DISCOVERING SYSTEM HEALTH ANOMALIES USING DATA MINING TECHNIQUES [1]

Ashok N. Srivastava, Ph.D.
Discovery and Systems Health Technical Area
Intelligent Systems Division
NASA Ames Research Center
Mail Stop 269-4, Moffett Field, CA 94035
ashok@email.arc.nasa.gov

ABSTRACT

We discuss a statistical framework that underlies envelope detection schemes as well as dynamical models based on Hidden Markov Models (HMM) that can encompass both discrete and continuous sensor measurements for use in Integrated System Health Management (ISHM) applications. The HMM allows for the rapid assimilation, analysis, and discovery of system anomalies. We motivate our work with a discussion of an aviation problem where the identification of anomalous sequences is essential for safety reasons. The data in this application are discrete and continuous sensor measurements and can be dealt with seamlessly using the methods described here to discover anomalous flights. We specifically treat the problem of discovering anomalous features in the time series that may be hidden from the sensor suite and compare those methods to standard envelope detection methods on test data designed to accentuate the differences between the two methods. Identification of these hidden anomalies is crucial to building stable, reusable, and cost-efficient systems. We also discuss a data mining framework for the analysis and discovery of anomalies in high-dimensional time series of sensor measurements that would be found in an ISHM system. We conclude with recommendations that describe the tradeoffs in building an integrated scalable platform for robust anomaly detection in ISHM applications.

INTRODUCTION

Modern ISHM systems contain hundreds or thousands of sensors producing both discrete and continuous measurements. The union of all the sensor signals at a given time t can be considered the observed state of the system. The entire record of the sensor measurements represents the evolution of the system through time. In this paper we focus on the situation where the sensor measurements are producing discrete and continuous signals. Discovering anomalies in continuous systems has been extensively treated in the data mining and statistics literature [5, 6]. We assume that the observed system evolution can be functionally described by the following equations:

$$\mathbf{h}_t = \Phi(\mathbf{h}^*_{t-1}) \tag{1}$$

$$\mathbf{x}_t = \Psi(\mathbf{x}^*_{t-1}, \mathbf{h}^*_{t-1}, u_t) \tag{2}$$

We assume that the function Φ determining the evolution of the system state $\mathbf{h}_t$ is unknown. The hidden system state $\mathbf{h}_t$ is assumed to be unobservable through the sensor measurements in $\mathbf{x}_t$We also discuss two situations, one where the state $\mathbf{h}_t$ is observable, and one where it is assumed to be hidden. The vector $\mathbf{x}$ is an N dimensional observed binary state vector, and $\mathbf{x}^*_{t-1}$ is the entire history of the observed state vector: $\mathbf{x}^*_{t-1} = [\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{t-1}]$. The quantity u_t is the observed input to the system. Because we assume that we do not know the function Ψ, we cannot rely on it to help us determine whether or not the observed vector $\mathbf{x}_t$ is anomalous.

The problem that we address in this paper is to develop a method to discover whether or not the current observed state $\mathbf{x}_t$ is anomalous or not based on the observed history of the system, as well as replicates of the system behavior. Thus, let $\mathbf{X}$ be a $T \times N$ matrix whose tth row $X(t, :)$ contains the values of the N binary sensors at time t during a single run of the system. Likewise, the nth column $X(:, n)$ contains the time series representing

[1] This paper will be published in the Proceedings of the Joint Army Navy NASA Air Force Conference on Propulsion, Charleston SC, June 2005.

the time-ordered sequence of states of the nth sensor. Different replicates of the system are indexed by l and are represented by different $\mathbf{X}_l$, for $l = 1 \ldots L$.

We make two assumptions regarding the sensor measurements in order to develop the ideas in this paper:

- the times in each $\mathbf{X}_l$ have all been normalized so that $t = 0$ (i.e. the first row of $\mathbf{X}_l$) corresponds to the start of the system.
- all replicates of the system l under consideration correspond to the the system evolving under similar operating conditions.

In order to ground this example to a real system, we consider the situation where there are $N \approx 1000$ different binary switches in an aircraft cockpit [2]. We assume that we have a recording of length $T \approx 1000$ of the switch settings, sampled at uniform intervals. Note, however, that each flight may have a different value of T, since different flights have different durations, etc. As the pilot maneuvers the airplane through its various phases of flight, the pilot flips these switches. We assume that we have $L \approx 10,000$ different flights recorded.

