Using Tree Augmented Naive Bayesian Classifiers to Improve Engine Fault Models

Daniel L.C. Mack
EECS Dept.
Vanderbilt University
Nashville, TN 37212

Gautam Biswas
EECS Dept.
Vanderbilt University
Nashville, TN 37212

Xenofon D. Koutsoukos
EECS Dept.
Vanderbilt University
Nashville, TN 37212

Dinkar Mylaraswamy
Honeywell Aerospace
1985 Douglas Drive N
Golden Valley, MN 55422

Abstract

Online fault diagnosis is critical for detecting the onset and hence the mitigation of adverse events that arise in complex systems such as aircraft and industrial processes. A typical fault diagnosis system consists of a reference model that provides a mathematical representation for various diagnostic monitors that provide partial evidence towards active failure modes and a reasoning algorithm that uses a set-covering scheme to establish fault candidates and their rankings. However, this approach often suffers from incompleteness in the reference models and simplifying assumptions made by the reasoning algorithms. Incompleteness in such models take several forms, such as absence of evidence, errors and incompleteness in the mapping between evidence and failure modes, while inaccuracies in the reasoning algorithm arise from simplifying noise models and independence assumptions. In this paper, we describe a a Tree Augmented Naive Bayesian Classifier (TAN) approach to systematically extend a reference model using data from a system operating with and without faults. We compare the performance of the TAN models against the expert-supplied Naive Bayes models using data generated by simulation of an aircraft engine, and demonstrate that the TAN improves classification accuracy by finding new causal links among the system monitors.

1 Introduction

Aircraft are complex systems containing several interacting components and subsystems such as propulsion, electrical, flight management, avionics, bleed etc. Smooth operations of these components are essential to maintain aviation safety. However, any operating system degrades over time and aircraft components are no exceptions. Monitoring the system online for detecting the onset of unfavorable conditions and onset of intrinsic faults is essential for increasing aviation safety.

The current state of online fault diagnosis is focused on installing a variety of sensors onboard an aircraft along with a reasoning software to automatically interpret the evidence generated by them to access the presence of faults. One such state of the art system is the Aircraft Diagnostic and Maintenance System (ADMS) (Spitzer, 2007) that is used on the Boeing B777. The ADMS uses a fault propagation system reference model that captures the interactions between aircraft components under various operating modes. This expert-derived model is called the system reference model. Generation of this reference model is a manual process and often the most tedious step in the practical development and deployment of an ADMS.

Many of the shortcomings of the ADMS can be attributed to incomplete and incorrect information in the system reference model. As the engineering teams acquire additional knowledge from an operating fleet, these translate as expert heuristics rather than a systematic upgrade to the reference model that was generated at design time. In other words, a gap exists for systematic upgrades and increments to the reference model as vast amount of operational data is collected by operating airlines. Closing this gap using advances in data mining is the focus of this paper. In this paper we describe a specific data mining approach for augmenting an existing aircraft engine reference model as an alternative to ad hoc approaches.

Statistical analysis and designing classifier for discovering knowledge from real-world data is extensively studied. For example, Witten (Witten & Frank, 1999) describe several data mining approaches for producing black box models. Unfortunately, such models are extremely difficult to verify making them almost impossible to certify for airworthiness. Further, the lack of transparency in these models make it difficult to *append* this new knowledge to existing ADMS reference models. Hence for practical success, these data mining approaches have to "build upon" existing reference model structures rather than create something new and incur considerable engineering overhead cost.

