion of material was lost [2]. Incomplete tracking of DoD hardware location and history was the primary reason for the loss of accountability. In addition, valuable location, maintenance, reliability, and diagnostic information was not properly obtained, analyzed, or retained. The inability to access and use this information presented a significant obstacle to making effective program decisions, reducing maintenance and support costs, and enabling next-generation approaches to system support (e.g. advance diagnostics, reliability centered maintenance, and prognostics).

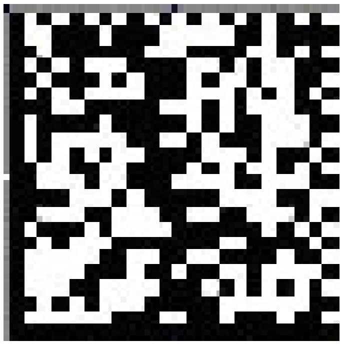

Figure 1–Sample UID data matrix providing specific identification information for a US Navy unit. The data matrix is analogous to a two-dimensional bar code.

The DoD began several initiatives to get more effective control over its logistics. Two of these initiatives, Unique Identification (UID) [3] and Radio Frequency Identification (RFID) [4] are of interest in this paper. Both UID and RFID are DoD requirements [5],[6]. In short, between these two initiatives the location and history of individual items will be tracked worldwide.

RFID uses Radio Frequency tags on pallets, shipping containers, etc. to track a unique number. That number is in the supply database and is linked to the contents, destination, and origin of the package. UID uses the data matrix (which is *not* RF) illustrated in Figure 1 to identify individual items. The data matrix is similar to a two-dimensional representation of a bar code, is scalable, can range in size from a few tenths of an inch to up to 14 inches, and can represent up to 2,000 characters. A crucial part of UID is the unique number – the Unique Item Identifier (UII). The UII is not just assigned by the manufacturer or depot. There is a protocol followed to generate the UII [7],[8]. The number is then uploaded to a DoD registry where it is checked for uniqueness and logged into the system. The UID that encodes the UII is then linked to a particular part for the lifetime of the part. UIDs can be applied as labels, plates, laser etched directly, and a through variety of other techniques [9]. Proper identification of an asset is essential to correlating pedigree and reliability prediction of the asset as well as enabling reliability centered maintenance, advanced diagnostics and prognostics.

Tracking serialized items is a significant challenge. Serial numbers and even part numbers are often "recycled" by manufacturers. Rebuilding an item from one configuration to another adds further confusion. The application of UID will make serialized item tracking easier and more effective. There are a number of databases that have track-serialized item's history, but they are limited to a single program. UID will enable the large-scale accumulation of failure, repair and maintenance data on a wide variety of systems.

Each time a maintenance action is executed that installs, removes, inspects, tests, or services a missile, section or major component, the part number, serial number, and in some cases manufacturer lot must be manually entered into a computer. This manual key entry is time consuming and error prone, which causes inventory inaccuracies. These discrepancies require quality checks to ensure the errors are caught and corrected. Correcting problem assets in many cases delays data from entering the system. In addition, part number and serial number errors prevent the linking of data from various data sources, for example, linking inventory with reliability prediction tools.

The GAO has frequently pointed out the Federal Government's inability to track assets, the large amount of waste associated with this inability, and the necessity of tracking assets for sound financial management. In the long term, utilizing UID, the DoD will be able to make more informed decisions on missions by having the ability to consistently locate, control and evaluate assets. The UID will also establish a common interface across all organizational boundaries.

One of the principal criticisms of current approaches to system prognosis is that these approaches rely on "statistic" properties collected over populations of systems. Yet it is well known that while one can reasonably expect average behavior to conform to these statistics (assuming the collected statistics actually relate to that specific class of system) individual members of the population can exhibit significant variation from the expected value. Even most diagnostic systems work with "representative models" of the system being tests and rarely account for unique characteristics of the particular system.

Within the context of this work, creating specialized models by specific asset can support the UID program. This has the advantage of incorporating operating and environmental conditions for the specific asset into the model and tracking the behavior of that specific model over time. The UID program permits tracking to occur at the specific system (and even component) level, thus providing a much finer grained view into the past performance of the system. This permits more focused analysis to improve predictions on future performance and future failure as well.

3. DIAGNOSTIC BAYESIAN NETWORKS

Developing system models for diagnosis is complex and often depends on a detailed understanding of system performance and test engineering. Learning diagnostic models from field maintenance data offers considerable potential to develop or refine diagnostics for fielded systems. Simulation can also be used to generate data for purposes of learning. Several approaches exist for learning such models including case based reasoning, decision tree induction, neural networks, and Bayesian methods. We suggest applying Bayesian methods to diagnosis because they derive classification "rules" based on sound mathematical principles (namely, probability theory), they can adapt easily as more data is obtained, and they have been demonstrated empirically to perform well on a broad range of classification problems.

Previously, we provided a detailed derivation of a simple model for Bayesian diagnosis [2],[11]. Here we summarize those results by pointing to two specific issues in Bayesian networks. First, determining the appropriate structure of the network can be difficult in that it requires a detailed understanding of the random variables of the problem to be solved and the conditional probability relationships between those random variables. In fault diagnosis, the first step of identifying the random variables is relatively straightforward because the random variables correspond to the tests and diagnoses.

Determining the appropriate relationships between the random variables and their conditional probabilities is more problematic. Initially, we assume we are able to determine which fault is detected by which test and that the tests were designed with such detection in mind. Thus we do not need to determine the actual relationships. We only need to consider the probabilities on those relationships. An approach based on instrument uncertainty was provided in [11], but in this paper, we derive the probabilities from a set of training data corresponding to actual test results and associated diagnoses.

The second issue to be addressed is the computation required for performing diagnosis with these networks. In general, exact inference in Bayesian networks is NP-hard. However, we propose a specific architecture for the network that reduces the computational problem to one with polynomial complexity. Specifically, 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.

Given the conditional independence of the diagnoses, we can then 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**.

