Distributional Smoothing in Bayesian Fault Diagnosis

Stephyn G. W. Butcher, *Member, IEEE*, and John W. Sheppard, *Fellow, IEEE*

***Abstract*—Previously, we demonstrated the potential value of constructing asset-specific models for fault diagnosis. We also examined the effects of using split probabilities, where prior probabilities come from asset-specific statistics and likelihoods from fleet-wide statistics. In this paper, we build upon that work to examine the efficacy of smoothing probability distributions between asset-specific and fleet-wide distributions to further improve diagnostic accuracy. In the current experiments, we also add environmental differentiation to asset differentiation under the assumption that data are acquired in the context of online health monitoring. We hypothesize that the overall diagnostic accuracy will be increased with the smoothing approach relative to a fleet-wide model or a set of asset-specific models. The hypothesis is largely supported by the results. Future work will concentrate on improving the smoothing mechanism and in the context of small data sets.**

***Index Terms*—Bayesian classifier, diagnosis (fault), machine learning, smoothing.**

I. Introduction

RECENT results exploring the merits of fleet-wide versus asset-specific Bayesian diagnostic models suggest that circumstances can exist where using fleet-wide data in asset-specific models can yield significant improvements in overall diagnostic accuracy. These circumstances largely hinge on data heterogeneity, quantity, noisiness, and their effects on the estimates of the models' probability distributions [1], [2].

This paper reports on our most recent results applying distributional smoothing to probability estimates in Bayesian diagnostic models seeking to combine fleet-wide and asset-specific coverage. Other fields facing similar circumstances when using Bayesian approaches, e.g., natural language processing (NLP), apply smoothing to estimates of probability distributions. While most of these smoothing methods are not directly applicable to Bayesian diagnostic models, we present an alternative approach to distributional smoothing that is directly applicable to them. We have found that our models using smoothed probability estimates can be more accurate over a wider variety of data quality and quantity than any of our other models.

Manuscript received October 31, 2007; revised June 2, 2008. First published August 15, 2008; current version published January 5, 2009. This work was supported in part by contracts from the U.S. Army and the U.S. Navy. This paper was presented in part at the IEEE AUTOTESTCON, Baltimore, MD, September 2007. The Associate Editor coordinating the review process for this paper was Dr. Tadeusz Dobrowiecki.

The authors are with the Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: sbutche2@jhu.edu; jsheppa2@jhu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2008.928874

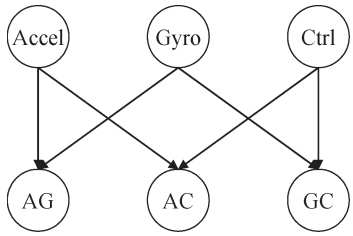


Fig. 1. Simple SAS Bayesian Network.

The outline of this paper is as follows. Sections II and III will describe the research problem and motivation as well as an approach to address the problem. Section IV discusses related work. In Section V, we explore the experimental design. Sections VI and VII will present and discuss the experimental results (including future work), respectively. We conclude in Section VIII.

II. Research Problem

Our research centers on learning Bayesian diagnostic models from test and maintenance data for an entire fleet. As an example of a diagnostic problem that uses a Bayesian network, suppose that we are considering the built-in test from the stability augmentation system (SAS) of a helicopter. SASs provide stability control for the three axes of the aircraft, namely, roll, pitch, and yaw. Without loss of generality, we will consider just the roll axis. In evaluating the performance of roll stability control in SAS, we might consider the health of at least three components, i.e., the roll control unit, the roll gyro, and an accelerometer. For our example, we assume that if the expected output of the control unit agrees with the actual derived roll outputs from the accelerometer and roll gyro, then the system is functioning properly. On the other hand, if any two of these three elements disagree, a fault exists in one of the two units involved in the disagreement. This scenario can be represented with the Bayesian network shown in Fig. 1.

To interpret the elements of this network, Accel, Gyro, and Ctrl correspond to the diagnoses of whether the accelerometer, gyro, or control unit are faulty, respectively. AG represents the observation associated with comparing the accelerometer output with the gyro output. AC compares the accelerometer output with the control output, and GC compares the gyro output with the control output.

Using this network, suppose we indicate that AC and AG both fail, but GC passes. Logically, we would expect Accel to be faulty. Given a set of probability distributions for each of the nodes in the network, we might find posterior probabilities

of {Accel: 0.539; Ctrl: 0.069, Gyro: 0.230; NF: 0.000}. Thus, we would conclude from the tests that Accel is the most likely to have failed.

The above-described Bayesian networks are either constructed from domain knowledge or learned from actual test and maintenance data. In the latter case, one would accumulate data recorded over the current life of the system and derive the probability distributions from that data. Typically, such a data-driven approach uses all of the data available to construct a model for a particular system. That is, they use data about the entire fleet of helicopters and construct fleet-wide models. As a practical matter, however, such data may be sufficiently heterogeneous that relevant diagnostic information is lost through aggregation. For example, specific helicopters may be from different production runs, they may have been exposed to different usage profiles, or they may have used different replacement parts. Similarly, the test equipment may have been operated in different environments, or the test data may have been obtained from online health monitors that themselves may be affected by different usage profiles.