It is this engineered approach to data mining that makes our approach somewhat unique. In other words, the data mining does not start from a clean sheet of paper, but from an existing ADMS reference model structure. In section 2, we describe a typical reference model structure along with the reasoning algorithm (called the W-algorithm). Next, we systematically enumerate the missing or partially correct information in this state of the art reference model. These gaps that formalize the data mining problem is described in section 3. Then we discusses the use of Tree-Augmented Bayesian Networks (TANs) as a data driven modeling structure that for diagnosis with casual probabilistic models in section 4. The data mining approach is illustrated using data from a high fidelity simulator. Section 5 discusses the CMAPS-S simulator and the data selection task for our experiments. Section 6 describes the experimental results using the CMAPS-S data set, and a comparison of a naive Bayesian model that replicates a reference model against a model derived using the TAN classifier learning algorithm. Metrics are defined for evaluating classifier performance, and a number of different experiments are run to examine different stages of flight. Section 7 presents a summary of our approach, and outlines our directions for future work for diagnostic and prognostic reasoning using the data mining algorithms.

2 Background on Reference Models

Model-based strategies for diagnosing large, complex, real-world systems rely on domain experts to craft the *reference models* used for monitoring and isolating faults. The complexity of the system makes it almost impossible to create complete physics-based models with reasonable resources; a more pragmatic solution is to rely on expert-generated cause-effect models. In simple terms, the reference model of the system being monitored can be represented as a bipartite graph consisting of two types of nodes: failure modes and evidence. The set F defines all *distinct* failure modes defined or enumerated for the system under consideration. A failure mode $fm_i \in F$ may be occurring or not occurring in the system. This is defined as the state of the failure mode. In our model, we allow only binary (occurring or not-occurring) states for the failure mode. We use the following shorthand notations regarding these assertions.

$$\begin{aligned} fm_i = 0 &\Leftrightarrow \text{The failure mode is not occurring} \\ fm_i = 1 &\Leftrightarrow \text{The failure mode is occurring} \end{aligned} \tag{1}$$

Every failure mode has a priori probability of occurring in the system. This probability is given by $P(fm_i = 1)$. A failure mode fm_k can occur (or not occur) independently of another failure mode fm_j occurring. That is, $P(fm_k = 1|fm_j = 1) = P(fm_k = 1)$.

To isolate and disambiguate the failure modes, the model also defines an entity called "evidence". The jth evidence is denoted by e_j and the set E denotes all distinct monitors defined for the system under consideration. The diagnostic monitor associated with the ith evidence can either *indict* or *exonerate* a subset of failure modes called its ambiguity group. The monitor m_i can take three mutually exclusive values allowing a monitor to express indicting or exonerating or unknown support for the failure modes in its ambiguity group. The notations are described in equation (2).

$$\begin{aligned} m_i = 0 &\Leftrightarrow \text{ Exonerating evidence} \\ m_i = 1 &\Leftrightarrow \text{ Indicting evidence} \\ m_i = -1 &\Leftrightarrow \text{ Unknown evidence} \end{aligned} \tag{2}$$

Ideally we want a monitor associated with evidence e_i to fire only when the failure modes in its ambiguity group are occurring. Given the fact that the i'th failure mode is occurring in the system, d_{ji} denotes the probability that there will be a monitor providing an indicting evidence under this condition.

$$d_{ji} \Leftrightarrow P(m_j = 1|fm_i = 1), \tag{3}$$

d_{ij} is called the detection probability of failure mode monitor fm_j with respect to the ith evidence. A monitor may fire when there is no failure mode present in the system. False alarm probability is the probability that an indicting monitor is present when there are no failure modes occurring in the system. That is,

$$\epsilon_i \Leftrightarrow P(m_i = 1|fm_j = 0, \forall fm_j \in F) \tag{4}$$

In summary, a reference model describes the relation between failure modes and monitors. The reference model is a 6-tuple defined as: $[E, F, D, Pr, \epsilon]$ where: E is evidence set, F is failure mode set, D is detection probabilities, Pr is a priori probability of failure modes, ϵ is false alarm rate for monitors.

