Improving Diagnostic Accuracy by Blending Probabilities: Some Initial Experiments

Stephyn G. W. Butcher **John W. Sheppard**

Numerical Intelligent Systems Laboratory
Department of Computer Science
The Johns Hopkins University
3400 N. Charles Street
Baltimore, Maryland 21218
{sbutche2,jsheppa2}@jhu.edu

Abstract

Inspired by the impending availability of asset specific data on several US Department of Defense programs, in a previous paper we looked at the possibility that a set of Bayesian diagnostic models constructed from asset specific data would outperform a single Bayesian diagnostic model constructed from all of the data. There were situations where a set of asset-specific classifiers was superior to a single composite classifier but it wasn't universally the case. The hypothesis in this paper is that a blended classifier can be constructed to take advantage of the best of both worlds: have a composite classifier's accuracy when its individual accuracy was greater and have an asset specific classifier's accuracy when its accuracy was greater. Our experiments suggest that a split classifier—one that uses asset-specific data to estimate priors and composite data to estimate the likelihoods—can more correctly represent the distributions in the underlying diagnostic problem.

Introduction

In a previous paper, we investigated the possible advantages and disadvantages of creating asset specific diagnostic models instead of a single diagnostic model covering all assets (Butcher *et al.* 2006). The theoretical impetus was the general observation that when constructing any classifier, the more closely the distribution of the sample data matches the distribution of the target population, the more accurate the model will be. In terms of Bayesian diagnostics, this suggested that if assets or groups of assets experienced distinct failure patterns, a set of diagnostic models tuned to the individual assets should be more accurate than a single diagnostic model covering all assets. However, such an approach requires that test and maintenance data be tagged with unique identifying information.

The practical impetus for the paper was the incipient availability of the required asset specific data through the Department of Defense's Item Unique Identification (IUID) numbers. When DoD's IUID program is fully implemented, maintenance and testing data that is tracked to specific assets will be readily available, which is exactly what this approach would require.

To test our hypothesis, we performed experiments comparing the accuracy of a single classifier against a set of asset specific classifiers over ranges of data sizes and levels of noise in the data. The main result was that a set of asset-specific classifiers was often better than a single composite classifier especially when noise levels were high and data was sparse. Unfortunately, the hypothesis was not supported universally. Moving in the direction of less noise, it was often the case that asset-specific classifiers were less accurate overall than the composite classifier. There was also a broad middle ground where they were both equally accurate.

The purpose of this paper is to answer a question left previously unanswered: if a set of asset specific classifiers is sometimes better and sometimes worse than a single classifier but just as good the rest of the time, when should each be used?

While the patterns in our results suggested that weighting the two classifiers might be the solution, we rejected applying a simple ensemble approach. Instead we hypothesized that we could *blend* the probabilities within each classifier of the set and be as accurate as either of the original approaches, thus attaining the best results of each. Surprisingly, our results show that a set of classifiers using blended probabilities was able to *outperform* both the original asset specific set and the composite classifiers. This result led us to identify a simpler blending scheme for the constituent probabilities based on the structure of the problem. We leave for future work problems that may require the more complicated blending scheme.

The plan of the paper is as follows. The first section will give some background on the Bayesian approach to diagnostics. The second section will discuss some related work in ensemble methods. The third section describes the experimental design. The fourth section reviews the results of our previous experiments. In the fifth section we look at our current results. In the final section we summarize our findings and make suggestions for future work.

Bayesian Approaches to Diagnostics

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. In previous work, we showed how diagnosis and classification are related through the D-Matrix Model (Sheppard and Butcher 2007). We decided to investigate Bayesian methods to diagnosis because they derive models based on sound mathematical principles, 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, Sheppard and Kaufman (2005) provided a detailed derivation of a simple model for Bayesian diagnosis. We have also demonstrated how both the Naïve Bayes Classified (NBC) and the Tree-Augmented Bayesian network (TAN) perform on a small sample of IUID-enabled data for a US Navy weapon system (Sheppard *et al.* 2006). For the purposes of this paper, the NBC will be sufficient for the experiments we are going to perform.