All of these circumstances may conspire to produce training data that contain a degree of inconsistency when aggregated. To the degree that these heterogeneities exist, a model learned from aggregated data will be less accurate, and such a decrease in accuracy is predicted by machine learning theory [3]. Because a Bayesian diagnostic model is a type of classifier, the more closely the distribution of the training data matches the distribution of the target population, the more accurate the classifier will be [4]. In diagnostic terms, the more the maintenance and test data used to build the model reflect failure rates and test/diagnosis relationships the model will actually encounter when used in the field, the more accurate the diagnoses will be.

Thus, the alternative—at the other extreme—is to build a model for each individual asset (a car, plane, or GPS unit with a specific serial number) under the assumption that each asset is *sui generis* rather than one of an entire fleet of such assets. We realize that this assumption is equally unrealistic and that is largely what motivates the research reported here.

We used the naive Bayesian classifier for our learned diagnostic model in the reported experiments [5]. We decided on the naive Bayes classifier because of its robustness, low computation complexity, and ease in learning. In addition, using such a simple model permits us to focus our attention on the affects of the smoothing approach without commingling with learning an appropriate model structure. The naive Bayesian classifier is represented by

$$D = \arg\max_{D_i \in \mathbf{D}} P(D_i) \prod_{j=1}^{n} P\left(o(T_j)|D_i\right) \tag{1}$$

where D_i is some diagnosis, $P(D_i)$ is the prior probability estimate of a particular diagnosis in the data set, and $P(o(T_j)|D_i)$ is the frequency of some discrete test outcome $o(T_j)$, e.g., Pass or Fail, for some test T_i, considering only the particular diagnosis D_i or the likelihood.[1] Thus, the possible inconsistencies created by aggregation show up in the estimates of the priors $P(D_i)$ and likelihoods $P(o(T_j)|D_i)$.

Because of the possibility that some likelihood estimates may be zero because of missing data, likelihood estimates are often calculated using the m-estimate as [3]

$$P\left(o(T_i)|D_j\right) = \frac{n_c + mp}{n + m} \tag{2}$$

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. This prevents the equation for the naive Bayesian classifier from degenerating if any likelihood estimate is zero.

Note that the m-estimate modifies the likelihood estimate by adding in some fraction of a probability mass p. That fraction is determined by some number of virtual examples m. In many cases, p is simply chosen to be a small innocuous value that is sufficient to prevent the formula from zeroing out, and m is often set to 1. However, this need not be the case. If there is knowledge of p, or if p can be learned, then using that p should yield a more accurate estimate.

Based on the above observation on heterogeneous data, our original experiments investigated whether a set of diagnostic models, each built with asset-specific data, would have a higher overall accuracy than a single diagnostic model built with aggregate data for the entire fleet. Our experiments used different quantities of synthesized data reflecting assets with different failure rates and varying levels of measurement noise[2] so that we could control the presence of heterogeneity. Our results showed that a set of asset-specific models could be more accurate than a single fleet-wide model but not always.

In our original experiments, we modeled asset differences by using various failure rates for their respective components. Usually, failure rates are used to estimate $P(D_i)$ in (1). The current research adds the ability to capture usage patterns or environmental effects. By including usage patterns or environmental effects, we are able to capture different relationships between test outcomes and diagnoses within the data. Even with this change, the initial results were the same; a set of asset-specific models could be more accurate than a single fleet-wide model.

Although the results were consistent, the fact that some of the asset-specific models were less accurate than the fleet-wide model was problematic. Specifically, for a given quantity of data N, as noise increased, the accuracy of the asset-specific models increased relative to the fleet-wide model. We attribute this to better estimates of the prior probabilities in the asset-specific models. Additionally, for a given level of noise in the data, the larger the data set size, the more accurate the asset-specific models were. We attribute this to better estimates of different likelihoods for the asset-specific models.

While the patterns in these results are interesting, the mixed results leave open the question of knowing when to use the fleet-wide model and when to use the asset-specific model. However,

[1]For a more in-depth discussion of Bayesian approaches to diagnostics, see some of our previous papers [1], [2], [8].

[2]Details of the type of noise used are discussed in Section V.

the results did point to an approach to achieve our goal. In the presence of noise, other things being equal, we need to use as much data as possible to average out that noise. This supports aggregating the data. However, in the presence of heterogeneity, other things being equal, we need to use as specific data as possible.

III. Distributional Smoothing

Based on these observations, we hypothesize that a new model that uses smoothed distributional estimates for the likelihoods in the naive Bayesian classifier will be more accurate than either model alone. The rationale for our smoothing approach is based on the analysis of our prior experiments [1], [2]. In those experiments, we observed that the accuracy of the asset-specific versus fleet-wide classifier heavily depended on the amount of noise in the data and the sample size, and this dependence was nonlinear. We also observed a strong dependence on the distribution of the priors from the asset-specific data, regardless of noise and sample size. Therefore, we decided to focus on smoothing the likelihoods alone.