Figure 1 illustrates an example reference model graphically, with fault modes (hypotheses) as nodes on the right, and diagnostic monitors (DM) on the left.Each

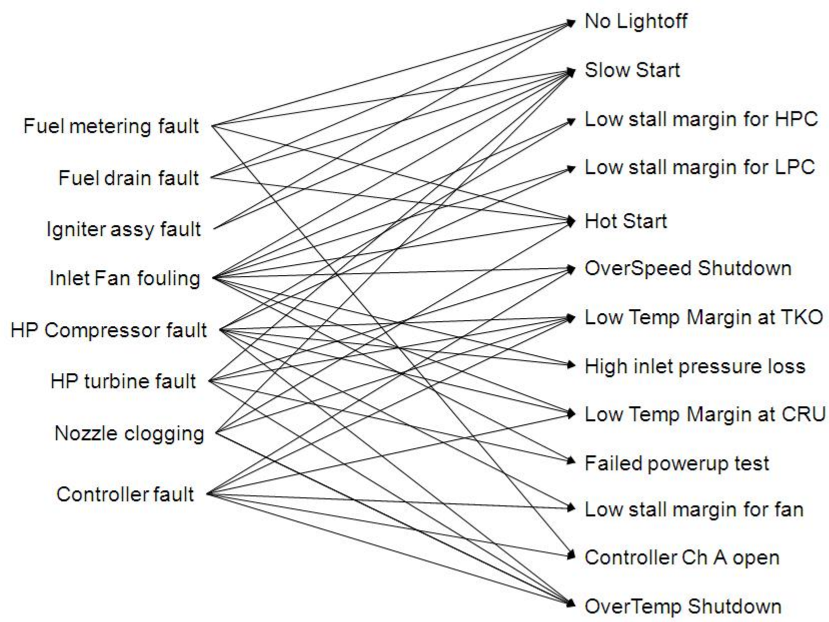

Figure 1: Example Reference Model

link would contain a detection probability, i.e., conditional probability $P(m_j = 1|fm_i = 1)$. In addition, fault nodes on the right contain the a priori probability of fault occurrence, i.e., $P(fm_i)$. Probabilities on the DM nodes indicate the likelihood that a particular monitor would indicate a fault in a nominal system. Bayesian methods are employed to combine the evidence provided by multiple monitors to estimate the most likely fault candidates.

The reasoner algorithm (called the W-algorithm) combines an abductive reasoning algorithm with a forward propagation algorithm to generate and rank possible failure modes. This algorithm operates in two steps: (1) *Abductive reasoning step*: Associated with each DM is an *ambiguity set*, $AG = \{fm_1, fm_2, \cdots fm_k\}$. This step assumes that the firing of the DM implies at least one of the faults in the ambiguity set has occurred; and (2) *Forward reasoning step*: For each fm_i belonging to AG, we extract all of the other DMs that support fm_i. We call this the list of supporting DMs, or the monitors of interest, i.e., $S - DM_i$ for fm_i. As additional monitors fire, AG reduces in size, and ideally, to a single element. Additional details about the reasoning algorithm is described in (Honeywell, 2010). The reasoning algorithm generates multiple single fault hypothesis, each hypothesis asserting the occurrence of exactly one failure mode in the system. The basic probability update rules assume independence of monitor firing events. In other words, $P(m_j, m_k|fm_i) = P(m_j|fm_i)\ P(m_k|fm_i)$ for all monitors m_j and m_k. The independence assumption implies that the reasoning algorithm treats the reference model as a Naive Bayes classifier. The direct correspondence between the reference model for diagnosis and the simple Bayesian structure provides opportunities to use a class of generative Bayesian model algorithms to build these model structures from data and enhance the existing structures produced by a domain expert. We discuss the problem statement next.

However, in cases of incompleteness or errors in the reference models, data mining approaches that use historical data from previous aircraft flights can be used to improve the model accuracy and precision. It is important that the data used by the learning algorithms include both nominal and faulty flights to extract the correct relations between the DMs and fault modes. We discuss this in greater detail in a subsequent section.

3 The Data mining problem

The reasoning algorithm nay not not reduce the ambiguity group to a single fault element. For example, all of the evidence (i.e., DMs) required to isolate the single fault may not fire, leaving the size of the ambiguity set to be greater than 1. In this case, the reference model is incomplete. This gap can be addressed by employing heuristics rules or systematically discovering new diagnostic monitors from vast amount of historical data.