Clearly, the vast majority of the flights in both civilian and military aircraft are such that the pilot flips the appropriate switches at the appropriate times in the appropriate *order*. Due to standard operating procedures (SOPs) and extensive training, one would expect the deviation between different flights to be small for switches that govern the operation of major systems in the aircraft.

This paper addresses the problem of identifying anomalies in the sequences of states of the switches that are observed. The anomaly is only defined with respect to the sample of L different flights, or replicates, since we do not have an explicit model of how the system evolves through time, i.e., we do not, and cannot know Φ or Ψ in Equations 1 and 2.

Note that the data sets involved in this simple example are extremely large. For the values given above, the data set $D = \{\mathbf{X}_1, \cdots, \mathbf{X}_L\}$ is on the order of 1.25 gigabytes in size. In the event that the data set D is refreshed daily, we have an incredible escalation in data set size and associated storage and maintenance costs. The methods that we develop need to be able to scale to the massive data sets without requiring supercomputing power and significant storage media. With these assumptions we would like to build a model $P(\mathbf{X})$ from a data set $D = \{\mathbf{X}_1, \cdots, \mathbf{X}_L\}$. The data $\mathbf{X}_l$ arise from different airlines flying the same route using the same airplane. Given that we are going to have a large volume of data any model for $P(\mathbf{X})$ must be able to learned and probed quickly. [3]

TWO MODELS OF THE DATA GENERATING PROCESS

We next describe two models of the data generating process that obey Equations 1 and 2, but that have very different statistical characteristics, and therefore very different implications on the ISHM problem. The first model assumes that the data are generated by a process where each observation is probabilistically independent of each other observation, and that the state $\mathbf{h}_t$ is fully observed. Since the state vector is fully observed, we can assume that it is part of the observation vector $\mathbf{x}_t$ without loss of generality. The independence can be expressed as the following expression, for all t, t', n, n':

$$P(\mathbf{x}_t(n), \mathbf{x}_{t'}(n')) = P(\mathbf{x}_t(n))P(\mathbf{x}_{t'}(n')) \tag{3}$$

This data set represents a simple 'benchmark' or 'baseline' model which we can use to evaluate the performance of the algorithms that we describe in this paper. In order to generate data that matches this assumption, we create a $T \times N$ data set where $(T = 1000, N = 500)$ using binomial random variables with parameters $(n = 1, p = 0.3)$. We will refer to this data set as the *baseline data set*. A key aspect of this data set is that we

[2] Note that switches with m multiple settings can always be converted to m binary signals.

[3] As the data will be accumulated daily it might also be important that the model can easily be updated incrementally.

assume that the sensor measurements are a full representation of the state of the system. Thus, methods that are successful in finding faults in this data set would be those that do some sort of envelope detection either on the sensor measurements themselves or a transformed representation of the measurements. This data set will illustrate the properties of envelope detection algorithms that are commonplace in the ISHM literature.

The second model of the data generating process is much more complex, but it allows for unobserved system anomalies and dynamically evolving states and observable vectors. We will refer to this data set as the *dynamic data set*. The model is based on the Hidden Markov Model, which is widely used in speech recognition [7] and has also been discussed in the IVHM/ISHM literature [1, 3]. We next briefly describe the Hidden Markov Model and its applications to IVHM.

The HMM is a dynamic model consisting of H hidden states that are assumed to be not directly observable, with S observation symbols associated with each state. The model has an initial probability distribution π which is an $S \times 1$ vector where π_i is the probability that the system begins in state i. The HMM starts in an initial state according to the distribution π and then the hidden state h_t moves to the next state based on a Markov transition matrix A. The matrix $A_{ij} = P(h_{t+1} = j|h_t = i)$ which gives the probability distribution of the next hidden state given the current state. While the system is in state i it is assumed to emit an observation vector $\mathbf{x}_t$ according to the distribution $B_{jk} = P(\mathbf{x}_t = k|h_t = j)$.