The starting point for distributional smoothing is using (2) for the m-estimate. When estimating the asset-specific likelihood, we use asset-specific data to calculate n_c and n and the *fleet-wide likelihood* as the value for p. When we estimate the fleet-wide likelihood, we use fleet-wide data to calculate n_c and n and a small value for p. We use a formula for m consistent with the following form when determining the asset-specific likelihood:

$$m = \frac{k}{f\left(o(T_i), D_j\right) g(n)} + 1 \tag{3}$$

where k is a user-defined parameter to control how much weight goes toward the asset-specific estimate versus the fleet-wide estimate, $f(o(T_i), D_j)$ is a function that relates the asset-specific and aggregate distributions for the (T_i, D_j) pairs, and $g(n)$ is a function of the data set size for a particular diagnosis D_j. The resulting function will cause m to decrease as the noise and amount of data increases.

This formula represents a generalization of the result provided in [8]. Consequently, several options now exist for $f(o(T_i), D_j)$. If we observe that we are dealing with two different probability distributions, where the asset-specific distribution tends to be the "preferred," we can apply a function of the divergence between the distributions for $P(o(T_i)|D_j)$, such as the Kullback–Leibler divergence

$$D_{\mathrm{KL}}(P_{\mathrm{asset}} \| P_{\mathrm{agg}}) = \sum_i P_{\mathrm{asset}}(o(T_i)|D_j) \lg \frac{P_{\mathrm{asset}}\left(o(T_i)|D_j\right)}{P_{\mathrm{agg}}\left(o(T_i)|D_j\right)} \tag{4}$$

or the chi-squared statistic

$$\chi^2 = \sum_i \frac{\left(P_{\mathrm{asset}}\left(o(T_i)|D_j\right) - P_{\mathrm{agg}}\left(o(T_i)|D_j\right)\right)^2}{P_{\mathrm{agg}}\left(o(T_i)|D_j\right)} \tag{5}$$

where P_{agg} is the aggregate probability distribution, and P_{asset} is the asset-specific probability distribution. A simple measure inspired by prior experiments was the conditional variance of the aggregate distribution

$$\mathrm{Var}\left(o(T_i)|D_j\right) = E\left[\left(o(T_i) - E\left[o(T_i)|D_j\right]\right)^2 |D_j\right]. \tag{6}$$

Similarly, several options exist for $g(n)$, ranging from $g(n) = n$ to some polynomial or exponential function of n. In our experiments, we found that $f(o(T_i), D_j) = \mathrm{Var}(o(T_i)|D_j)$ and $g(n) = n^{1/q}$ worked well.

Before we describe the experimental design used to test our hypothesis, we will look at some of the related work in Bayesian diagnosis and distributional smoothing.

IV. Related Work

The idea of applying Bayesian methods, in general, and Bayesian networks, in particular, to diagnosis is not new. Early Bayesian methods involved manually constructing models as an alternative to rule-based expert systems [9]. Perhaps the best-known Bayesian network method is the "Quick Medical Reference-Decision Theoretic" (QMR-DT) model [10]. The QMR-DT model was a "bipartite" network, where the diagnoses were root nodes, and the tests/observations were leaf nodes. Diagnoses were directly connected to tests. This model is similar to the naive Bayes model [see (1)] in that a naive Bayes network is also bipartite. QMR-DT, however, does not employ the conditional independence assumption.

More recently, Lerner *et al.* applied Bayesian networks to perform fault diagnosis in dynamical systems [11]. Their approach made use of a hybrid dynamic Bayesian network (DBN) [12] to represent the dynamics of the system. This approach is similar to the factorial hidden Markov model (HMM) approach used by Singh *et al.* [13]. In their approach, factorial HMMs were used to incorporate historical information for the purposes of multiple-fault diagnosis.

Although Bayesian techniques are used in many fields of computer science, the N-gram models used in NLP use smoothing techniques that bear some resemblance to the distributional smoothing technique for Bayesian diagnostics described in this paper [14]. One use of the N-gram technique is to classify text. For each type of text, a separate N-gram model is trained on texts of that type, for example, astronomy articles for one model and astrology articles for another. The typical N-gram sets $N = 3$ and is called a "trigram." A trigram is the conditional probability of a third word given the first and second words. The trigram model is built by calculating all of the trigrams from the training texts $P(w_3|w_1, w_2)$, $P(w_4|w_2, w_3)$, etc. It is, essentially, a second-order Markov chain. Using these models, we can then find the product of those probabilities for a text we want to classify and determine which one has the higher probability. In this case, whether the new text is astronomy or astrology. This simple approach uses unsmoothed maximum-likelihood estimates for the trigrams.