The primary assumption for NBC is that the evidence variables in the network (i.e., the tests) are conditionally independent of each other given the class (i.e., diagnosis). 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, we will apply a simple network structure corresponding to the form shown in Figure 1. Note that this structure can be modified where there is a separate Boolean node D_i for each diagnosis rather than a single composite diagnosis node. This leads to the so-called naïve Bayes "multi-net" (Friedman *et al.* 1997; Duda *et al.* 2001).

Under the naïve Bayes model, we attempt to find the specific diagnosis that maximizes the *a posteriori* probability of the diagnosis given the set of observations (i.e., test results). Let $o(T_i)$ be the discrete outcome (e.g., PASS or FAIL) for some test T_i). Then

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

Unfortunately, the problem remains that the size of the joint distribution over the tests is exponential in the number of tests. But under the naïve Bayes assumption, we can simplify the classification rule to the following:

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

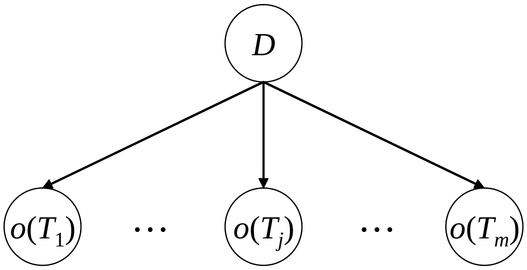


Figure 1. Bayesian diagnostic model

Given a set of training data mapping test results to faults detected, we can "learn" and NBC by observing that $P(D_i)$ is the frequency of occurrence of a particular fault in the data set. Similarly, $P(o(T_j) \mid D_i)$ is the frequency of test outcome $o(T_j)$ considering only the particular diagnosis D_i. What is remarkable about this simple model is the considerable effectiveness it has demonstrated in numerous experiments and implementations (Langley *et al.* 1992).

An important ramification of the NBC rule is that if any $P(o(T_j) \mid D_i)$ should happen to be zero then the entire value of the expression for that particular D_i zeroes out. This is not generally what we want, especially if we have learned our network from sparse training data. To prevent the classification rule from "breaking", the typical solution is to use a default estimate of the likelihoods. Although we will have more to say about this later, the approach we use is an m-estimate calculated as follows (Mitchell 1997):

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

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.

In spite of the relative accuracy of NBC, we note that a simple trained classifier is unlikely to be sufficient by itself to accurately diagnose faults. Accurate diagnosis generally requires a model created initially by experts and matured as more data is acquired. The *full* diagnostic problem will probably only be solved using classifiers with other types of diagnostic models (Wilmering and Sheppard 2007).

Related Work

One approach to combining models to improve diagnostic accuracy is through the use of "ensemble methods." Ensemble methods seek to improve accuracy by combining recommendations from multiple classifiers (Polikar 2006). Ensemble methods vary widely and include, for example, examples 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 (Hu *et al.* 2004; Li *et al.* 2005). Mixtures of experts create a meta-classifier that combines the results of simpler classifiers and have been successfully used with Bayesian approaches to classification (Titsias and Likas 2000; Bishop and Svensen 2003).

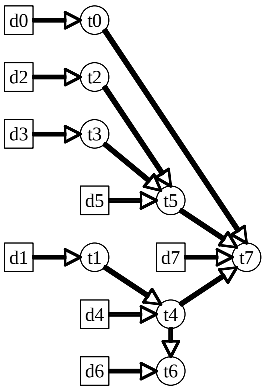
Our research differs from typical ensemble methods in a number of ways. First, while we create a set of classifiers, each classifier is tied to a specific asset. There is no voting because the correct classifier can be determined by context. Second, when creating the classifiers, we apply "blending" at a lower level of abstraction than at the level of the classification results. Although we emphasize the goal of obtaining the best accuracy of either the asset-specific or composite classifiers, we seek to achieve this by combining asset-specific and composite data to estimate the probabilities for each asset-specific classifier.