The second source of error arises from the “independence assumption”. This assumption of independence may lead to certain hypotheses being assigned higher likelihood than the evidence truly implies. This assumption is made primarily because, causality (or correlation) between evidence in the system is extremely difficult to derive while the system is being designed and assembled. Such knowledge can only derived from an operating fleet. unknown but assumed to be either irrelevant or insignificant.

As implied above, the reference model when viewed as a single fault diagnoser can be interpreted as a Noisy-OR classifier, which is a simplified form of a standard Bayesian Network. A number of Machine Learning techniques for building Bayesian networks from data have been reported in the literature (Friedman, Geiger, & Goldszmidt, 1997) We have studied a number of these approaches in the framework of diagnostic and prognostic reasoning. Some important considerations have been the notion of independence among the monitors that support the diagnostic reasoning, and the incorporation of temporal relations through monitor variables as well as the causal structure implied by the reference model. We have also considered state-based hidden Markov Models (HMMs) and even more general Dynamic Bayesian Network (DBN) formulations to capture the dynamics of aircraft behavior and effects of faults on system behavior and performance. However, our initial set of choices has been governed by two important factors:

1. The data mining algorithms should be designed to provide information that supplements existing expert-generated reference models, as opposed to providing different formulations and different reasoner structures. It is very important that the experts be able to interpret the results of the data mining algorithms, and characterize them as:
   (a) new relations between monitors and fault hypotheses that will improve the reference model;
   (b) additional monitors (both simple and advanced) that help differentiate and provide support for specific diagnostic hypotheses; and
   (c) refinements to the conditional probability values between hypotheses and monitors.
2. The computational complexity of the data mining algorithms should be manageable, so that they can be used as exploratory analysis tools by the domain experts. We envision a successive refinement process, where the expert request a sequence of experimental runs, each built from their observations and interpretations from previous results generated by the algorithms, in a way that they can interpret the causal relations between faults and monitors, and discover the dependence among the monitors for different fault situations. The expert may also consider different analysis scenarios to estimate methods for increasing the accuracy (while reducing false positives) in the diagnostic reasoner.

Having established this framework, we stay within the Bayes net paradigm, and add additional criterion that the models derived by applying the data mining algorithms have similar structure and correspondence with the initial expert-supplied reference models.

4 Data Mining with Tree Augmented Naive Bayesian Networks

The choice of the data driven techniques to apply to a particular class of problems is very much a function of the nature of the data and the problem(s) to be solved using the data. For example, using data we can systematically test and relax the independence assumptions employed in the reference model, especially if it is useful for diagnosis. There are several interesting alternatives, but one that fits well with our reference model structure is the Tree Augmented Naive Bayesian (TAN) Method (Friedman et al., 1997) . The TAN structure is a simple extension of the Naive Bayes network. Like Naive Bayes, the Root node, corresponding to one or more fault modes, is casually connected to every evidence (monitor) node. In addition, the TAN structure relaxes the assumption of independence between the evidence nodes, and allows most evidence nodes to have a second parent, which can be a related evidence node. This maintains the directed acyclic graph requirements and produces a more nuanced tree that captures relationships among the monitors. Generation of this structure is not as computationally expensive as a general Bayesian network.

An example TAN structure is illustrated in Figure 2. The root node, labeled class, is the fault hypothesis under consideration. The other nodes represent supporting evidence for the particular fault hypotheses. In this particular structure, Rolltime, associated with the shutdown phase of the aircraft is the root observational node. Dependencies among some of the monitors, e.g., Rolltime and dipEGTC, are captured as additional causal links in the TAN structure.