Notice that the HMM assumes a discrete state space for the hidden variable $\mathbf{h}_t$, and that it is assumed that the observation vector $\mathbf{x}_t$ is also discrete. In our case, the number of symbols $S \approx 2^N$ where $N = 500$. Thus, we need to perform a preprocessing step to identify a small number of representative states. This step is as much art as science, and is usually performed through the use of a clustering algorithm. Banerjee et. al. [2] have given an excellent formulation to the problem of clustering high dimensional data based on the von Mises Fischer distribution. We use their algorithm along with standard clustering methods such as the k-means algorithm for grouping observations into symbols. In the event that the state vector $\mathbf{x}$ contains both discrete and continuous parameters, the same clustering methodologies can be applied to generate the small number of representative states.

The implications of the HMM formulation for ISHM problems in interesting. The HMM allows for modeling situations where we assume that the system state is evolving according to an unobserved set of dynamics defined by the Markov transition matrix. As the system evolves, the state may go into one or more 'failure modes' that not directly observable at the sensors. However, based on the distribution of observed sensor measurements, it may be possible to determine a failed state. This is the avenue which we explore in this paper. In the simulations for this paper, we chose $H = 6$, $S = 6$, and the probability matrices A and B such that the system can fall into a failure mode (state 6) with probability 0.05. When the HMM construct is applied to real data, these values of H and S will generally be much larger, and the transition probabilities to the failed state will be much smaller.

As noted above, the data set size is potentially very large. For the systems described in this paper, the data is provided as the set of matrices $D = \{\mathbf{X}_1, \cdots, \mathbf{X}_L\}$. The first observation that can be made is that the matrices $\mathbf{X}_l$ can be converted to all highly sparse matrices, meaning that most of the data is all not changing at any given time. Thus, we can convert the data into a very compact representation as follows:

1. Record the initial values for all binary variables $\mathbf{x}_0$. This vector represents the initial state of the system at time 0.
2. Record only those variables that transition from the initial state to the next state. Since the variables are all binary, this can be recorded simply as the index number of the switch that changed.
3. If needed, record the time at which the transition occurred.

This representation of the data reduces the data set size significantly. Our experiments indicate that for real data sets from approximately $10,000$ flights, the data set size drops to approximately 1% of the original size without

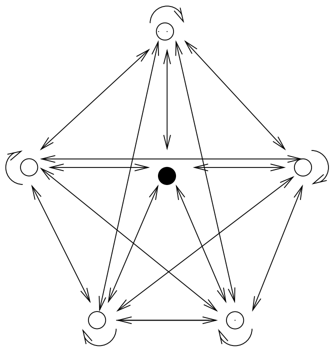

Figure 1: This figure depicts the state model of state transitions from normal operation (clear circles) to the failed state (solid circle in the center of the diagram). Notice that the model allows for the system to move from a failed state back to normal operation. This models intermittent problems which can arise in complex systems. For generality, we have included bi-directional arrows and a fully connected graph. In many cases, systems can only move into a failed state but not back to normal operation. Prior system information such as the fact that states i and j cannot be connected can be included in the model.

any loss of information. However, this data representation imposes certain constraints on the underlying algorithms. We will discuss this tradeoff in the next sections.

MODELING APPROACH

At opposite extremes there are two potential simplifications to make regarding this problem. We could focus either on the temporal aspects of the problem (non-stationary behavior across rows), or on the correlation aspects between the sensors (stationary behavior across columns). The most general kind of model, where we look at correlations (or higher order analogues) between $X(t, n)$ and $X(t', n')$, we discard as being too complex as a starting point. To obtain initial insight quickly we focus on a simple model for the time dependence and a simple model for the stationary behavior. However, our proposed approach can be extended to relax these simplifying assumptions.

In the simplest approximation we completely ignore the time dependence, and build a model for the likelihood of any particular row $\mathbf{x}$ of $\mathbf{X}$. Different rows (i.e. different times) are assumed to be independent so that $P(\mathbf{X}) = \prod_t P(\mathbf{x}_t)$ where t ranges over the rows of $\mathbf{X}$ and $\mathbf{x}_t = \mathbf{X}(t, :)$. Thus, we simplify things to the point of only having to build a model for $P(\mathbf{x})$ (i.e., for a single row).