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

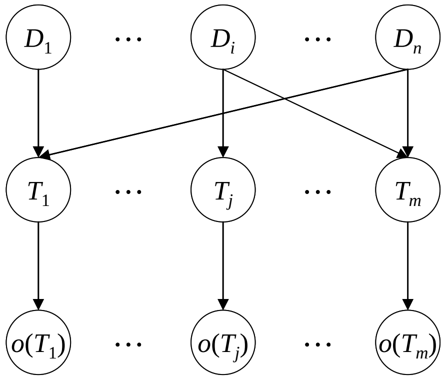

Figure 2–Diagnostic Bayesian network structure where test measurements are shown to depend only on diagnoses, and test observations (i.e., outcomes) depend on test measurements. Outcomes are conditionally independent of diagnoses *given the test measurements*.

Here, α is a normalizer over the set $\mathbf{D}$, equal to

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

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 [12]. In particular, we consider each diagnosis to be a separate classification. In this case, diagnosis reduces to assigning the class label corresponding to the maximum *a posteriori* probability. More formally,

$$h_{MAP} = \arg\max_{\omega \in \Omega} [\lambda_\omega \Pr(x \mid \omega) \Pr(\omega)]$$

where ω represents the "actual state" of the unit being tested and λ_ω is the loss associated with ω being the incorrect classification. The resulting classification is referred to as the *maximum a posteriori hypothesis*. In other words, the diagnosis yielding the highest posterior probability is proposed as the most probable fault.

Observe that $\Pr(o(T_j) \mid D_i) \in \{0, 1\}$ as described earlier, so the members of the sum are restricted only to those tests that observe D_i. Then we only need to consider $\Pr(D_i)$, 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

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

In other words, we provide the most probable diagnosis as a means of minimizing expected error (i.e., risk or loss) in the diagnostic process.

4. DBNs and Prognosis

The traditional approach to fault diagnosis assumes tests are applied at a specific point in time from which one can infer the condition of the system under test and make a diagnosis. The problem of prognosis, while essentially an extension of diagnosis, is complicated by the fact that time becomes a significant factor in the analysis. In fact, one can represent the prognosis problem as a time series prediction problem in which one attempts to infer a future state from some sequence of past states.

It turns out that the Bayesian approach to diagnosis can be generalized in a straightforward way to address prognosis as well. In the most basic case, consider the state of the system as if it can be represented at some time t as a single random variable s_t. Assume, further, that the state at time $t + 1$ depends only upon the system state at time t. Then we can represent the time series corresponding to the system state progression as a first-order Markov chain.

Missing in our model is the fact that we do not have direct knowledge of the underlying state of the system. Specifically, we perform tests to observe conditions of the system, from which we infer the system state. Consequently, the basic Markov chain is not sufficient for our purpose—we need to differentiate between observable random variables and "hidden" (or unobservable) random variables. This leads to the concept of a Hidden Markov Model (HMM) [13].

Formally, an HMM = $\langle N, M, A, B_j, \pi \rangle$, where N is the number of states in the model (denote the states as $S = \{s_1, \ldots, s_N\}$), M is the number of distinct observation symbols per state (denote the symbols as $V = \{v_1, \ldots, v_M\}$), A is the state transition probability distribution $A = \{a_{ij}\} = \Pr(q_t+1 = s_j \mid q_t = s_i)$, B_j is the observation probability distribution in state s_j, $B_j = b_j(k) = \Pr(v_k \text{ at } t \mid q_t = s_j)$, and π is the initial state distribution, $\pi = \{\pi_i\} = \Pr(q_0 = s_i)$ [13].

The Markov chain and the HMM can be formulated as special cases of a graphical model first formalized by T. Dean and K. Kanazawa called the "dynamic Bayesian network" [14]. DBNs have been studied further by Kevin Murphy who provided alternatives for representation, inference, and learning [15]. The purpose of a DBN is to model probability distributions over semi-infinite collections of random variables, Z_i, that progress according to some temporal model. Typically, the random variables are partitioned into three subsets indexed by time—$Z_t = (U_t, X_t, Y_t)$ where U_t is the set of inputs at time t, X_t is the set of hidden (i.e., unobservable) variables at time t, and Y_t is the

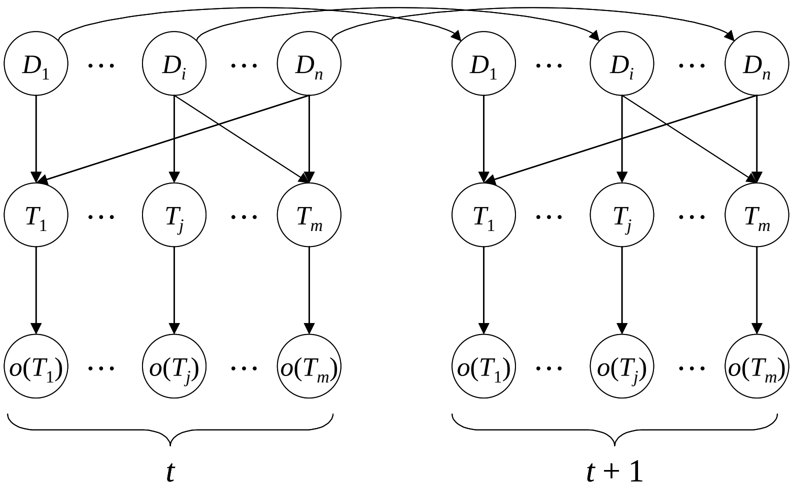

Figure 3–Dynamic Bayesian network for prognosis. The DBN is represented as a sequence of traditional diagnostic Bayesian networks where each diagnosis depends only upon itself in the previous time step. Test measurements are assumed to be conditionally independent of corresponding measurements in the previous time step given the diagnoses they detect. Such conditional independence holds for test outcomes as well.