Experimental Design

To test our hypotheses, we first generated synthetic data. We use a hypothetical system consisting of eight components that can be arranged in various ways. Each component is subject to failure and that failure is detected by a combination of eight tests that can either pass or fail. Depending on how the components are arranged, the diagnostic characteristics of each system are captured by a corresponding D-Matrix (Simpson and Sheppard 1994). Figure 2 shows the arrangement of components and the corresponding D-Matrix for the system used as a basis to generate the data for the experiments in this paper. Each d*i* corresponds to a component that can fail and each t*j* corresponds to a test. In the case of failure, then d*i* corresponds to the diagnosis. Each row in the D-Matrix is a signature relating expected test outcomes (Pass = 0 or Fail = 1 for each test) to a particular diagnosis.

Because the main purpose of the experiments is to test hypotheses related to how asset specific data might be used to improve diagnostic accuracy, the data needs to be different for different assets. One way to introduce the required differences is to associate a different component failure distribution with each asset. For example, "Asset A" might always have trouble with component 3 (d3). In this case, the probability of d3 failing will be relatively higher than the probability of d0–d2 and d4–d7 failing. For these experiments, we created ten such hypothetical failure distributions each expressing a different behavioral property. The actual distributions used and the properties they represent are described in Butcher *et al.* (2006).

The D-Matrix (system) and component failure distributions (assets) form the foundation for generating the synthetic data. For each data set, *N* data points are generated for each asset using the D-Matrix and a particular fault distribution. For example, if *N* = 100, creating data for "Asset A" (described above) involves creating data that includes seven each of signatures d0–d2 and d4–d7 but 53 of signature d3. This process is repeated for each asset and each *N* value: 25, 50, 100, 250, 500, 1000, 2500 and 5000.



	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

Figure 2. Logic model and D-matrix

At this point, we have a collection of data sets that represent the D-matrix perfectly. However, in the real world, our test results are not likely to originate from clean Pass or Fail test readings, nor are they always perfect. The actual measurements are subject to varying levels of noise. We also note that an NBC can easily learn the diagnostic concept represented by any D-matrix to 100% accuracy as long as each row in the matrix corresponds to a unique diagnosis because the concept represented is linearly separable (Sheppard and Butcher 2007). In the present case, this will happen whether the data is segregated or aggregated but for these experiments we require at least the possibility that these diverge. So to both inject a degree of realism into the data and to prevent the problem from becoming trivial, we add noise to the data.

Assume we have determined typical "raw" values for a test passing and failing for a specific system and use those values as the means of a Gaussian distribution of equal variance (since we would be using the same measurement device). Based on Bayes decision theory, assuming equal loss, the optimal decision threshold is midway between the two means (Duda *et al.*, 2001). Using different variances, we introduce noise into the data in the following manner. When a test signature is copied into the data set, each test is examined. A random value is generated with the corresponding Pass or Fail distribution, and the result is compared to the decision threshold[1]. The outcome is then determined based on where the value falls with respect to this threshold. For example, if the result of a particular test is supposed to indicate a Pass (a "0" in the data), a random value is generated with the passing mean and the specified variance. If the resulting value is within the nominal limits,

[1] For these experiments, we assume only a single test limit is applied to determine Pass or Fail. In fact, this is easily extended to the more realistic case but was deemed unnecessary for these experiments.

the test outcome is kept as a PASS. If it is lower than the nominal limit, the test result is changed to FAIL. Standard deviations (rather than variances) of 0.00 to 0.1 in 0.01 increments are used for a total of 11 different noise distributions.

In Butcher *et al.* (2006), we ran experiments on three systems, ten assets, eight data set sizes, and 11 noise levels for a total of 2,640 data sets. Using this data, NBCs were created for each of ten assets using asset-specific data for a particular system, data set size (*N*), and noise level as well as a composite NBC using aggregated data. Thus the composite NBC was trained and tested with 10*N* data examples whereas the asset-specific classifiers were each trained with *N* examples. This comports well with real world experience—if one had data for ten assets and had the option of creating ten classifiers or one aggregate classifier, one would not throw 90% of the data away.