INDEPENDENT SENSORS INDEPENDENT OF TIME

The most naive model for an observation $\mathbf{x}$ assumes independence between columns (sensors), i.e.,

$$P(\mathbf{x}) = \prod_{n=1}^{N} P(x(n)) \tag{4}$$

where $x(n)$ is the nth component of $\mathbf{x}$. For binary data a Bernoulli assumption can be used to model the value of the bit $x(n)$ so that

$$P(x(n)) = \rho_n^{1-x(n)}(1-\rho_n)^{x(n)} \tag{5}$$

where ρ_n is the probability that the nth bit is zero. The maximum likelihood estimates for the parameters ρ_n are obtained simply from frequency counts down the columns $\mathbf{X}_l(:, n)$ for all l in the data set D, and dividing by the

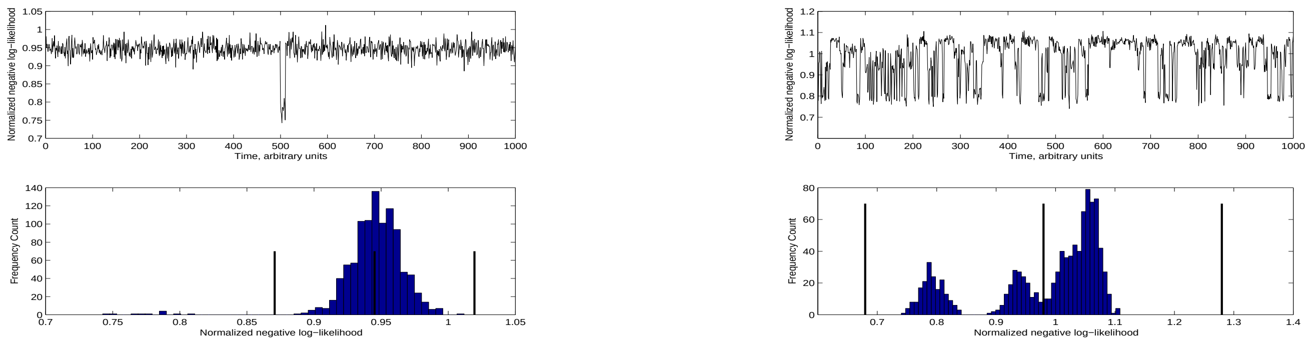

Figure 2: We computed the likelihood of each observation for both the benchmark data set and the dynamic data set as a function of time. A failure mode in the benchmark data set is apparent in the drop in the likelihood of the points around time step 500. These are shown in the top panel of plots for both data sets respectively. Notice that most of the data lies within the ± 3 sigma confidence interval of the mean likelihood as indicated in the lower panel. The unimodal characteristic of the benchmark data set is clear in the lower left hand plot. The multimodal characteristic of the observation sequence is also apparent in the lower right hand plot. The observations are not indicative of a failure state.

total number of rows in D.

We constructed this model based on the data and have plotted key statistics in Figure 2. The plot indicates the negative log-likelihood of the data given the Bernoulli model in Equation 5. The negative log-likelihood is computed by taking the negative log of Equation 4.

We introduced a 'failure' into the benchmark data set at time step 500 in the left-hand side of Figure 2 in order to determine the way the model would show the likelihood of the failure. The failure was introduced by inserting vectors with a higher mean value than that specified in the overall model. Those points are caught by the likelihood model, and have lower likelihood values compared to the rest of the time series. Thus, an envelope detector would easily catch this type of system failure.

The right-hand side of Figure 2 shows the likelihood of the observations given the Bernoulli model above for the dynamic system. In this example, notice that the likelihood function has at least three modes (lower right-hand side of the figure). The system stays well within the 3σ envelope that we implemented. Thus, this envelope detector does not capture the fact that the system is actually in a failure mode a significant fraction of the total observation time.

This plot is similar to those found in the change-point detection or the process control literature, in that it shows the likelihood of observations as a function of time. Observations that fall outside of the "expected" bounds can be tagged for further observation. As noted before, this model completely ignores the potential relationship between the binary sensors or switches, as indicated in the in Equation 4.

DEPENDENT SENSORS INDEPENDENT OF TIME

Another approach that is often taken to analyze such data is based on the singular value decomposition (SVD) of the data set $\mathbf{X}_l$ [4]. This method works by factoring the matrix $\mathbf{X}_l$ into three matrices as follows:

$$\mathbf{X}_l = \mathbf{U}\Sigma\mathbf{V}^T \tag{6}$$

The matrix $\mathbf{U}$ is a $T \times N$ orthogonal matrix, Σ is an $N \times N$ diagonal matrix, and $\mathbf{V}$ is an $N \times T$ orthogonal matrix. SVD (which is closely related to principal components analysis) identifies the directions of maximum