set of outputs at time *t*. Then, given the set *Z*, a DBN is defined to be a pair $\langle B_1, B_\rightarrow \rangle$, where B_1 is a Bayesian network defining the prior distribution $\Pr(Z_1)$, and $B_\rightarrow$ is a "two-slice" temporal Bayesian network defining the distribution $\Pr(Z_t \mid Z_{t-1})$ such that

$$\Pr(Z_t \mid Z_{t-1}) = \prod_{i=1}^{N} \Pr(Z_t^i \mid \mathrm{Pa}(Z_t^i))$$

where Z_t^i is the i^{th} node at time *t*, which could be a component of any of the partitions, and $\mathrm{Pa}(Z_t^i)$ are the parents of Z_t^i in the network. Of interest is the fact that the parents of a node, $\mathrm{Pa}(Z_t^i)$, can either be from the same time slice or from the previous time slice (i.e., the resulting model is restricted to being a first-order Markov model). Of course, the general formulation of DBNs allows for higher-order models simply by expanding the allowable set of parents to previous time slices.

To put this definition in the context of prognosis, we can construct a DBN for prognosis by "chaining" successive BBNs together. Under the first-order Markov assumption, we only need to represent two slices of the DBN and then "unroll" as necessary in processing the model. For example, Figure 3 shows how to link the BBNs in sequence. Note that only the diagnoses are linked through time since they change state directly. Changes in observation state are derived from the underlying state changes in the system. This approach is distinct from the HMM that links observations together.

To perform inference with the DBN (and thereby predict future states), first, infer the current state (i.e., the state in the current time slice) from the test observations. Next, "unroll" the DBN to the desired number of time slices (assuming the state progressions occur in discrete time steps—DBNs can handle continuous time, but the computation is more complex). Then, propagate beliefs through time by observing that

$$\begin{aligned}\Pr(D_i^{t+1}) = {} & \Pr(D_i^{t+1} \mid D_i^t)\Pr(D_i^t) \\ & + \Pr(D_i^{t+1} \mid \neg D_i^t)\Pr(\neg D_i^t)\end{aligned} .$$

In fact, given the assumption that only diagnoses progress in state through time and that a diagnosis only depends upon itself in the previous time step, this part of the model reduces to a simple Markov chain, which can be either discrete time or continuous time.

Key to constructing the DBN is defining the temporal transition probabilities. In the simplest case, failure probabilities estimated from the failure rates can be used. When better information is available (e.g., based on historical data as is to be provided by the UID program), probabilities derived from this information can be used. The point is that the DBN is fully general and can be adapted to available knowledge about the system being analyzed. Theoretically, causal relationships between faults (i.e., a fault at time step *t* causes another fault to occur at time step *t* + 1) can be represented directly with the DBN as well (even though such models are rarely useful).

5. LEARNING BAYESIAN NETWORKS

The Bayesian network shown in Figure 2 presents a substantial improvement over past efforts at creating diagnostic networks; however, several issues remain. First, we often do not have direct access to measurement data or

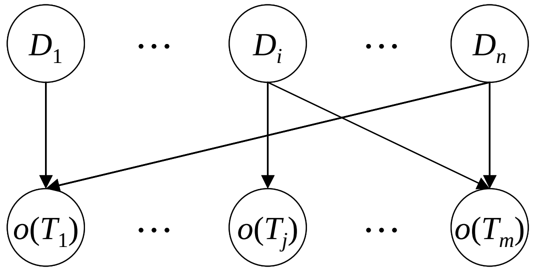

Figure 4–Simplified diagnostic Bayesian network where the test measurements are "abstracted" out of the model. This is done as a convenience since typically raw test measurements are not used in fault diagnosis.

the relationship between measurements and outcomes. Thus, we must map outcomes directly back to diagnoses. This can be handled with a "simple" bipartite Bayesian network such as the one shown in Figure 4. Second, having reduced the structure of the network to represent dependencies between outcomes and diagnosis (which were originally assumed to be conditionally independent given the measurement), we must determine appropriate probabilities for the conditional probability tables. We have lost the ability to derive these probabilities directly from measurement uncertainty because we no longer have the measured values represented. Third, even with these "simplified" networks, the conditional probability tables associated with each of the test results grows exponentially in the number of diagnosis (or faults) detected. These latter two issues are the focus of work reported in this paper.

To address both the network architecture and probability table issues, we will discuss methods of "learning" Bayesian networks from data. First, we will focus on a simple Bayesian model called the naïve Bayesian network. Then, given some of the deficiencies of the naïve Bayes model, we will consider approaches, again based on machine learning, to reduce the impacts of these deficiencies leading to our so-called "not-so-naïve" Bayesian networks. We hypothesize that, a) the naïve Bayesian network will provide "reasonable" performance in diagnosis, even with its strong assumptions and that b) the non-so-naïve Bayesian network will provide some level of improvement (perhaps significant) in diagnostic accuracy over the strictly naïve Bayesian network.

5.1 Naïve Bayesian Networks

The primary assumption for naïve Bayesian networks is that the evidence variables in the network (i.e., the tests) are conditionally independent of each other given the class (i.e., diagnosis). To start our discussion on the implications of this assumption, let us define our diagnostic networks as if they contain only one diagnosis variable with *n* possible values (corresponding to each of the diagnostic conclusions D_i). Thus, the structure of our networks reduces to the form shown in Figure 5. Note that the structure from Figure 4 can be retained if we construct so-called naïve Bayes "multi-nets" [16].

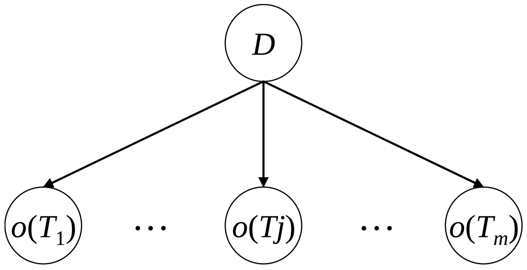

Figure 5–Naïve Bayes diagnostic network. Using this structure assumes all test outcomes depend upon a single diagnosis variable where the set of values for *D* corresponds to the set of possible diagnoses. The structure also enforces independence of the diagnoses so that modeling $\Pr(o(T_j) \mid d_1 \ldots d_m)$ becomes unnecessary.