The TAN Structure can be generated in several different ways that includes (1) a *greedy search* with the constraint that *illegal edges* (i.e., a node having more than one parent from the evidence nodes) are disallowed(Cohen, Goldszmidt, Kelly, Symons, & Chase, 2004); and (2) a *Minimum Weighted Spanning Tree* (MWST) approach that builds a minimum spanning tree to capture the dependencies among monitors,

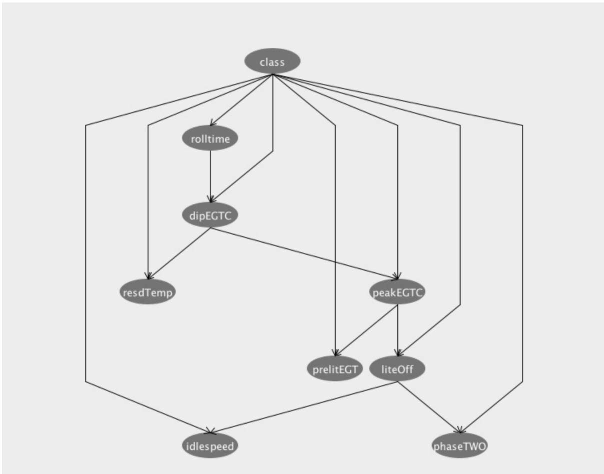

Figure 2: Example TAN Structure

and then connects the Root (fault mode) to all of the monitor nodes (Friedman et al., 1997). In either case, a decision has to be made about the monitor node to use as the observational root node in the derived tree structure. The derived TAN structure is static, i.e., it does not include temporal information explicitly.

A standard algorithm (e.g., Kruskal's algorithm(Kruskal, 1956)) is applied to generate the MWST. The values that are used as the edge weights all utilize a form of a log-likelihood, such as a Bayesian value (Chickering, Heckerman, & Meek, 1997) or a Bayesian Information Criterion (BIC) (Schwarz, 1978). If the values are discrete (either naturally or through a discretization process), the use of the Bayesian likelihood metric is preferred. This is a simple metric that can quickly calculate the likelihood that a variable is dependent on another. If the values are continuous, then the BIC is better since it deals with continuous distributions (like a Gaussian Normal) more easily. The values are calculated by every pair of evidence nodes and the order of the nodes matter, since the graphs are directed. This is typically stored in a matrix, and then Kruskal's algorithm is used to construct the tree using a simple search algorithm.

The MWST version of this algorithm is implemented in the data mining toolkit called Weka (Hall, Eibe, Holmes, Reutemann, & Witten, 2009) It does not handle continuous features, and instead uses a discretization algorithm which looks to bin each of the features into sets that best discriminate among classes. This produces better classifiers, but it may create very fine splits for features that results in excessive binning(thus building very large conditional probability tables).

Sensor	Notes
Altitude	real number; unit is feet
Mach Number	Real number, the unit is Mach
Throttle Angle	Real number, measured in degrees
Fuel Flow	Real number, measure in percent
Stall Margin of HPC	CI
Stall Margin of LPC	CI
Stall Margin of Fan	CI
Temperature of High Pressure Turbine Outlet	Real number, measured in Centigrade
Temperature of the Fan Inlet	Real number, measured in centigrade
Temperature of the Low Pressure Turbine Outlet	Real number, measured in centigrade
Pressure of the Fan Inlet	Real number, measured in PSI
Physical Fan Speed	Real number, measured in RPM
Physical Core Speed	Real number, measured in RPM

Table 1: Sensor values and Monitors (Conditional Indicators) for the CMAPS-S Engine Data

5 The CMPAS-S Data

The CMAPS-S data set is generated from a simulator developed at NASA's Glenn Space Center (Frederick, DeCastro, & Litt, 2007). The engine simulator takes into account the wear and tear on a turbine engine over multiple flights, and it can produce data for a number of sensors for climb, cruise, and descent modes of operation. The simulator parameters can be set to run in nominal and faulty modes of operation.