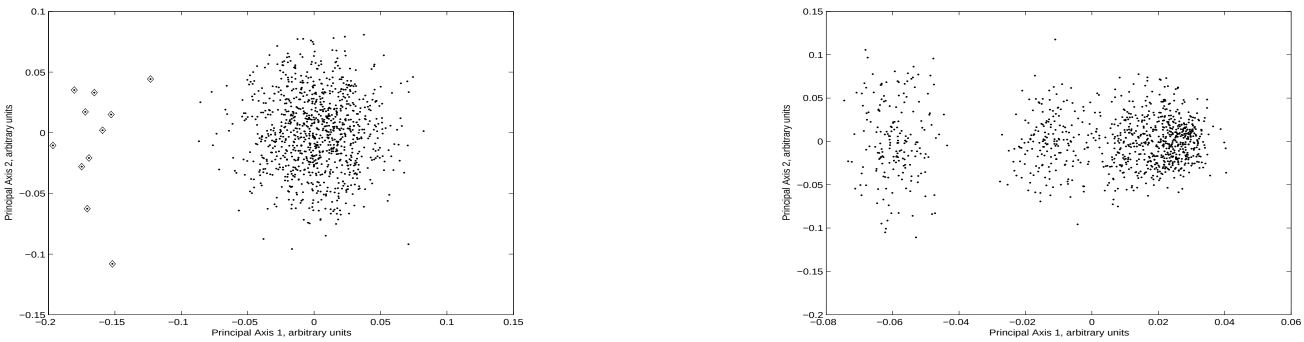

Figure 3: Using the same data sets as in Figure 1, we plot the two dimensional representation of the data points as projected onto the first two principal components. The data points corresponding to the 'highly unlikely' values above fall outside the modal region in this plot. These points are indicated by diamonds. The benchmark data set (left) clearly shows the failure modes. However, this transformation does not show any signs of failure for the dynamic data set.

variation in the group of points defined by the rows of the data matrix. These directions can be shown to be the eigenvectors of the associated covariance matrix. Once the top m eigenvectors are identified (these correspond to those with the largest m eigenvalues), the observation vectors are left-multiplied with the eigenvectors. This results in an m dimensional representation of the observation vectors, where $m < N$. The parameter m is chosen in order to explain the maximum amount of variation in the data with the minimum number of eigenvectors.

It has been shown that plotting the first two columns of the unitary matrix $\mathbf{U}$ can give diagnostic information about the individual data points in the original data matrix [8]. We next demonstrate the effect of analyzing these binary streams using SVD. Figure 3 shows the distribution of points in $\mathbf{X}$ for the two benchmark and the dynamic data sets. The outlying points in the left hand side of the figure correspond to the relatively unlikely events that occur near time step 500 in the likelihood time series.

These methods, while useful for analyzing high dimensional time series, do not capture the important dynamic aspects of the system. Instead, they treats each row in the time series matrix as an independent observation. More over, they amount to building an envelope detector in either the original space of the time series or a transformed space, as in the case of the SVD. The right-hand side of Figure 3 shows the SVD results for the data generated by the dynamic system. Again, as in the case of the previous method, the system captures at least three modes of operation based on the observed signal. However, it is unable to identify any failures, because they are hidden in the dynamics based on Equations 1 and 2. Through these two examples, we have motivated the need for dynamic statistical models to help detect potential hidden failure modes.

DYNAMIC MODELS

In our approach regardless of how $P(\mathbf{x})$ is modeled time dependence is addressed by building a model for $P(\mathbf{X}) = P(\mathbf{x}_1, \mathbf{x}_2, \cdots)$ for all rows (times). One reasonable modeling possibility here relies upon the Markov assumption where we assume that $\mathbf{x}_t$ is conditionally independent of all previous rows $\mathbf{x}_{t'}$ (with $t' < t$) except for x_{t-1}. In this case this simplification would give $P(\mathbf{X}) = P(\mathbf{x}_1) \prod_{t>1} P(\mathbf{x}_t|\mathbf{x}_{t-1})$. A naive Hidden Markov Model (HMM) is inappropriate to learn this model as any given row vector can assume 2^N possible values. Thus, we would first have to reduce the number of features (perhaps by clustering with a mixture model) describing $P(\mathbf{x})$ and then build an HMM over these features.