Under the naïve Bayes model, we still consider the diagnosis problem as finding the class label (i.e., diagnosis) that maximizes the *a posteriori* probability of the specific class given the set of observations:

$$\begin{aligned} D &= \arg\max_{d_i \in \mathbf{D}} \Pr(d_i \mid o(T_1), \ldots, o(T_n)) \\ &= \arg\max_{d_i \in \mathbf{D}} \Pr(o(T_1), \ldots, o(T_n) \mid d_i) \Pr(d_i). \end{aligned}$$

But the problem remains that the joint distribution over the tests is exponential in the number of tests. The naïve Bayes assumption states that we can treat each of these observations as if they are conditionally independent given the diagnosis, and this leads to the classification rule:

$$D = \arg\max_{d_i \in \mathbf{D}} \Pr(d_i) \prod_{j=1}^{n} \Pr(o(T_j) \mid d_i).$$

Given a set of training data mapping test results (outcomes) to actual faults repaired, we can "learn" the naïve Bayes network by observing that $\Pr(d_i)$ is simply the frequency of occurrence of a particular fault in the data set and similarly, $\Pr(o(T_j) \mid d_i)$ is the frequency of test outcome $o(T_j)$ considering only the particular diagnosis d_i.

In considering the complexity of naïve Bayes networks, we must account for the complexity to learn the models as well as to use the models for diagnosis. The complexity of learning depends on deriving the probability estimates for $\Pr(d_i)$ and $\Pr(o(T_j) \mid d_i)$. Let n = the number of examples, $\delta = |D|$, and $\tau = |T|$. It is reasonable to assume $\delta < n$. Then the complexity for deriving $\Pr(d_i)$ for all $d_i \in D$ is $O(n + \delta) =$

$O(n)$, and the complexity for deriving $\Pr(o(T_j) \mid D_\mathrm{i})$ for all T_j and d_i is $O(\tau \times n + \tau \times \delta) = O(\tau \times n)$. Classification involves multiplying $\tau + 1$ probabilities for each diagnosis and maximizing, so the complexity of classification is $O(\tau \times \delta)$.

5.2 Tree-Augmented Naïve Bayesian Networks

While simple to construct, the naïve Bayes network imposes a strong independence assumption on the network structure and hence the diagnostic model. In diagnosis, it is rarely the case that test results are conditionally independent given the diagnosis. In fact, in many cases, we find that tests are highly dependent given the diagnosis they are intended to detect. One approach to handling such dependencies without enumerating the full joint distribution over the tests is to re-insert dependence relationships as indicated by the model. At least three methods for implementing this approach have been proposed in the literature—the semi-naïve Bayesian network (SNBN) [17], the tree-augmented Bayesian network (TAN) [16], and the Bayesian network augmented naïve Bayesian network (BAN) [18].

In the SNBN, a naïve Bayesian network is constructed to start, and then the conditional entropy is computed between pairs of test attributes in the network. This is compared to the entropy of the network as a whole. Combining a pair of nodes into a composite node (represented by the pair's joint probability distribution) changes the entropy of the network. Constructing the SNBN then consists of incrementally combining the pairs of nodes that decrease entropy the most until some threshold is reached. Note that this can lead to very large composite nodes, so alternatives have been suggested where the size of the composites are constrained to consist of no more than k simple nodes. This leads to so-called k-dependent semi-naïve Bayesian networks (k-SNBN) [19].

In the BAN, rather than attempt to construct sets of joint distributions, the observation that a general Bayesian network can be used to model such joint distributions is used to construct the network. In this case, a general Bayesian network is learned while ignoring the presence of the class node D. The Bayesian network is learned using a scoring metric such as the Bayes information criterion (BIC) or the minimum description length (MDL). After constructing the Bayesian network, the class node is reinserted and conditionally independent probabilities associated with that node.

In this paper, we consider a third approach as a "compromise" position that has been demonstrated to work well on other classification tasks—the tree-augmented Bayesian network. In the TAN, the process of determining the dependencies between test nodes is similar to the BAN in that a Bayesian network is constructed while ignoring the class label D. After constructing the network, the class label is then added back in with the conditionally independent probabilities derived for a naïve Bayes network. The difference is that it is assumed we can construct a separate Bayesian network among the attributes, and that network corresponds to a tree structure. The advantages of this assumption are that a) the tree structure can be learned very quickly, and b) inference with the resulting network is very fast because each node has at most two parents (one of which is the class label).

In general, the approach to constructing the TAN is as follows. First, we will construct a complete graph among the test nodes in the network. This graph will be undirected to start. Second, we will associate a weight to each edge in the network. The specific weight chosen will be discussed below, but we will choose a weight such that a higher value is more desirable. Third, we will find the "maximum-weight" spanning tree of the graph using these assigned weights to drive the search. Fourth, we will pick one of the attributes at random and designate it to be the "root" of the tree. Fifth, we will derive a rooted tree structure by traversing the spanning tree from the root node and making all traversed edges directed, pointing away from the root. Sixth, we will reinsert the class node into the network and make it a parent of every test node. Finally, having identified dependencies between a child node $o(T_i)$ and its parent node $o(T_j)$, we will derive the conditional probabilities $\Pr(o(T_i) \mid o(T_i), D)$ from the training data.

Key to constructing the TAN is the derivation of the weights associated with a pair of test nodes. The approach suggested by Friedman, Geiger, and Goldszmidt [16] is to use the conditional mutual information of the nodes. Specifically, we consider the mutual information of a pair of test nodes given the diagnosis node. This can be computed for every pair of test nodes as follows:

$$I(o(T_i), o(T_j) \mid D) = \sum_{o(T_i), o(T_j), D} \Pr(o(T_i), o(T_j), D) \log \frac{\Pr(o(T_i), o(T_j) \mid D)}{\Pr(o(T_i) \mid D) \Pr(o(T_j) \mid D)}.$$

Similar to the approach used to build SNBNs, the conditional mutual information provides a measure of how much one can reduce the entropy (or uncertainty) of the network by adding the indicated dependence relationship. In fact, it can be shown easily that if $o(T_i)$ and $o(T_j)$ are conditionally independent given D, $I(o(T_i),o(T_j)|D) = 0$ and that the greater the dependence relationship, the higher the value of I.