For all experiments, the NBC learning algorithm was repeated with 30 trials using 66% of the data to train the NBC and 34% of the data to test the NBC during each trial. New data was generated for each trial. All random selection was stratified first by system (if necessary) and then by diagnosis (class). The *m*-estimate was set with p = 0.001% and $m = 1$. The value of p was set low to make sure that the classification rule doesn't degenerate on the one hand but, on the other hand, the classification is not influenced. Choosing a diagnosis at random breaks all classification ties.

Previous Results

In Butcher *et al.* (2006), we hypothesized that a set of asset specific classifiers where each classifier was trained with its own data would be more accurate overall than a single composite classifier trained using all of the data. However, we didn't necessarily expect this to be true for all *N* and noise levels. Although the NBC has been shown to train well on relatively few examples (Langley *et al.* 1992), consider an aggregated data set of $N = 100$ samples. If there are ten assets, this leaves, on average, only ten instances per asset. If there are ten possible diagnoses, this leaves, on average, only one diagnosis per asset. At this point, we may not even be able to determine if assets actually have substantially different fault distributions. Thus we were really addressing two questions: *should* we segregate the data and *when* should we segregate it.

While considering the first question of *should* we segregate, we observed patterns suggesting an answer to the second question of *when*. During the present round of experiments we validated our previous findings and the results at noise levels 0.0, 0.05 and 0.1 are presented in Table 1. The table shows the number of asset-specific classifiers that were at least as accurate as the composite classifier based on a *t*-test for the difference of means (α = 0.05).

This pattern was typical of results reported by Butcher *et al.* (2006). At low noise levels, both approaches were equally accurate, and this was generally true in the noise range of 0.0 to 0.03. After 0.03, there was a transitional period where some asset-specific classifiers were more accurate than the composite classifier and some were less accurate than the composite classifier. However, once a high enough noise level was reached, usually about 0.06, the accuracy of asset-specific classifiers began to increase relative to the composite classifier. This trend was accelerated when the samples were smaller and the noise levels higher.

Table 1. Asset-specific classifiers vs. composite classifier.

	Noise Level (Std. Dev.)		
N	**0**	**0.05**	**0.1**
25	10	7	7
50	10	5	4
100	10	5	8
250	10	0	10
500	10	1	10
1000	10	2	10
2500	10	8	10
5000	10	8	10

These results were encouraging for asset-specific classifiers in general but because the asset-specific classifiers were not universally superior, we needed a way to decide when to use each approach. However, because the pattern was fairly regular, this suggested that there might be a way to get the best of both worlds: get the accuracy of composite classifiers when they are more accurate and get the accuracy of asset-specific classifiers when they are more accurate. Nevertheless, we were intent on finding an approach that was not *ad hoc*.

New Results

Based on the patterns observed in our prior results, we hypothesized that a data-driven *blended* classifier built from both composite and specific data might achieve a higher level of *overall* accuracy than either previous approach taken singly. Formally, if $C(N,\sigma)$ is classifier *accuracy* as a function of data set size and noise, then we sought an approach with the following as the best case scenario:

$$C_b(N,\sigma) = \max\{C_c(N,\sigma), C_u(N,\sigma)\}$$

where b is blended asset-specific, c is composite and u is unblended asset-specific. As previously discussed, this is not an ensemble approach. There is still a set of asset-specific classifiers being trained, one for each asset. Instead each classifier uses probabilities learned from both asset-specific data and combined data for all assets. The open question was how to blend these two data sources within each asset-specific classifier to achieve the best case scenario.

As previously described, naïve Bayesian classification proceeds by choosing the class that maximizes the product of the prior probability of the diagnosis, $P(D_i)$, and the likelihoods over the tests given the diagnosis, $P(o(T_j) \mid D_i)$.

When determining each probability, the *m*-estimate is used to adjust the calculation as described above. The key components of the *m*-estimate are the values of *m* and *p*, the number of virtual examples and the Bayesian probability estimate of that likelihood. In the absence of better information (which is usually lacking), the *m*-estimate of the likelihood is assumed to take on a constant uniform distribution.