We briefly describe a method to cluster high-dimensional data sets like those described in this paper into a set of categories that represent a smaller class of discrete observations. There are several approaches to clustering, including Gaussian Mixture Modeling (GMM) and the k-means clustering algorithm. Both of these algorithms make Gaussian assumptions about the underlying distribution of the data. The data generating process for the ISHM observation signals described in this paper are far from Gaussian. They are high-dimensional, sparse, discrete signals.

Empirical studies have shown that for high dimensional sparse data sets, the cosine measure of similarity between two vectors is a better measure of similarity than the Euclidean distance. A recent paper [2] developed the mathematics to perform clustering using the cosine measure of similarity. The cosine distance implies that the data are generated according to the von Mises Fisher distribution. We follow the formulation in [2] closely:

$$P(\mathbf{x}|\Theta) = \sum_{c=1}^{C} P(c)P(\mathbf{x}|\theta_c) \tag{7}$$

In this case, we assume that the vectors $\mathbf{x}$ have been normalized to unit length and C is the total number of clusters in the model. For p dimensional data vectors, we have the von Mises Fisher (vMF) distribution:

$$P(\mathbf{x}|\mu,\kappa) = c_p(\kappa)\exp(\kappa\mu^T\mathbf{x}) \tag{8}$$

where μ is a unit vector corresponding to the mean of the distribution and $\kappa \geq 0$ is the measure of dispersion. The constant $c_p(\kappa)$ is given by:

$$c_p(\kappa) = \frac{\kappa^{(p/2)-1}}{(2\pi)^{(p/2)}I_{(d/2-1)}(\kappa)} \tag{9}$$

where $I_{(r)}(\kappa)$ represents the modified Bessel function of the first kind of order r. With the vMF distribution as defined above, Banerjee et. al. (2003) derive the Expectation Maximization algorithm to optimize a mixture of vMF distributions [2]. Preliminary results indicate that this algorithm has superior performance on high dimensional clustering problems compared to the k-means algorithm, showing substantial improvements over existing algorithms.

We cluster each vector $\mathbf{x}_t$ into one of C clusters, choosing the most likely cluster (based on the values of $P(\mathbf{x}|\theta_c)$). Thus, the dimensionality of the problem is drastically reduced, from a search space of size 2^N down to a relatively small number C. These cluster memberships are used as input into the HMM training algorithm [4].

The HMM is trained using the Baum-Welch maximum likelihood parameter estimation algorithm [7]. This algorithm has been discussed in detail and is commercially available and will not be discussed here. The algorithm accepts the sequence of symbols (corresponding to the C cluster memberships), along with initial guesses of the transition probability matrix and the symbol emission matrix, and produces an estimate of the true transition and emission probability matrices. The sizes of these matrices also encode the number of hidden states in the model as well as the number of distinct symbols.

In the process of building the dynamical model, there are two key parts of the model where model information can be inserted into the system to guide the discovery of the underlying states. The first part is the state transition matrix, which gives the distribution of the next hidden state given knowledge of the current state. This distribution can be modified to encode system information. For example, knowledge that a system mode sequence is unidirectional (state i can go to state j but not to state k) can easily be encoded in the transition matrix. Modifications to the algorithm can be made to enforce such constraints over the transition matrices. The second part is the symbol emission matrix, which can be modified using similar principles as the state transition matrix.

[4]The training and testing algorithms used in this paper are available in Matlab. In order to demonstrate the properties of the dynamic model alone, we have used the true state information in these demonstrations.

Just as the envelope detection methods outlined earlier in the paper had drawbacks in being insensitive to dynamical anomalies, the dynamical model has drawbacks as well. These include:

- The HMM is a complex model with a large number of free parameters. The parameter estimation problem is complex and requires a significant amount of data to ensure a good quality estimate.

- The step of converting high-dimensional time series into a small set of symbols injects noise into the process that may complicate discovering the hidden state. If this process is done through a clustering algorithm, as described in this paper, the clusters memberships can have significant run-to-run variability.

- The hidden states may not be directly interpretable, and may not fully capture a hidden fault in the system.