The complexity of learning TAN's has the complexity of learning NB as its lower bound. The difference in complexity comes from calculating the conditional mutual information between all of the tests and then deriving the spanning tree. Friedman *et al.* showed that the complexity of calculating $I(o(T_i),o(T_j)|D)$ is $O(\tau^2 \times n)$, and the complexity of finding the maximum weight spanning tree is $O(\tau^2 \lg \tau)$. Thus the overall complexity for learning a TAN

Table 1. Diagnostic data sets from the STANDARD Missile and the UCI Machine Learning Repository. All test sets corresponding to “diagnostic” problems. Nominal tests are analogous to test outcomes $o(T_j)$ while continuous tests are analogous to the test measurements T_j. The “Missing Values” column indicates if data is missing for one or more of the attributes in the data set. The “Ljubljana” data sets correspond to data available for academic use only.

Dataset	**Instances**	**Diagnoses**	**Tests (nominal)**	**Tests (continuous)**	**Missing Values?**
STANDARD Missile	217	12	321	0	Yes
Breast Cancer (Ljubljana)	286	2	9	0	Yes
Breast Cancer (Wisconsin)	699	2	9	0	Yes
Dermatology	366	6	33	1	Yes
Heart spectrograph	267	2	22	0	No
Hepatitis	155	2	19	0	Yes
Lymphography (Ljubljana)	148	4	18	0	No
Postoperative Patient	90	3	8	0	Yes
Primary Tumor (Ljubljana)	339	20	17	0	Yes
Soybean	683	19	35	0	Yes

is dominated by calculating $I(o(T_i),o(T_j)|D)$. Classification with TAN is still $O(\tau \times \delta)$ since all we have done is double the number of probabilities stored.

5.3 Bayesian Multinets

Given the basic formulation of the diagnostic Bayesian network as represented in Figure 4, we would like to restore the naïve Bayes network and the TAN network to reflect this structure. To accomplish this, we will also consider the multinet classifier where $n = |D|$ separate classifiers will be constructed, treating each potential value of D as a separate binary classification problem [16]. This has the advantage of enabling us to combine the results of the classifier either by selecting the class with the maximum posterior probability or to treat the classifiers in a combined sense to consider the possibility of multiple faults. For comparison purposes, however, we will evaluate the multinets according to the first option, which enforces the single fault assumption. We note that the complexity does not change for learning multinets when compared to their single-network counterparts.

6. EXPERIMENTAL APPROACH

Given the goals of the DoD UID program, our objective was to evaluate various Bayesian methods for diagnosis and prognosis. Because the results in this paper focus on the applicability of “not-so-naïve” approaches to constructing Bayesian classifiers, we concentrated on the accuracy of these algorithms for diagnosis. We claim that, assuming the availability of test data captured over time, we can construct similar prognostic Bayesian networks simply by deriving $\Pr(D_t \mid D_{t-1})$ from the training data like any other conditional dependence relationship in the data. We are in the process of testing this hypothesis and plan to provide the results in a future publication.

Since the purpose of this research is to explore the applicability of Bayesian models to fault diagnosis and prognosis under the UID program, we sought data from the US Navy to support this research but also used several data sets from the UCI Machine Learning Repository [20]. The specific databases are described in Table 1. Each of the UCI data sets, while medical in nature, reflects data collected on real diagnostic problems. The US Navy provided a subset of data on the STANDARD Missile corresponding to test and maintenance data collected on real missile systems. Below, we provide initial results with this data and discuss some of the difficulties associated with preparing the data for analysis.

Each of the algorithms was run on each data set using a hold-out method where 66% of the data was used for training and 34% was used for testing. Thirty replications of the hold-out experiments were run, randomizing the data before each run, to calculate confidence intervals at the 95% level. These were used to determine if there were any significant differences in performance based on a paired *t*-test. When only two classes existed in the data, the experiments included naïve Bayes (NB-S) and TAN (TAN-S). When the data had more than two classes, four algorithms were tested—single naïve Bayes (NB-S), naïve Bayes multi-nets (NB-M), single TAN (TAN-S), and TAN multi-nets (TAN-M). The Waikato Environment for Knowledge Analysis (WEKA) was used to test the algorithms [21]. WEKA’s multi-class classifier was used to implement both NB-M and TAN-M.

By default, WEKA fills in missing data based on the mode of the attribute values. We were concerned this would skew the results, so all of the datasets were pre-processed to estimate missing values using the probability distributions for each attribute in the dataset. Since the lack of test data can itself, provide information, missing attribute values for the STANDARD Missile were handled differently.

Table 2. Experimental results (accuracy with 95% confidence interval). NB-S corresponds to the simple naïve Bayes network, TAN-S to the simple TAN network, NB-M to the naïve Bayes multinet, and TAN-M to the TAN multi-net. All experiments were run with 30 hold-out trials to determine statistical significance. Entries indicated by "—" correspond to cases not run because the diagnosis problem had only two classes.