Our approach to blending the data sources leverages the *m*-estimate. We still create a set of asset specific classifiers where each classifier is calculated from asset specific data. The blending comes from applying a modified *m*-estimate calculated using the composite data, $p = P(o(T_j) \mid D_i)$. Additionally, we change the weight of the Bayesian estimate as data set size and noise vary with the aim of matching the patterns we observed in our previous results. Instead of using $m = 1$, we used an *m* that was proportional to *N* and the noise level:

$$m = \frac{k}{\mathrm{var}(o(T_i), D_j)\sqrt[q]{N}} + 1$$

where *k* and *q* are user defined constants. This formula was based on our observation that, keeping noise constant, as *N* increased, we wanted to weight towards the probability calculated from asset specific data and away from the probability calculated from composite data represented by the *m*-estimate. This formula reduces the number of virtual examples, *m*, representing the *composite* based probabilities as *N* increases. With *N* held constant and the noise level increasing, we also wanted to weight more towards the probability calculated from asset specific data and away from the probability calculated from composite data. The formula above fills both requirements, and Table 2 shows some sample calculations of *m* for various *N* and noise levels. In practice, *m* was calculated based on the actual data because the specific noisiness of the data is generally not known *a priori*. As a result, every probability is calculated with its own *m* value. When the variance was zero, it was simply omitted from the formula.

Probabilities calculated from *composite* data were calculated in the usual fashion and used an *m*-estimate with $m = 1$ and $p = 0.001\%$.

Table 2. Example values of *m*

	Noise Level (Std. Dev.)		
N	**0**	**0.05**	**0.1**
25	68400.0	2737.0	685.0
50	38388.7	1536.5	384.9
100	21545.3	862.8	216.4
250	10040.6	402.6	101.4
500	5636.5	226.4	57.3
1000	3163.3	127.5	32.6
2500	1474.6	59.9	15.7
5000	828.0	34.1	9.3

Table 3. Asset specific classifiers vs. composite classifier

	Noise Level (Std. Dev.)		
N	**0**	**0.05**	**0.1**
25	10	5	10
50	10	7	10
100	10	8	10
250	10	9	10
500	10	9	10
1000	10	9	10
2500	10	9	10
5000	10	8	10

Table 4. Comparing asset-specific classifier accuracies

N	**Blended better than Unblended**	**Blended worse than Unblended**	**Blended same as Unblended**
25	3	0	7
50	10	0	0
100	10	0	0
250	5	0	5
500	0	0	10
1000	0	0	10
2500	0	0	10
5000	0	0	10

To test our hypothesis, we constructed a composite classifier ("composite"), a set of asset-specific classifiers ("unblended specific"), and a set of blended asset-specific classifiers ("blended specific") by training and testing them as described in the experimental design section. We used $k = 100$ and $q = 1.2$ in our formula for *m.* The results are shown in Table 3 at noise levels 0.0, 0.05 and 0.1. Similar to Table 1, Table 3 shows the number of blended classifiers out of ten that were at least as accurate as the composite classifier. Cross-referencing with Table 1, we can see that the blended specific classifiers achieved a higher overall level of accuracy as compared to the unblended specific classifiers when compared to the composite classifier. For example, at noise level 0.05 and $N = 250$, no unblended specific classifier was at least as accurate as the composite. However, we can see that nine out of ten blended specific classifiers were at least as accurate as the composite.

Table 4 compares unblended specific and blended specific directly against each other for noise level 0.1. The table shows that at low *N*, the blended specific classifiers were more accurate than the unblended specific classifiers whereas at higher *N*, they have the same accuracy.

These results validated our hypothesis. However, as we looked at the results from a different perspective, we noticed something surprising. Because of how the blended specific classifiers were constructed, we did not contemplate in our hypothesis that they might be *more* accurate than either the composite classifier or unblended specific classifiers in some cases. As previous stated, we supposed that the bounds of accuracy for the blended classifier would be the accuracies of the composite classifier and the appropriate unblended asset-specific classifier. This didn't turn out to be the case. Instead, the blended classifier was more accurate than either classifier in several cases. Table 5 shows the results for noise level 0.1. There is clearly a pattern, especially at smaller *N*, showing the blended classifier was more accurate on average. This advantage declined as *N* increased.