The problem arises in NLP that no matter how many texts are used in training, it is always possible that a new text will have something slightly different than anything seen before. This is the same problem that the m-estimate is designed to handle in general Bayesian classification. The NLP response is to use

smoothing or a related technique called backoff (and sometimes both). A variety of smoothing methods have been introduced for N-gram models over the years, including Add-λ (which is similar to the m-estimate) [14], Witten–Bell discounting [15], and Good–Turing discounting [16].

Essentially, backoff is a technique employing the strategy that states if a trigram equals zero, use the bigram. If the bigram equals zero, use the unigram. Finally, if the unigram is zero, back off to a uniform probability [17]. While our approach has much more in common with smoothing, to a certain degree, our proposed technique does "back off" from the asset-specific estimate to the fleet-wide estimate during the smoothing process. The similarity ends there, because it can also move in the other direction as well (i.e., from fleet wide to asset specific).

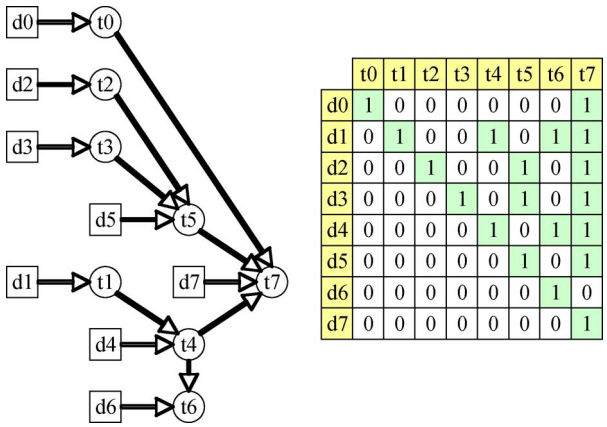
As an alternative to creating a single smoothed model, many have suggested using ensembles of models or combining models through averaging to improve the prediction accuracy. Ensemble methods seek to improve accuracy by combining recommendations from multiple classifiers [18]. Ensemble methods widely vary and include, for example, bagging, boosting, and mixtures of experts.

Bagging normally involves the creation a set of classifiers by using bootstrapping to resample the available data. Boosting involves creating successive classifiers trained on the mistakes of the previous classifier. Both approaches have been used in classifiers used for diagnostics [19], [20]. Mixtures of experts create a meta-classifier that combines the results of simpler classifiers and have been successfully used with Bayesian approaches to classification [21], [22].

An alternative ensemble-based approach involves generating several models and combining their predictions through model averaging. Madigan and York describe an approach to Bayesian model averaging where they generate a baseline model and then generate alternative models using a Markov chain Monte Carlo approach [23]. Meila and Jaakola discuss approaches to performing exact model averaging over tree-based Bayesian networks [24], and Dash and Cooper showed how to perform exact model averaging over naive Bayesian classifiers [25]. All of the ensemble methods differ from ours in that they construct multiple models and combine their predictions. Model averaging specifically differs from our approach, not only in averaging over multiple models but in training over the same data set as well. We derive and smooth estimates of probabilities within a single model using data drawn from different populations.

V. Experimental Design

To test our hypothesis, we generated synthetic data that were known to reflect asset-specific heterogeneity both in terms of failure rates, test outcomes, and diagnoses. We started with known and consistent diagnostic relationships modeled using a *D-Matrix* [26]. The particular D-Matrix represented eight tests over eight possible component failures. Because each row in the D-Matrix represents a test signature or pattern of passing and failing tests and a corresponding diagnosis, rows were repeatedly extracted from the matrix in proportion to different failure distributions to create the data. These different failure distributions are one kind of asset-specific variation that may be present in fleet-wide data and are eventually used in the Bayesian diagnostic model to estimate the prior probability distributions. The logic model and D-Matrix for this idealized system used for data generation can be seen in Fig. 2.



	t0	t1	t2	t3	t4	t5	t6	t7
d0	1	0	0	0	0	0	0	1
d1	0	1	0	0	1	0	1	1
d2	0	0	1	0	0	1	0	1
d3	0	0	0	1	0	1	0	1
d4	0	0	0	0	1	0	1	1
d5	0	0	0	0	0	1	0	1
d6	0	0	0	0	0	0	1	0
d7	0	0	0	0	0	0	0	1

Fig. 2. Sample logic model and D-Matrix for experimental evaluation.

On the other hand, the test signatures themselves end up represented in the Bayesian diagnostic model as estimates of the likelihoods. These likelihoods have two important characteristics. First, they represent a transformation of the test outcome and diagnosis relationships in the D-Matrix rows to probabilistic representations. This reduction is advantageous when the relationships need to be learned from (possibly noisy) test data. Second, the likelihoods in the naive Bayesian classifier represent an assumption about the conditional independence between tests given the diagnoses. Despite the assumption not generally holding in practice, the naive Bayesian classifier consistently performs well. In fact, a naive Bayesian classifier can learn the diagnostic model represented by a D-Matrix with 100% accuracy, as long as no diagnosis is present more than once in the D-Matrix [5], [27].