Dataset	**NB-S**	**TAN-S**	**NB-M**	**TAN-M**
STANDARD Missile	33.99±3.61	43.27±3.74	34.22±3.53	43.14±3.65
Breast Cancer (Ljubljana)	72.61±4.02	68.42±3.19	—	—
Breast Cancer (Wisconsin)	97.16±0.74	95.24±1.32	—	—
Dermatology	97.80±1.27	93.02±2.27	97.72±1.39	97.05±1.27
Heart spectrograph	78.00±3.84	80.42±4.10	—	—
Hepatitis	85.06±4.11	84.49±4.64	—	—
Lymphography (Ljubljana)	84.16±3.59	80.63±4.77	84.49±3.52	82.42±3.97
Postoperative Patient	66.92±4.71	63.52±6.57	66.26±4.54	63.53±6.31
Primary Tumor (Ljubljana)	52.02±3.36	48.73±3.09	53.12±3.22	50.62±3.01
Soybean	92.12±1.50	93.62±1.59	91.80±1.39	93.06±1.42

Table 3. Pair-wise comparison of algorithms. Values in a cell indicate the number of times the algorithm indicated by the column heading outperforms the algorithm indicated by the row heading. The value in parentheses indicates the number of times the difference recorded is significant at the 95% level.

	NB-S	TAN-S	NB-M	TAN-M
NB-S	—	3 (1)	3	2 (1)
TAN-S	7 (2)	—	4 (1)	4 (1)
NB-M	3	2 (1)	—	2 (1)
TAB-M	4	2	4	—

Table 4. Identification of pairs of algorithms yielding significant differences in performance at the 95% level and the data sets that yielded the performance difference.

Dataset	**Better**	**Worse**
STANDARD Missile	TAN-S	NB-S, NB-M
	TAN-M	NB-S, NB-M
Dermatology	NB-S	TAN-S
	NB-M	TAN-S
	TAN-M	TAN-S
Wisconsin Breast Cancer	NB-S	TAN-S

Specifically, all attributes corresponding to test (or parameter) codes either passed, failed, or were not tested. Therefore, all "missing" attributes were, assigned a "not-tested" value. Given the structure of the WEKA tool set and the pre-processing performed on the data, no user-defined parameters were required to be set.

One issue common to Bayesian methods is how to handle cases where, for a particular class, no data exists for the various attribute values. The usual approach to handling this problem is to apply the so-called *m*-estimate (based on the concept of Dirichlet smoothing, which is a generalization of Laplace smoothing) to approximate the probabilities. In this case, probabilities are approximated as

$$\Pr(o(T_i) \mid D_j) = \frac{n_c + mp}{n + m}$$

where n_c is the number of instances in the data pairing particular values for $o(T_i)$ and D_j, n is the total number of instances in the data corresponding to diagnosis D_j, p is a prior estimate for the probability, and m is the number of "virtual" examples in the data.

The usual approach to applying the *m*-estimate is to estimate the probabilities during training on the training set. The current implementation of WEKA does not provide the *m*-estimate but estimates missing data using the mean or the mode of the data over the entire data set. We decided to use the *m*-estimate over the entire data set as a pre-processing step to define a probability distribution over the attribute values and then select values according to that distribution. We recognize that this approach violates the assumption that test data and training data are generated independently but used the approach as a matter of convenience rather than attempting to rewrite the WEKA code. This issue will be resolved in future experiments. For our preprocessor, we used a simplified *m*-estimate equal to $1 + n_c / n$.

7. Results

The results of the experiments are shown in Table 2. Table 3 then shows a pair-wise comparison of the algorithms across all data sets. Each cell in this table shows the number of times the algorithm listed in the column heading beats the algorithm listed in the row heading. For example, this table shows that NB-S beats TAN-S seven out of ten times.

Of particular interest in these results is the fact that NB (both varieties) beats TAN more often than not. Of course, the results also show that the differences are statistically significant at the 95% level on only 11 out of the 40 comparisons (indicated by the number in parentheses in the appropriate cells in Table 3). If we adjust the significant test to the 90% level (a relatively low confidence level), only one additional significant difference arises. Table 4 lists the pairs of algorithms with significant differences in performance and the data sets where the differences were observed. It is interesting that all significant differences were observed on the dermatology, Wisconsin breast cancer, and STANDARD Missile data sets only.

Since our focus is on the STANDARD Missile, it is also interesting to note that the differences in performance are closer to what was originally expected. Specifically, we note that, while the individual accuracies are low (34–43%), the differences in performance favor the "not-so-naïve" networks. This provides a clear example of the need to model interdependencies of test results to improve overall accuracy of diagnosis.

8. DISCUSSION

We first note that our initial hypothesis is confirmed in that both versions of the networks (NB and TAN) performed very well. In fact, their performances on the UCI data sets were comparable to the best results reported in the literature. Our disappointment comes in that a) few of the differences in performance were significant at the 95% confidence level, and b) the raw differences observed actually seemed to favor the naïve Bayes versions. This was contrary to our expectations when beginning the experiment. The one exception to this observation—performance on STANDARD Missile data—was encouraging. Here we see a "real-world" data set for a complex system, and the advantages of the "not-so-naïve" model become evident.

So why were these results observed? We believe the answer is based on how the additional dependencies are included in the TAN network. Specifically, TAN inserts dependencies according to the maximum-weight spanning tree algorithm. Suppose a high-weight edge is being considered for addition, but addition of that edge would create a cycle. Then, even though the conditional mutual information calculation has determined there is a dependency between the associated attributes, this dependency will not be included in the Bayesian network. Furthermore, the algorithm could lead to the addition of several low-dependency relations in the network just because they do not create cycles in the spanning tree.

As future work, we plan to test this hypothesis by considering alternative approaches to creating the augmentations. Specifically, we will explore general Bayesian network learning, *k*-dependent Bayesian network learning, and "forest-augmented" naïve Bayes (FAN) learning. We are also exploring an approach to deriving naïve Bayes multi-nets where the set of attributes contained in each naïve Bayes network will be derived from the set of non-redundant attributes for the given class label. This approach is based on prior work in logical dependency modeling [22] where subsumption of test-to-diagnosis dependency sets is used to determine inter-test dependency.

```
      LDN64267AMIS124647|
     8/11/1995 7:54:00 AM|
    GS|5959014-009|861525|
    11/17/1995 11:50:00 PM|
SSLO IS SWEEPING. R&R PLT 1.|
         10300|0|1|
       21030|0|-0.53|
         21010|0|1|
         21430|0|1|
         21440|0|1|
         21450|0|1|
         21460|0|1|
        18010|0|3.4|
        21040|1|-1|
            ...
```

Figure 6–Sample STANDARD Missile data record containing the UID, the repair date, the test date, the diagnostic text, and a sequence of parameter codes with measurements and pass/fail outcomes.