Table 5. Average accuracies for the classifiers

		Average Percent Accuracy	
N	**Composite**	**Blended Specific**	**Unblended Specific**
25	60.4%	68.6%	62.3%
50	63.4%	68.5%	55.7%
100	64.9%	70.4%	63.0%
250	65.4%	70.6%	68.4%
500	64.9%	70.4%	69.9%
1000	65.4%	70.9%	70.9%
2500	65.5%	71.2%	71.2%
5000	65.4%	71.3%	71.3%

Despite the surprise, we were able to hypothesize an explanation for these results after considering the relationship between the data and the Bayesian model more closely. For a given system, the data reflected two things, different noise levels and different failure distributions—one for each asset. However because the noise level and test signatures where the same for all assets, each classifier was trying to estimate the same *likelihoods*—no matter if it was the composite or one of the unblended asset-specific classifiers. Generally because the composite classifier had 10 times the data, at low *N*, it was better at estimating these likelihoods.

However, the asset-specific classifiers were better at estimating the *prior* probabilities because these depended solely on the failure distributions and each asset had a different and distinctive failure distribution. Whether or not the composite or asset-specific classifier was more accurate depended directly on whether, depending on the noise level and *N*, the prior probabilities or likelihoods were more important for classifier accuracy.

To test this hypothesis, we created a third type of asset-specific classifier, a split asset-specific classifier ("split specific"). Each split specific classifier used data for that asset alone to calculate the prior probabilities and used data aggregated over all assets to calculate the likelihoods.

Comparing the number of split specifics that are as good or better than the composite we find that the split specifics are as good as or better than the composite classifier in the majority of cases. If we compare these results to those we obtained in previous experiments for the unsplit (original) specifics we see that the split specifics out perform the unsplit specifics. As expected, when compared to the blended specifics, the results are comparable (Table 6).

As a result of these three sets of experiments, we are able to answer each of our questions. First, sets of asset specific classifiers can be more accurate than a single composite classifier if the failure distributions are significantly different for the specific assets. However, because of how noise and data sizes interact, it isn't always clear when to use the composite classifier versus the asset specific classifier. Second, the experiments in this paper show that we can construct asset specific classifiers using blended probabilities that do obtain the best of both worlds.

Finally, we also demonstrate in the second set of experiments that if it is the case that the difference between the assets exists solely in their failure distributions, we can create a set of asset specific *split* classifiers that use asset specific data to estimate the priors and the aggregated data to estimate the likelihoods of the Bayesian model. Looking at the specific noise level of 0.1, we find a similar pattern of split specifics being at least as accurate or more accurate than the unsplit specifics (Table 7). When we compare the accuracies of blended and split specific classifiers directly, we find almost no difference in accuracy at all.

Table 6. Split-specific classifiers vs. composite classifier

	Noise Level (Std. Dev.)		
N	**0**	**0.05**	**0.1**
25	10	5	8
50	10	6	9
100	10	7	9
250	10	9	9
500	10	9	9
1000	10	9	9
2500	10	9	9
5000	10	9	9

Table 7. Comparing unsplit and split probabilities

N	**Split better than Unsplit**	**Split worse than Unsplit**	**Split the same as Unsplit**
25	1	2	7
50	1	1	8
100	1	0	9
250	10	0	0
500	10	0	0
1000	10	0	0
2500	7	0	10
5000	0	0	10

It should be noted that although our original research was inspired by the IUID system of the Department of Defense, the approach is valid any time the population of assets can be split into subsets with different failure distributions. For example, different runs or lots of assets may have different failure distributions. As maintenance actions are taken on a population of assets, especially over a protracted period of time, some parts may go out of production to be replaced by components with a different manufacturer. The systems may appear to be the same at the test level but the differing parts may give rise to different failure rates. In short, whenever subsets of a system population begin to exhibit different failure distributions, the accuracy of Bayesian diagnostic models can be improved by using subset (possibly asset) specific data to construct the prior probabilities and aggregate data to construct the likelihoods.