As a first step, we select appropriate sensor measurements and transform them into a sequence of monitor values for the data mining task. Since the reference model structure and the reasoner do not directly include temporal information, the data is separated into the different modes of operation. For this study all of the data for fault analysis was extracted from the cruise mode of operation. In this mode, most sensor values remain at about the same level, except for the measurement noise. Therefore, for this study each flight was represented by a vector of monitor values, and the entire data was made up of n data points corresponding to n flights, and each flight vector included a number of condition indicators (CIs). Table 1 shows the different features in the CMAPS-S data set. Some features are marked as a "condition indicator"(CI), which is a term for complex features that can be used to indicate when an engine is experiencing abnormal behavior. A threshold on these values would produce the *health indicator* (also called a diagnostic monitor, DM) that a reference model would relate to a fault mode. These features are used in lieu of a reference model since this data didn't come with one explicitly. The rest of the features are not the complete set from the data, but represent the sensors and thus features that would be most likely available in the data of other complex systems of this nature. These would be the features initially added to see if the algorithms could incorporate them successfully. A condition indicator is either a single sensor measurement set, or a combination of multiple sensor values.

The CMAPS-S data was generated in a way that the fault(s) and their time of introduction was known, so it was easy to assign the *nominal* and *faulty* labels for each data stream. The CMAP-S data models three faults: (1) a fan fault (Fan), (2) a High Pressure Compressor fault (HPC), and (3) a High Pressure Turbine fault (HPT). The data with its many faults allows for several possible models. One could construct different models for the three different faults with each model differentiating between the fault in question and a nominal mode. It could also be treated as a multi class learning problem, and build one model that attempts to distinguish between different faults and nominal operations. This multi-fault scenario is useful, as the model built may produce insight on how to differentiate between several ambiguous faults as they appear in the reference model. These different models are all worth building and the CMAPS-S data makes it easier to look at these combinations than real data that may not have as much faulty data that covers multiple faults.

6 Experiments

To evaluate the ability of our data mining techniques to improve the reference models, we have conducted a set of experiments using the simulated data from the CMAPS-S engine system to establish whether the TAN-based model produces a better diagnostic classifier than a reference model that is implemented as Naive Bayesian Classifiers(our experience indicates that most expert models are in the form of a Naive Bayes Classifier). Our experiments compare the performance results of the Naive Bayes versus the TAN models.

In the CMAPS-S data, there are several features that represent advanced sensors. These may not be available in other datasets for a variety of reasons (i.e., sensors may not exist on systems from with these datasets were created; the ability to build complex features may be limited). The first experiment uses the feature set defined in the baseline reference model, and extracts a classifier structure by running our machine learning algorithms. The next experiment adds additional sensors that are not conditional indicators, to see if using these sensors can improve diagnostic accuracy while reducing false alarms.

A system study of the performance of the algorithms requires running of n-Fold Cross Validation experiments. Dividing the data into n equally sized and distinct sets of samples, each with the balance of classes maintained like the original set allows for the creation of $n-1$ training sets with the last set being held out as the test set. This can be done n times, and the metrics values generated are then averaged over each of the n runs. This experimental style helps test the robustness of the classifier and keeps the metrics from being overly optimistic or pessimistic depending on the random construction of one hold out set. The experiments are broken down by the different types of fault in the CMAPS-S, as well as the multi fault case, where one model is built to distinguish between multi fault modes as well as the nominal case.

6.1 Experimental Results

The data generated for the experimental study included the three faults discussed before, and the analysis was conducted in the cruise mode with the aircraft flying at an altitude of 35,000 feet. The data mining algorithms were run to derive individual models for the three single fault modes, as well as a combined model of all the three faults. Tables 2 and 3, summarize our experimental results in terms of the accuracy metrics, i.e., overall accuracy (Acc), false positives (FP), and false negatives (FN).