The results of the algorithm are presented in Figure 4. The upper panel of the diagram shows the transition of the system from other states (labeled as ′0′) to the failed state (labeled as ′1′). This data set is designed to have an intermittent failure mode that is undetectable at the observation level (see Figures 2-3 for discussion). The lower panel shows the hidden state estimated by the Hidden Markov Model. This model has a 69% true-positive rate and a 71% true-negative rate. [5]

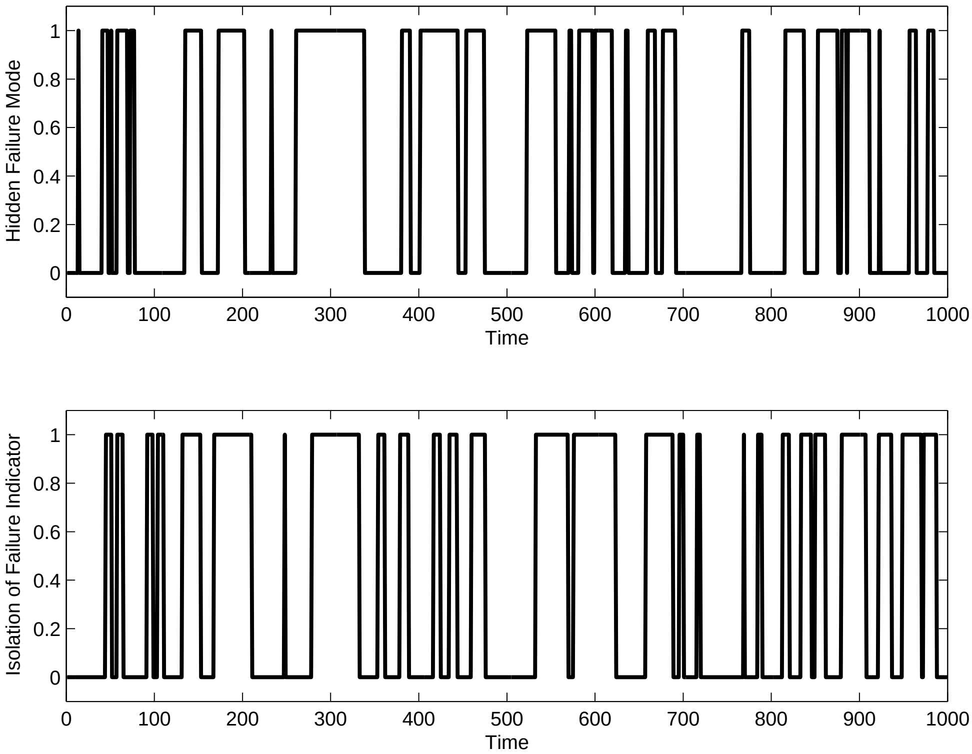

Figure 4: The top panel shows the transition of the system from a normal state into the failed state (labeled as ′1′). The lower panel shows the detection of the hidden state as it transitions into the failed state. The algorithm has a true-positive rate of 71%.

[5]The true-positive rate is defined as the percentage of times the model correctly predicts the hidden state as being in the failed mode. The true-negative rate is defined similarly.

DISCOVERY WORKBENCH FOR ISHM

In order to provide a framework for testing both the static time independent models based on envelope detection, as well as the dynamic modeling capabilities of HMMs and visualization methods, we are building a prototype system called the *Discovery Workbench for ISHM* that comprises methods to animate and visualize as well as analyze signals from large discrete and continuous sensor suites. An essential component of the system is for visualization of complex signals. However, for high-dimensional sensor signals, it is impossible to view the complex relationships between the signals.

Thus an important preprocessing capability for this system, or any system that handles and visualizes high-dimensional data, is to have methods to either reduce the dimensionality of the problem or to automatically filter 'interesting' from 'noninteresting' information. Currently, the SVD method of dimensionality reduction is available in the system, as well as methods to cluster high dimensional data to generate symbol sequences. More work needs to be performed to develop adequate filtering mechanisms that isolate sensor readings that are of interest that may be co-varying in time. Sensors embedded within complex system generate voluminous datasets whose analysis typically includes exploration, feature selection, feature detection and dimensionality reduction, model construction, and classification. To analyze time series and discover useful patterns, analysts must select a suitable method, and must fine-tune method parameters to obtain optimal results for a given application. We designed the discovery workbench to help analysts explore and fine-tune potential methods easily and rapidly. In order to ensure the scalability of the working system, we plan to build the system in Java with interfaces to traditional monitoring systems through standard interfaces.