Fortunately, the MTBF for modern systems is generally high. Unfortunately for those attempting to construct diagnostic models, this means failure data is not going to come all at once or even frequently except for the largest asset populations. It follows that such identifying data should be collected at the start of a program because these differences may not become apparent until much later.

Future Work

The results in this paper were largely driven by our assumptions about what form asset specific behavior might take, how it might show up in the data, and how it could affect building Bayesian diagnostic models. We assumed that asset specific behavior took the form of different failure distributions, and because of the way Bayesian models are built, this affected only a certain part of the model, the prior probabilities.

However, this isn't the only way that asset specific behavior might manifest. From the point of view of Bayesian diagnostics, under what conditions might the likelihood distributions be affected? The likelihoods are the probability of a test result given a particular diagnosis. This would imply that data contain a mixing of asset behavior and testing. How might this come about?

As an example, consider a situation where diagnostic procedures are being prepared for a GPS system that will be installed on a wide variety of aircraft, and that GPS system has a built in health monitor. For example, it could be installed on a C-17 used for long-haul transport with little chance of requiring any extreme maneuvering where we might expect the equipment to function, and fail, as planned. Suppose, however, that the same model GPS system is installed on an F/A-18 that is flying maneuvers in hostile territory. Then more extreme maneuvers may be required, and this could lead to stresses on health monitoring equipment such that the test likelihoods need to shift to reflect greater sensitivity. Finally, consider a GPS system of the same model installed on a high-altitude trainer (e.g., a 747 used by NASA for astronaut training) where it is likely the aircraft will undergo frequent negative G-force maneuvers. A completely different set of test likelihood distributions might appear because of how the measurements are taken.

As a practical illustration of how this might affect learning a Bayesian diagnostic model, consider the D-Matrix for the functional system. Assume that we have three assets operating in environmental conditions such that a health monitor records the following data. For the first asset, data is recorded exactly as would be expected by the D-Matrix (Figure 2). For the second asset, the third test consistently gives the wrong reading. If the expected reading for diagnosis *i* was PASS, then would it read FAIL; FAIL, then PASS. For the third asset, one test in each signature consistently reads falsely in five of the eight test signatures (see Figure 3 for illustrative D-Matrices). In addition to these systematic differences, all readings are subject to varying levels of Gaussian noise (as before).

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

Environment #1

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

Environment #2

Figure 3. Environment-dependent D-Matrices

Because all of this data comes from the same system, the temptation is to aggregate all of the data together to arrive at the best possible model. Assuming a uniform distribution of failures for all three assets, it can easily be shown that aggregation in this case actually harms overall classifier accuracy. In fact, even with no noise, a composite classifier built from equal data from all three assets is only 87.5% accurate compared to 100.0% for the asset-specific classifiers—whether *N* is 25 or 5000.

As the noise level increases, accuracy for every classifier begins to fall but asset-specific classifiers still remain more accurate than the composite classifier. However, after a certain point, around noise level 0.08, there are more ties between the composite classifier and the asset-specific classifiers, especially at low *N*. For example, after running some preliminary experiments with noise level 0.08 and *N* = 50, the accuracy of the composite classifier was 51.5% but the accuracies of the asset-specific classifier for asset No. 1 was 48.4%; No. 2, 52.9%; and No. 3, 53.8%. In addition, none of the differences are statistically significant. Not until *N* = 250 are all of the asset-specific classifiers more accurate than the composite classifier (with statistically significant differences).

This small experiment suggests a number of things. First, in situations where the likelihoods are asset specific, we may have to reverse the "split" classifier so that the priors are estimated from composite data and likelihoods are estimated from asset-specific data. Second, where there are asset specificities arising both from differing failure distributions and test patterns, we may be required to blend probabilities as we did in the first experiment in this paper.