To introduce a degree of asset-specific variability in the fleet-wide data with respect to the likelihoods, we must sample from slightly different D-Matrices. We start with the baseline D-Matrix and create small changes. For example, perhaps a certain test always passes or always fails, or a certain test always fails for a particular diagnosis. The rationale for these changes is that usage profiles, extreme environmental effects, or variations in online testing conditions may have caused the real-world diagnostic relationships to diverge from the baseline.

To simulate the different conditions affecting diagnostic performance, we used five different failure distributions and five different D-Matrices, creating a total of 25 different combinations of specific assets. Examples of failure distributions include uniform probability and "one bad actor" distributions, where one component was significantly more likely to fail than the others. Examples of the different D-Matrices—reflecting different usage profiles or environmental conditions—included the baseline D-Matrix and D-Matrices, where one test always failed and another where one test always passed. All variations are fully described in [8].

As previously noted, a naive Bayesian classifier can learn training data directly derived from a D-Matrix with 100%

accuracy. Therefore, to introduce a degree of realism into the data, various levels of noise were added. This was accomplished by converting an expected pass or fail result of each test from the test signature into two real values: one for pass and one for fail. Each real value was then perturbed by independently and identically distributed Gaussian white noise with zero mean of different variances. If a passing value fell below a certain threshold, it was converted into a false positive or failure; otherwise, it was kept as a pass. Similarly, if a failing value fell above that threshold, it was converted into a false negative or pass; otherwise, it was kept as a failure.

When generating the data, noise was introduced by varying the standard deviation of the real values from 0.0 to 0.1 in 0.01 increments. Different data set sizes for each asset were also generated, ranging from 25 to 5000 observations. For each noise level and data set size, three naive Bayesian diagnostic models were constructed: a fleet-wide model, a set of asset-specific models, and a set of combined models using distributional smoothing. Because the fleet-wide model aggregates all of the available data, each fleet-wide model is trained using data sets with size MN, where $M = 25$ is the number of assets.

We ran 30 trials for each experiment (N and noise level combination). Each trial used 66% of the data to train and 34% of the data to test the model. New data were generated for each trial. Results were averaged over the trials, and a two-tailed difference of means test (t-test) was used for all comparisons with a significance level of 0.05. All random selection was stratified first by asset (if necessary) and then by diagnosis. The m-estimate was set with $p = 0.001\%$ and $m = 1$ in all cases, except in the combined model, where distributional smoothing was used to estimate the likelihoods. In that case, the special m and p were used from (3). For the combined model's estimate of the smoothed likelihoods, the user-defined parameters k and q were set to 100 and 1.2, respectively. Choosing a diagnosis at random broke all classification ties. For a more in-depth explanation of the experimental design, see our previous papers [1], [2], [8].

TABLE I
NUMBER OF ASSET-SPECIFIC MODELS AS GOOD AS THE FLEET-WIDE MODEL (OUT OF 25)

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	25	25	24	25	25	20	15	15	13	11	10
50	25	25	25	25	25	16	11	8	5	8	8
100	25	24	25	25	25	17	7	7	9	12	16
250	25	25	25	25	25	7	2	10	16	25	21
500	25	25	25	25	25	6	17	20	25	25	23
1000	25	25	25	25	8	14	23	23	24	25	23
2500	25	25	25	25	10	24	25	25	25	25	23
5000	25	25	25	25	19	25	25	25	25	25	23

TABLE II
NUMBER OF SMOOTHED MODELS AS GOOD AS THE FLEET-WIDE MODEL (OUT OF 25)

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	25	25	24	25	25	25	25	25	25	25	25
50	22	22	22	25	22	23	25	25	25	25	25
100	25	25	25	25	25	25	25	25	25	25	25
250	25	25	25	25	25	25	25	25	25	25	25
500	25	25	25	25	25	25	25	25	25	25	25
1000	25	25	25	25	25	25	25	25	25	25	25
2500	25	25	25	25	25	25	25	25	25	25	25
5000	25	25	25	25	25	25	25	25	25	25	25

VI. RESULTS

Our experimental results are presented in Table I. We compare the differential performance of asset-specific models or smoothed models with the fleet-wide model. We look at the case where the set of models is "just as good as" the fleet-wide model (t-statistic ≥ -1.96) and when the set of models is "better than" the fleet-wide model (t-statistic > 1.96). With five failure distributions and five usage profiles, there are 25 different possible models.

Table I shows the results for the asset-specific model compared with the fleet-wide model. This pattern is what originally inspired this research when we were investigating the accuracy of asset-specific models. When the noise level is zero, all of the asset-specific models are at least as good as the fleet-wide model (with a few random hits here and there). This trend continues until about noise level 0.04, when some of the asset-specific models begin to lose accuracy relative to the fleet-wide model. As noise increases, the drop off in accuracy occurs at increasingly smaller N but also returns with increasingly smaller N. For example, at noise level 0.05, the accuracy steadily drops off from an initial value of 20 but begins to rebound at $N = 1000$. On the other hand, with noise level 0.08, the drop off starts after $N = 25$ but begins to rebound with $N = 250$.