9. ANALYSIS OF STANDARD MISSILE DATA

One of the goals of this research was to evaluate various NBN architectures on a real diagnostic data set. The US Navy provided warranty data from the STANDARD Missile for this purpose. Each record of the data file consists of the UID for the unit under test[1], the repair date, an indication that the section of the missile[2], the part number, the serial number, the test date, diagnostic text, and a sequence of "parameter codes." The parameter codes correspond to the tests performed leading to the associated diagnosis, and the data provides a triple of values for each parameter:

$$\langle \textit{parameter code, pass/fail code, raw measurement} \rangle$$

An excerpt of a record from the STANDARD Missile Guidance Set is provided in Figure 6. (The actual example contains 155 parameter codes, and this is typical of most of the examples in the data set.) In this example, all fields are delimited by the "|" character.

[1] Actually, this is a "virtual" UID that was developed as part of a Navy maintenance tracking database prior to initiation of the UID program.

[2] In this case, GS indicates the guidance section.

When examining the data in Figure 6, note that the diagnosis is identified as a remove and replace action (indicated by “R&R”) on plate 1 resulting from sweeping on the solid-state local oscillator (SSLO). One issue that needed to be resolved was correlating and reducing the diagnoses to a set appropriate for the level of maintenance. For example, one level could be defined corresponding to the plates within the guidance set, in which case, this example would diagnosis to Plate 1. On the other hand, the diagnosis could move down a level of indenture to the SSLO. If required, we could even isolate to the specific cause (SSLO sweeping).

Unfortunately, all of the examples provided identify the diagnosis through the diagnosis text field. This field is entered manually by a test technician and follows no specific standard. Thus the issue of determining the set of diagnoses is complicated by the fact the data may contain identical diagnoses that are worded slightly differently. For this data set, a domain expert at NSWC Corona examined each of the instances and categorized each instance as one of 12 diagnostic conclusions.

The next step required in preparing the data for processing is converting the data records into a form compatible for processing by a feature-based classifier such as NB or TAN. For training purposes, all records must identify values for each of the available attributes in a composite feature vector over the data set. Thus, the set of parameter codes was compiled and compared against the data to create a matrix correlating tests and diagnoses. The data set provided contained 217 records (i.e., examples), 12 distinct diagnoses (after a domain expert examined the specific examples), and 321 parameters.

We ran all four algorithms on the data provided and had apparently disappointing results. As shown in Table 2, we had accuracies ranging from 34% to 43%, which is significantly lower than the accuracies obtained from the other data sets. Note, however, that these accuracies are substantially higher than “chance” (~8%), indicating that the networks were able to discern valuable diagnostic information in the data. This poor performance can be explained by observing that several of the examples only had single tests indicting the corresponding failures. Even applying the *m*-estimate will not compensate for such tight coupling of test results to diagnoses.

Manual analysis of large data sets, such as was performed for the experiments described here, is not feasible in general. Fundamentally, the following issues need to be resolved for automating the processing of the STANDARD Missile data:

(1) Descriptions of the diagnostic classes need to be normalized to eliminate the issue of identical diagnoses being considered as unique, simply because of slight wording variations.

(2) Diagnostic classes either need to be reduced or more data obtained to provide a reasonable sample from which prior probabilities Pr(D_i) can be derived.

(3) The feature set (i.e., set of parameter codes) need to be reduced to a set of essential features. This accomplishes two things: First it reduces the size of the exemplar matrix to a more manageable size, and second it reduces the sparseness problem within the data set itself.

10. Future Work

The data collected during maintenance of the STANDARD Missile illustrates the type of data that is well suited to meeting UID-based diagnostic requirements with Bayesian analysis. Data is collected by unit and identified by the UID of that unit. This permits data to be collected specific to the unit tracked and permits construction of appropriate diagnostic models for that unit. The data also provides date-time groups for test and repair that permit correlation of maintenance over an extended time frame. Ultimately, this will facilitate creating unit-specific prognostic models as well.

Key to fulfilling the vision of UID in the context of system diagnosis and prognosis is cleaning and preparing the data for subsequent analysis. Since the UID program is relatively new for the STANDARD Missile, the data does not yet exist in the appropriate form needed to facilitate automated Bayesian analysis. Therefore, the near term work for advancing this technology will be on applying data cleaning techniques as a pre-processing step.

In addition to addressing data cleaning/preparation, future research will focus on analyzing the reasons for the behavior seen in comparing naïve Bayes classification and not-so-naïve Bayes classification. One approach under consideration is simultaneously a) to relax the spanning tree requirement for adding links, and b) to apply concepts from forest augmented naïve Bayes networks (FAN) to prevent insertion of low-dependency links just because they are required to create a full tree. A second approach under consideration is applying techniques from model-based reasoning for deriving test-to-test dependencies from the D-matrix as a starting point for defining the architecture of the not-so-naïve Bayesian network.

11. Conclusion

In this paper, we provided a proposal for considering an alternative form for Bayesian reasoning in a system diagnostics environment. We compared four different algorithms that have been published previously in the machine learning community to diagnostic data sets from the UCI Machine Learning Repository and considered requirements for applying these algorithms to a real-world diagnostic problem. We also applied algorithms to real-world data collected on the US Navy's STANDARD Missile. Results from the initial experiments were somewhat surprising in that they demonstrated naïve Bayes classification outperforming not-so-naïve Bayes classification (contrary to our hypothesis). We believe this can be explained by the preference bias inherent in the TAN algorithm and plan to explore alternative methods for augmenting the naïve Bayes network. Finally, we encountered several real-world problems with the STANDARD Missile data and suggested steps required to prepare the data for subsequent analysis. Even so, we were able to complete some preliminary experiments. The results on the STANDARD Missile data differed from the UCI data sets; however, in that NBNs were shown to be of significant benefit on this data.