Third, test noise has a completely different effect on asset-specific classifiers when the specificity derives from testing than when the specificity derives from failure distributions. In the latter case, asset specific classifiers were more accurate at high noise and low *N*. In the former case, asset specific classifiers are less accurate at high noise and low *N*. This suggests that the blending will have to happen differently for prior probabilities than for the likelihoods. Finally, constructing, applying and maintaining multiple diagnostic models is relatively more expensive than doing the same for just one. We need to be able find ways to detect the condition and estimate the expected payoff from increased accuracy. We plan to explore all of these subtleties in future research on this topic.

Conclusions

We built on previous work that demonstrated under certain conditions that a set of asset specific classifiers can be more accurate than a composite classifier built from the aggregated data. However, asset specific classifiers were not universally more accurate and this left open the question of when to build a set of asset specific classifiers and when to build a composite classifier.

In this paper, we presented experimental results supporting our hypothesis that a blended approach could improve diagnostic accuracy. The blended approach—while not an ensemble approach—uses aggregated data to create the *m*-estimates and then weights each by a number of virtual examples calculated as a function of both *N* and the level of noise. Although there were still some user-defined variables involved, this avoided an *ad hoc* approach to picking and choosing asset specific classifiers or composite classifiers.

We also showed that because of the structure of the Bayesian model and the kind of asset specificity we were considering, a simpler "split" classifier could be constructed by using asset-specific data to estimate the priors and composite data to estimate the likelihoods.

Finally, we looked briefly into situations where the asset specificity of the data might also affect test signatures in the field and in turn would affect the estimated likelihoods. In this case, we hypothesized that a similar "split" classifier would need to be created for maximal accuracy. We also hypothesized that in the presence of both kinds of asset specificity, we would need to return to the blended approach. All of these problems are left for future work including how we might identify these non-homogeneous situations.

Acknowledgments

The authors would like to thank the anonymous reviewers of the DX07 program committee whose comments greatly improved the quality of the paper. This work was supported, in part, by the US Navy, NAVAIR, PMA 260.

References

Bishop, C. M. and M. Svensen, "Bayesian Hierarchical Mixtures of Experts", *Uncertainty in Artificial Intelligence: Proceedings of the Nineteenth Conference*, 2003.

Butcher S., Sheppard J., Kaufman M., Ha, H. and MacDougall, C. "Experiments in Bayesian Diagnostics with IUID-Enabled Data" IEEE AUTOTESTCON 2006, *Conference Record*, Anheim, CA: September 2006., pp 605-614.

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

Engelmore, R., and Morgan, A. eds. 1986. *Blackboard Systems.* Reading, Mass.: Addison-Wesley.

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

Hu, Z.-H., Y.-G. Li, Y.-Z. Cai and X.-M. Xu, "An Empirical Comparison of Ensemble Classification Algorithms with Support Vector Machines"., *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, Shanghai, China, August 2004, 3520-3523.

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

Langley, P., Iba, W., and Thompson, K., "An Analysis of Bayesian Classifiers," *Proceedings of the Tenth National Conference on Artificial Intelligence*, San Mateo, CA: AAAI Press, 1992, pp. 223–228.

Li, Y., Y.-Z. Cai, R.-P. Yin, and X.-M. Xu, "Fault Diagnosis on Support Vector Ensemble", *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, China, August 2005, 3309-3314.

Mitchell, T. *Machine Learning*, New York: The McGraw-Hill Companies, 1997.

R. Polikar, "Ensemble Based Systems in Decision Making", *IEEE Circuits and Systems Magazine*, 3rd Quarter 2006, 21-54.

Sheppard, J. and Butcher, S., "A Formal Analysis of Fault Diagnosis with D-Matrices" to appear in *Journal of Electronic Testing: Theory and Applications*, 2007.

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

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

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

Titsias, M. K. and A. Likas, "Mixture of Experts Classification Using a Hierarchical Mixture Model", *Neural Computation* 14, 2000, 2221-2244.

Wilmering, T. J. and Sheppard, J., "Ontologies for Data Mining and Knowledge Discover to Support Diagnostic Maturation", forthcoming, The 18th International Workshop on Principles of Diagnosis (DX-07), Nashville, TN, May 2007.