The Naive Bayes model with only the CIs is considered representative of a reference model for analysis of core engine anomalies. A TAN structure including new causal relations results in a better reference model. The results in Tables 2 and 3 demonstrate the higher accuracy results for the TAN Structure for the FAN Fault and the multi-fault classifier. Their superior performance shows that even with a small number of features(3), introduction of two new causal links, the results improved considerably(99.4% to 67.9% for the Fan and 97.4% to 82.1% for mutli-fault). Figure 3 shows the representative TAN used in the multi-fault scenario. The CI corresponding to stall margin for the Low Pressure Compressor provided the best discrim-

	Fan			HPC			HPT			All THree		
	Acc	FP	FN	Acc	FP	FN	Acc	FP	FN	Acc	FP	FN
Naive Bayes Network	67.9	15.4	36.7	71.4	0	35.3	94.2	0	9.3	82.1	15.5	19.6
TAN	99.4	0.4	0.7	80.8	36.7	0	94.7	8.9	2.9	97.4	1.1	3.8

Table 2: Cruise Mode: Model with Only Conditional Indicators

	Fan			HPC			HPT			All THree		
	Acc	FP	FN	Acc	FP	FN	Acc	FP	FN	Acc	FP	FN
Naive Bayes Network	68.8	12.5	49.5	72.9	0	56.7	93.8	3.6	9.9	84.9	1.1	23.2
TAN	99.8	0	0.4	87.96	23.0	0	96.6	5.4	0.5	98.0	0.8	0.7

Table 3: Cruise Model: Model with Conditional Indicators + Sensor Measurements

inating evidence between different faults when only conditioned by the class variable. For the single fault classifiers, the Fan and HPC TANs outperformed the Naive Bayes, but the HPT classifier provided minimal improvement. The HPT Classifier seems to require a simple classifier and both models achieved over 90% accuracy. The HPC fault however with only fault indicators was the lowest performing set. Although the TAN did better by over 8%, this would indicate that the reference model for the engine may not be able to catch this fault, particularly from cruise data. This makes it an interesting case for further analysis.

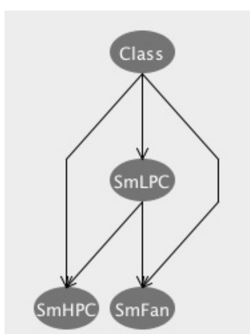

Figure 3: TAN Model for Multi-Fault Scenario with Only Conditional Indicators

For the second experiments where we consider additional sensors, there is an improvement in the accuracy numbers for all of the TAN models. This is highlighted by the HPC fault scenario, which was problematic in experiment 1, but the accuracy increased significantly. This improved the False Positive rate, while not increasing the corresponding false negative metric. This change made it significantly better than its Naive Bayesian counterpart as well as the over both models in the first experiment. This improvement without a negative cost to the error rates is true for the TAN models across all scenarios. When compared to the Naive Bayes models, the TANs improve not just with respect to the Naive Bayes learned with the same features, but also with the original models themselves. This is magnified by the additional information having a small negative impact in a few cases of the Naive Bayes models. Fault detection and isolation is already efficient with only the CIs available to the Naive Bayes classifier. This new information provided an advantage to the TANs where additional causal relations and information improve diagnostic accuracy.

Looking at the HPC scenario with the new information, Figure 4 displays the model structure generated in that scenario. In this case, the HPC accuracy improved with the TAN to 88% compared to the original HPC TAN at 80.8%, the Naive Bayes Model using the additional sensors at 72.9%, and the original Naive Bayes model at 71.4%. The accuracy results clearly indicate: (1) additional sensor information increases diagnostic accuracy and (2) Switching from a Naive Bayes to a TAN model improves diagnostic accuracy.

This improvement can be examined visually in 4, where in place of the three CIs, the Mach Number sensor becomes the observational root node. The new causal structure, captured in Figure 4 shows the Fuel Flow sensor as a parent to two of the CIs. Network structures such as the one for the HPC fault explicitly illustrate how additional sensor information can be included to enhance the accuracy of the reference model. In general, the new causal relations suggested can be examined by a domain expert who in turn can construct new and improved indicators to use in a reference model. The results generated by these data driven models can provide numbers on how the new information can improve the accuracy of the diagnoser, and how it may impact the error rates.