The results in Table I are the frame of reference for the remainder of the experiments. Ideally, we want all of the cells with values less than 25 to be 25 when using the fleet-wide model. However, rather than actually using the fleet-wide model, we performed distributional smoothing by including the fleet-wide data in the estimate of the likelihoods for the asset-specific models, resulting in a set of smoothed models.

As shown in Table II, the set of smoothed models was able to achieve the desired result. Except for a run at $N = 50$, all 25 of the smoothed models are at least as good as the fleet-wide model. While these results are encouraging, they must be tempered with the realization that, ultimately, we want the emphasis to be on the "*or better*" part of the "as good or better." After all, the fleet-wide model is always as good as itself. Table III shows the results of comparing the accuracy of asset-specific models against the fleet-wide model.

TABLE III
NUMBER OF ASSET-SPECIFIC MODELS BETTER THAN THE FLEET-WIDE MODEL (OUT OF 25)

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	1	1	2	2	1	2	1	1	2	2	3
50	1	1	1	1	1	1	2	1	0	0	0
100	1	1	1	1	2	1	1	1	1	1	2
250	1	3	1	1	2	1	1	4	5	10	12
500	1	1	1	1	1	1	3	8	11	17	19
1000	1	1	1	5	1	1	13	15	15	21	24
2500	1	1	2	2	2	15	18	16	17	23	25
5000	1	1	1	2	4	17	21	19	19	24	25

These results are typical of those we have found in previous research. Just as Table I shows that there are some asset-specific models that are less accurate than the fleet-wide model, there are some asset-specific models that are better. For example, at $N = 250$ and noise level 0.07, Table III shows that four of the asset-specific models are better than the fleet-wide model. Referring back to the same cell in Table I, we can see that ten were "as good or better." Thus, overall, for that cell, four asset-specific models were better, six were the same, and a full 15 were worse than the fleet-wide model in terms of accuracy. Note, however, that when N is large and the data are noisy, the asset-specific models are not just as good as the fleet-wide model; they are all generally better.

There is also another surprising and subtle result shown in Table III. Even when there is little or no noise, there are asset-specific models that do better than the fleet-wide model. This is not usually the case if the asset data are homogeneous. We, therefore, believe that the heterogeneities we introduced into the data and that may exist in actual test data can lead to the learning of inconsistent models.

The results for the smoothed models versus the fleet-wide model are shown in Table IV. Compared with the asset-specific models, there are some substantial gains over the fleet-wide model in terms of accuracy. This is particularly true once the noise level reaches 0.04 and beyond. It should be noted that where the gains are similar, such as the case with large N in the high noise area of the table, the asset-specific models also have some individuals that are worse than the fleet-wide model, whereas this is not the case for the smoothed models.

Table V shows our measure of the gains to be had from creating a set of smoothed probability models. Specifically, Table V shows the percent increase in accuracy, on average, over the fleet-wide classifier for the set of smoothed probability models. The gains are modest at low noise levels, which is to be expected. However, they are nearly 10% at the highest noise levels.

One of the areas we left for future research in our previous paper [8] was a sensitivity analysis of our results for different values of the user-defined parameters k and q in our implementation of (3). In the results just presented, we used $k = 100$ and $q = 1.2$. An initial sensitivity analysis examined the values that were 50% smaller and 50% larger while keeping the other parameter constant at the value used. In all cases, we looked to see how varying k and q affected the number of smoothed models better than the fleet-wide model.

TABLE IV
NUMBER OF SMOOTHED MODELS BETTER THAN THE FLEET-WIDE MODEL (OUT OF 25)

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	1	1	1	1	1	1	1	3	5	6	8
50	1	1	1	1	1	1	2	4	3	8	11
100	1	1	1	1	1	2	2	4	10	10	11
250	1	1	1	1	1	4	9	11	10	13	16
500	1	1	1	1	3	10	15	10	12	18	20
1000	1	1	1	1	5	15	16	16	15	21	23
2500	1	1	1	1	13	18	18	18	17	23	24
5000	1	1	1	1	16	17	20	19	19	24	25

TABLE V
AVERAGE PERCENT INCREASE IN ACCURACY BETWEEN THE SET OF SMOOTHED MODELS AND THE FLEET-WIDE MODEL

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	1.8	1.8	1.8	1.8	1.9	2.3	3.2	4.1	5.8	6.7	9.2
50	0.5	0.5	0.5	0.6	0.6	1.0	2.0	2.5	3.6	6.5	8.1
100	1.1	1.1	1.1	1.1	1.2	1.7	2.0	3.1	4.6	5.5	6.7
250	1.3	1.3	1.3	1.2	1.4	2.1	2.0	3.5	4.8	6.1	8.1
500	1.3	1.3	1.3	1.3	1.4	2.0	3.1	3.7	4.8	6.5	8.9
1000	1.3	1.3	1.3	1.3	1.5	2.1	3.1	4.1	4.8	6.8	9.3
2500	1.3	1.3	1.3	1.3	1.5	2.2	3.2	4.3	5.0	6.8	9.7
5000	1.3	1.3	1.3	1.3	1.5	2.2	3.3	4.3	4.9	7.1	10.0

The results showed that when we lowered k to 50, the total number of smoothed models better than the fleet-wide model dropped by four out of 660. Increasing k to 150 led to a net gain of only 18 out of 660. On the other hand, when q was decreased to 0.6, the net loss was 104 out of 660, but when q was increased to 1.8, the net gain was only 18 out of 660. This suggests that the parameter influencing the contribution of N to m (i.e., q) is very important to the overall accuracy. Even so, our settings still demonstrated strong performance, even without the more exhaustive evaluation of parameter values.