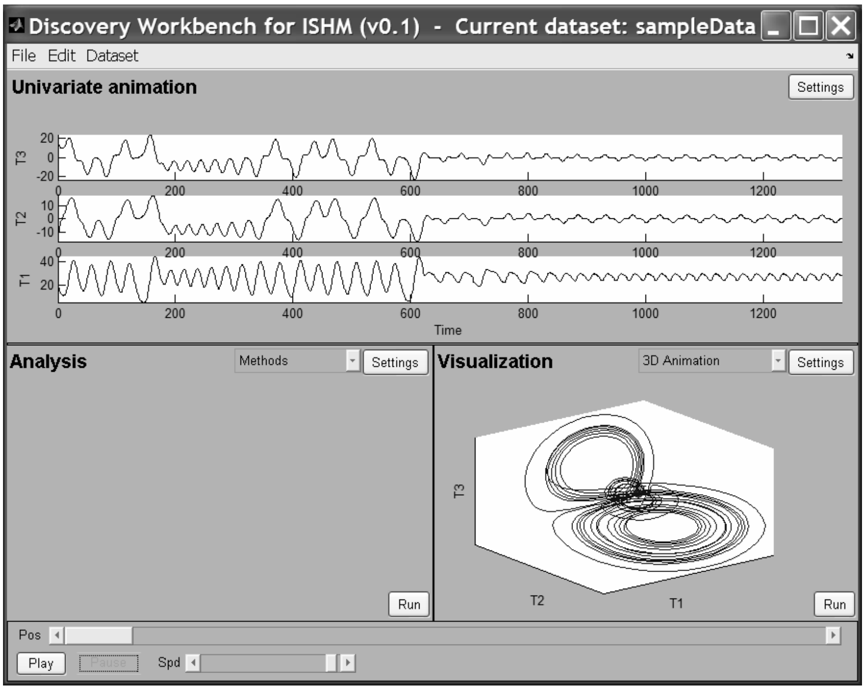

Figure 5: The Discovery Workbench for ISHM is a prototype that includes three functionality categories: *univariate animation* for examining individual series as they progress with time; *analysis*, for executing data mining and pattern recognition algorithms; and *visualization*, for examining multivariate time series and visualizing analysis results. The workbench is flexible in that users can add analysis and visualization methods easily by adhering to a simple interface specification. The workbench allows users to detach intermediate visualizations for viewing with subsequent results. The workbench is implemented in Matlab and runs on multiple platforms.

CONCLUSIONS

We have described the statistical properties of envelope detection methods as well as dynamic hidden Markov Models for applications in Integrated Systems Health Management. The envelope detection methods are statistically simpler than the dynamic models, but they assume that the failure mode is observable in the sensor suite. The dynamic models can allow for rapid discovery of failures that are not directly observable in the sensor suite. However, they suffer from being large and potentially very complex models. We motivated our work by discussing a large scale anomaly detection problem in the aviation domain and described a prototype workbench for the analysis, synthesis, and visualization of high-dimensional sensor signals.

ACKNOWLEDGEMENTS

The author thanks Bill Macready, Brett Zane-Ulman, Smadar Shiffman, Nikunj Oza, Dawn McIntosh, and Mark Schwabacher for valuable feedback and discussions. This work was supported by NASA's Aviation System Monitoring and Modeling Program Element under the Aviation Safety and Security Program.

References

[1] L. Atlas and G. Bloor, *An evolvable tri-reasoner ivhm system*, ISIS Vanderbilt Website (1999).

[2] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, *Generative model-based clustering of directional data*, 2003.

[3] K. R. Pattipati J. Ying, T. Kirubarajan and A. Patterson-Hine, *A hidden markov model-based algorithm for online fault diagnosis with partial and imperfect tests*, IEEE Transactions on SMC: Part C **30** (2000), no. 4, 463–473.

[4] I. T. Jolliffe, *Principal component analysis*, Springer, 2002.

[5] Eamonn J. Keogh, Selina Chu, David Hart, and Michael J. Pazzani, *An online algorithm for segmenting time series*, ICDM, 2001, pp. 289–296.

[6] M. Last, Y. Klein, and A. Kandel, *Knowledge discovery in time series databases*, 2001.

[7] L. R. Rabiner, *A tutorial on hidden markov models and selected applications in speech recognition*, Proceedings of the IEEE **77** (1989), no. 2, 257–286.

[8] D. B. Skillicorn, *Clusters within clusters: Svd and counterterrorism*, SIAM Workshop on Counterterrorism (2003).