Acknowledgments

The authors would like to thank the reviewers of this paper whose comments improved the overall presentation. In addition, we would like to thank the US Navy for furnishing the maintenance data on the STANDARD missile test case as well as for supporting portions of this research. We especially thank Bill Wolf for manually categorizing the examples in the missile data set.

References

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

[2] Government Accounting Office Report NSAID-92-258, "Operation Desert Storm, Lack of Accountability Over Materiel During Redeployment", September 1992.

[3] DoD UID website http://www.acq.osd.mil/dpap/UID/

[4] DoD RFID website http://www.acq.osd.mil/log/rfid/

[5] Under Secretary of Defense (Acquisition, Technology and Logistics) (OUSD(AT&L)) Memorandum, Update for Policy for Unique Identification (UID) of Tangible Items, September 3, 2004.

[6] Under Secretary of Defense (Acquisition, Technology and Logistics) (OUSD(AT&L)) Memorandum, Radio Frequency Identification (UID) Policy, July 30, 2004.

[7] DoD UID Data Elements and Data Formats website http://www.acq.osd.mil/dpap/UID/DataElements.htm

[8] Under Secretary of Defense (Acquisition, Technology and Logistics) (OUSD(AT&L)) Guidelines for the Virtual Unique Item Identifier (UII) Version 1.0, December 29, 2004.

[9] Department of Defense Guide to Uniquely Identifying Items Assuring Valuation, Accountability and Control of Government Property, Version 1.5, June 7, 2005.

[10] John W. Sheppard, "Accounting for False Indication in a Bayesian Diagnostics Framework," *AUTOTESTCON 2003 Conference Record*, IEEE Press, September 2003.

[11] John W. Sheppard and Mark Kaufman, "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.

[12] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, New York: John Wiley & Sons, 2001.

[13] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286, 1989.

[14] T. Dean and K. Kanazawa, "A Model for Reasoning About Persistence and Causation," *Artificial Intelligence*, 93(1–2):1–27, 1989.

[15] Kevin Murphy, *Dynamic Bayesian Networks: Representation, Inference, and Learning*, PhD Dissertation, Department of Computer Science, University of California, Berkeley, 2002.

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

[17] Ivan Kononenko, "Semi-naïve Bayesian Classifier," In Y. Kodratoff (ed.), *Proceedings of the Sixth European Working Session on Learning*, Berlin: Springer-Verlag, 1991, pp. 206–219.

[18] K. Ezawa, M. Singh, and S. Norton, "Learning Goal-Oriented Bayesian Networks for Telecommunications Risk Management," *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996, pp. 139–147.

[19] M. Sahami, "Learning Limited Dependence Bayesian Classifiers," Proceedings *of the Second International*

Conference on Knowledge Discovery and Data Mining, 1996, pp. 335–338.

[20] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, "UCI Repository of Machine Learning Database," http://www.ics.uci.edu/~mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[21] Ian H. Witten and Eibe Frank, *Data Mining: Practical* Machine *Learning Tools and Techniques*, 2nd edition, San Francisco: Morgan Kaufman, 2005.

[22] William R. Simpson and John W. Sheppard, *System Test and Diagnosis*, Norwell, MA: Kluwer Academic Publishers, 1994.

BIOGRAPHIES

John W. Sheppard is an Assistant Research Professor in the Department of Computer Science, Johns Hopkins University. Dr. Sheppard started his career as a research computer scientist for ARINC and attained the rank of Fellow. His research interests include algorithms for diagnostic and prognostic reasoning, machine learning and data mining in temporal systems, and reinforcement learning. Dr. Sheppard holds a BS in computer science from Southern Methodist University and an MS and PhD in computer science from Johns Hopkins University. He has over 100 publications in artificial intelligence and diagnostics, including an authored book and an edited book. He is a senior member of the IEEE and currently serves as Vice Chair of the IEEE Standards Coordinating Committee 20 (SCC20) on Test and Diagnosis for Electronic Systems, Secretary and Past Chair of the Diagnostic and Maintenance Control subcommittee of SCC20, and Official Liaison of the Computer Society Standards Activities Board to SCC20.

Stephyn G. W. Butcher is currently pursuing his MS in computer science at the Whiting School of Engineering, The Johns Hopkins University. He has served as a Lecturer in economics and Grader in computer science. He received his BA in economics from the California State University, Sacramento and his MA in economics from The American University, Washington, DC. His research interests are mainly in machine learning and include Bayesian networks and evolutionary computation with emphases on problems in classification and differential games. He plans to start work on a PhD in computer science in 2006.

Mark A. Kaufman received the BSEE degree from California State Polytechnic University, Pomona, and the MS degree in scientific and engineering management from West Coast University, Los Angeles, CA. He is a testability and diagnostics engineer with the Naval Surface Warfare Center, Corona Division, Corona, CA. He has been working in the metrology and test assessment areas for the Navy for over 20 years. He is a senior member of the IEEE and has held several offices within the IEEE including being the current Co-Chair of the Diagnostic and Maintenance Control subcommittee of IEEE Standards Coordinating Committee 20 on Test and Diagnosis for Electronic Systems, past Vice Chair of SCC20, and Vice Chair of the AUTOTESTCON conference. In addition, he has served as President and Chairman of the Board of the Measurement Science Conference.

Craig MacDougall works at the Naval Surface Warfare Center Corona Division, as a Missile Analyst for the STANDARD Missile. He is the project lead for implementing UID into the STANDARD Missile program. Craig has earned an M.S. in physics from the University of California Riverside in solid state physics, as well as an M.S. in systems engineering from the Naval Post Graduate School, where he focused on coordinated weapon management in Net-Centric-Warfare. His research interests are primarily in the fields of material sciences, thermo-dynamics and real-time signal processing.