7 Conclusions and Future Work

The results on experiments conducted over the CMAPS-S data illustrate the promise of the methodology and process we have been developing. To fur-

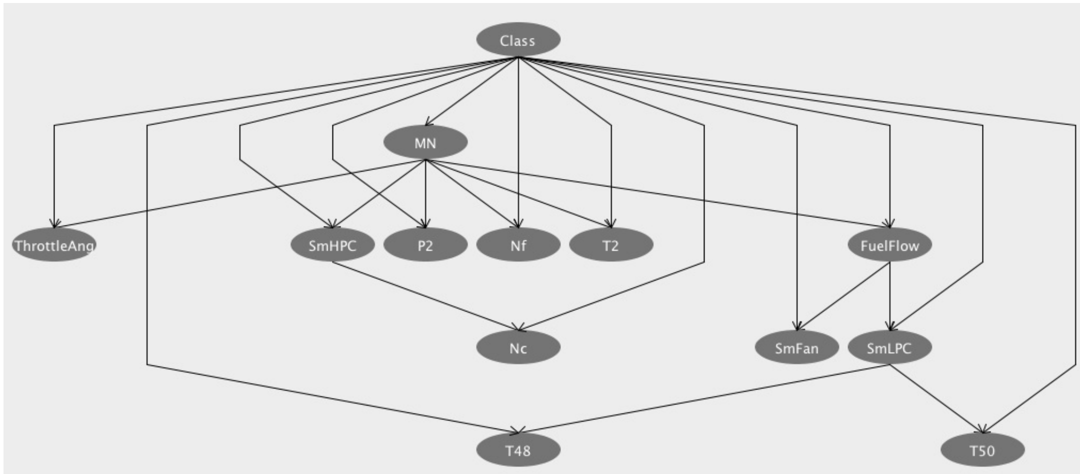

Figure 4: TAN Model for HPC Scenario with Conditional Indicators and Extra Sensors

ther validate our work, we have identified a number of directions and tasks we need to pursue as we move forward in this project.

- Using a Naive Bayes Classifier is an approximation of a hand built reference model. We would like to construct a model with a domain expert, implement a simple reasoner and test against the data driven models our data mining algorithms produce. Then we compare the new structure generated to enhance the existing reference model by adding new DMs.
- Simulation systems, such as CMAPS-S study particular systems, like the core engine functions in greater detail than any information that can be derived from sensors and monitors in current aircraft configurations. We are looking to develop methods by which detailed simulation data may be combined with actual aircraft flight data to carry on extensive analyses of diagnostic and prognostic events and their propagation through the aircraft system.

Acknowledgements

This project has been supported by NASA NRA NNL09AD44T

References

Chickering, D. M., Heckerman, D., & Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. In *In Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence.* Morgan Kaufmann.

Cohen, I., Goldszmidt, M., Kelly, T., Symons, J., & Chase, J. S. (2004). Correlating instrumentation data to system states: a building block for automated diagnosis and control. In *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6* (pp. 16–16). Berkeley, CA, USA: USENIX Association.

Frederick, D., DeCastro, J., & Litt, J. (2007). *Users Guide For the Commercial Modular Aero-Propulsion System Simulator* (Tech. Rep.). NASA.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning*, *29*, 131–163.

Hall, M., Eibe, F., Holmes, B., Geoffrey amd Pfahringer, Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, *11*(1), pp. 10-18.

Honeywell. (2010). Vehicle Integrated Prognostic Reasoner. *NASA Contractor Report to appear*, *NNL09AD44T*.

Kruskal, J., Joseph B. (1956). On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, *7*(1), pp. 48-50.

Schwarz, G. (1978). Estimating the Dimension of a Model,. *Annals of Statistics*, *6*.

Spitzer, C. (2007). Honeywell Primus Epic Aircraft Diagnostic and Maintenance System. *Digital Avionics Handbook*(2), pp. 22-23.

Witten, I., & Frank, E. (1999). *Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations.*