VII. DISCUSSION

In prior research, we examined the role of asset-specific models in improving the diagnostic accuracy for a fleet of

assets. While we found that asset-specific models could improve diagnostic accuracy, this was not always the case. Subsequently, we started to investigate ways that we might achieve the accuracy of both approaches in a single model. We realized that this would involve boosting the accuracy of the asset-specific model when the fleet-wide model was more accurate and boosting the accuracy of the fleet-wide model when the asset-specific model was more accurate. This investigation led to the idea of smoothing the distribution estimates using both fleet-wide and asset-specific data. Furthermore, we sought to make the smoothing endogenous by making the weighting factor a function of the characteristics, data quantity, and noise that we observed to affect the model accuracy.

The results in Table II support our hypothesis. We were able to get the desired effect by using distributional smoothing. With the results in Table II, we showed that the smoothed models were not worse than the fleet-wide model—unlike the unsmoothed asset-specific models. In the table, we showed that there were generally more smoothed models that were better than the fleet-wide model than asset-specific models that were better. Even when this was not the case, the number of smoothed models that were better was supported by the fact that none were worse. Finally, the table showed that the actual increase in accuracy, although data set and noise level dependent, could be substantial.

We plan to concentrate future research on four areas. First, the impact of the user-defined parameters on the formula for m should further be examined. This might improve not only the overall accuracy but also accuracy in cases of smaller N. In addition, examining various values of k and q on alternative data sets and models might provide more general insight into the range of their impact on model accuracy.

Second, our version of (3) was derived from empirical observation. In this paper, we generalized the equation to use a generic measure of distributional variation and suggested a couple of alternative specific measures. In future research, we would like to examine the effects of those specific measures on accuracy.

Third, smaller values of N may be a special case, warranting additional study. Specifically, we believe we need to investigate methods of leveraging small data sets to extract more information from them since, for many real-world systems, large amounts of training data may not be available.

Finally, since real systems are more complex than the artificial system studied here, we expect the test and maintenance data to contain nonlinearities. As we previously pointed out, the naive Bayes classifier is a linear discriminant. This would suggest abandoning naive Bayes in favor of learning more complicated network structures. In a previous work [7], we investigated the use of tree-augmented naive Bayes (TAN) in diagnostics [26]. This would be a good starting point.

VIII. Summary

Based on our research on asset-specific models for Bayesian diagnostics, we discovered that while heterogeneities in the data might support the use of asset-specific models, they were not always more accurate than a fleet-wide model. As a result, we have introduced a technique for distributional smoothing that used both asset-specific and fleet-wide data. We hypothesized that this technique could achieve the best accuracies of both models. To test the hypothesis, we constructed a data set that emphasized the types of data heterogeneities (namely, different failure distribution rates and test signatures) that could exist in a fleet-wide test and maintenance data set and would warrant asset-specific models. Our results showed that we could indeed improve accuracy by using the smoothing technique in a combined model.

Acknowledgment

The authors would like to thank M. Kaufman and C. MacDougall from the U.S. Navy for their input and inspiration in helping formulate the approach and the reviewers of IEEE AUTOTESTCON and this TRANSACTIONS for their helpful comments.

References

[1] S. Butcher and J. Sheppard, "Improving diagnostic accuracy by blending probabilities: Some initial experiments," in *Proc. 18th Int. Workshop DX*, Nashville, TN, May 2007.
[2] S. Butcher, J. Sheppard, M. Kaufman, H. Ha, and C. MacDougall, "Experiments in Bayesian diagnostics with IUID-enabled data," in *Proc. IEEE AUTOTESTCON*, Anaheim, CA, Sep. 2006, pp. 605–614.
[3] T. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
[4] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in *Proc. 10th Nat. Conf. Artif. Intell.*, 1992, pp. 223–228.
[5] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York: Wiley, 2001.
[6] J. Sheppard and M. Kaufman, "A Bayesian approach to diagnosis and prognosis using built in test," *IEEE Trans. Instrum. Meas.—Special Section on Built-In Test*, vol. 54, no. 3, pp. 1003–1018, Jun. 2005.
[7] J. Sheppard, S. Butcher, M. Kaufman, and C. MacDougall, "Not-So-Naïve Bayesian networks and unique identification in developing advanced diagnostics," in *Proc. IEEE Aerosp. Conf.*, Mar. 2006, pp. 1–13.
[8] S. Butcher and J. Sheppard, "Asset-specific Bayesian diagnostics in mixed contexts," in *Proc. IEEE AUTOTESTCON*, Baltimore, MD, 2007, pp. 113–122.
[9] G. Kleiter, "Bayesian diagnosis in expert systems," *Artif. Intell.*, vol. 54, no. 1/2, pp. 1–32, Mar. 1992.
[10] M. Shwe and G. Cooper, "An empirical analysis of a likelihood-weighting simulation on a large, multiply-connected medical belief network," *Comput. Biomed. Res.*, vol. 24, no. 5, pp. 453–475, Oct. 1991.
[11] U. Lerner, R. Parr, D. Koller, and G. Biswas, "Bayesian fault diagnosis in dynamical systems," in *Proc. 17th Nat. Conf. AAAI*, 2000, pp. 531–537.
[12] K. Murphy, "Dynamic Bayesian networks: Representation, inference, and learning," Ph.D. dissertation, Dept. Comput. Sci., Univ. California, Berkeley, CA, 2002.
[13] S. Singh, K. Choi, A. Kodali, K. Pattipati, J. Sheppard, S. Namburu, S. Chigusa, D. Prokhorov, and L. Qiao, "Dynamic multiple fault diagnosis and solution techniques," in *Proc. Int. Workshop DX*, Nashville, TN, 2007, pp. 383–390.
[14] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Upper Saddle River, NJ: Prentice-Hall, 2000.
[15] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Trans. Inf. Theory*, vol. 37, no. 4, pp. 1085–1094, Jul. 1991.
[16] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, no. 3/4, pp. 237–264, 1953.
[17] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no. 3, pp. 400–401, Mar. 1987.
[18] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, Jul.–Sep. 2006.
[19] Z.-H. Hu, Y.-G. Li, Y.-Z. Cai, and X.-M. Xu, "An empirical comparison of ensemble classification algorithms with support vector machines," in *Proc. 3rd Int. Conf. Mach. Learn. Cybern.*, Shanghai, China, Aug. 2004, pp. 3520–3523.

[20] Y. Li, Y.-Z. Cai, R.-P. Yin, and X.-M. Xu, "Fault diagnosis on support vector ensemble," in *Proc. 4th Int. Conf. Mach. Learn. Cybern.*, Guangzhou, China, Aug. 2005, pp. 3309–3314.
[21] C. M. Bishop and M. Svensen, "Bayesian hierarchical mixtures of experts," in *Proc. 19th Conf. Uncertainty Artif. Intell.*, 2003, pp. 57–64.
[22] M. K. Titsias and A. Likas, "Mixture of experts classification using a hierarchical mixture model," *Neural Comput.*, vol. 14, no. 9, pp. 2221–2244, 2002.
[23] D. Madigan and J. York, "Bayesian graphical models for discrete data," *Int. Statist. Rev.*, vol. 63, pp. 215–232, 1995.
[24] M. Meila and T. Jaakola, "Tractable Bayesian learning of tree belief networks," in *Proc. 16th Conf. Uncertainty Artif. Intell.*, 2000, pp. 380–388.
[25] D. Dash and G. Cooper, "Exact model averaging with naïve Bayesian classifiers," in *Proc. Int. Conf. Mach. Learn.*, 2002, pp. 91–98.
[26] W. Simpson and J. Sheppard, *System Test and Diagnosis*. Norwell, MA: Kluwer, 1994.
[27] J. Sheppard and S. Butcher, "A formal analysis of fault diagnosis with *D*-Matrices," *J. Electron. Test.: Theory Appl.*, vol. 23, no. 4, pp. 309–322, Aug. 2007.
[28] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, no. 2/3, pp. 131–163, Nov. 1997.

Stephyn G. W. Butcher (M'04) received the B.A. degree in economics from California State University, Sacramento, the M.A. degree in economics from The American University, Washington, DC, and the M.S. degree in computer science from Johns Hopkins University, Baltimore, MD, where he is currently working toward the Ph.D. degree in computer science with the Whiting School of Engineering.

He has served as a Lecturer in economics and Grader in computer science. His research interests are mainly in machine learning and include Bayesian networks and evolutionary computation.

John W. Sheppard (M'86–SM'97–F'07) received the B.S. degree in computer science from Southern Methodist University, Dallas, TX, and the M.S. and Ph.D. degrees in computer science from Johns Hopkins University, Baltimore, MD.

He has recently been named the Right Now Technologies Distinguished Professor in Computer Science at Montana State University, Bozeman, and holds an appointment as an Associate Research Professor in computer science at Johns Hopkins University. He 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. He has authored more than 100 publications in artificial intelligence and diagnostics, including an authored book and an edited book.

Dr. Sheppard currently serves as the Vice Chair of the IEEE Standards Coordinating Committee 20 (SCC20) on Test and Diagnosis for Electronic Systems, the Secretary and Past Chair of the Diagnostic and Maintenance Control subcommittee of SCC20, and the Official Liaison of the Computer Society Standards Activities Board to SCC20.