A Bayesian state-space approach for damage detection and classification

A Bayesian State-space Approach for Damage Detection and Classification

Zoran Dzunic*[†], Justin G. Chen[†], Hossein Mobahi*[‡], Oral Buyukozturk**[§], John W. Fisher III*[¶]**

*Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology,
77 Massachusetts Avenue, Cambridge, MA 02139, USA
**Department of Civil and Environmental Engineering, Massachusetts Institute of Technology,
77 Massachusetts Avenue, Cambridge, MA 02139, USA
[†]PhD Candidate, [‡]Post-Doctoral Associate, [§]Professor, [¶]Senior Research Scientist

ABSTRACT

The problem of automatic damage detection in civil structures is complex and requires a system that can interpret sensor data into meaningful information. We apply our recently developed switching Bayesian model for dependency analysis to the problems of damage detection, localization, and classification. The model relies on a state-space approach that accounts for noisy measurement processes and missing data. In addition, the model can infer statistical temporal dependency among measurement locations signifying the potential flow of information within the structure. A Gibbs sampling algorithm is used to simultaneously infer the latent states, the parameters of state dynamics, the dependence graph, as well as the changes in behavior. By employing a fully Bayesian approach, we are able to characterize uncertainty in these variables via their posterior distribution and answer questions probabilistically, such as "What is the probability that damage has occurred?" and "Given that damage has occurred, what is the most likely damage scenario?". We use experimental test data from two laboratory structures: a simple cantilever beam and a more complex 3-story, 2-bay structure to demonstrate the methodology.

Keywords: Graphical models, Bayesian inference, structural health monitoring, state-space model, damage classification

1 INTRODUCTION

Structural inspection has been necessary to ensure the integrity of infrastructure for almost as long as structures have existed, ranging from informal subjective methods such as visual or hammer testing, to quantitative modern methods including ultrasound, x-ray, and radar non-destructive testing techniques. These testing methods are relatively intensive as they depend on the experience of the inspector and the time to inspect suspected damaged locations in the structure. Inspections are typically carried out periodically, however if additional sensors could be added to the structure such that some indication of where potential locations of damage might be such that they can be closely inspected, it would be useful for reducing the time and effort necessary for structural inspection.

Structural health monitoring (SHM) involves instrumenting a structure with sensors and deriving some information from the data they collect in order to determine if the structure has changed [1]. This change in the structure could then be attributed to some sort of damage that would be more closely investigated. In general, data is processed into features that may indicate these changes in the structure and in some cases statistical or probabilistic discrimination of these features are used to separate data collected from intact and changed structures [2]. Statistical methods are essential for being able to discriminate feature changes as a result of structural changes from measurement or environmental variability.

Bayesian inference is a probabilistic method of inference that allows one to form probabilistic estimates of certain parameters given a series of observations. The method can be used in a couple of different ways in SHM including model updating of structural parameters [3], monitoring by inferring structural parameters over time [4], and determining the optimal placement of sensors [5]. Bayesian inference can be used in either a model-based situation where a structural model is either formulated or updated as a basis for damage detection, a data-based situation where there is no prior information on the structural model and only the sensor data is used, or a mixture of the two situations.

We apply a recently developed framework for Bayesian switching dependence analysis under uncertainty [6] to time-series data obtained from accelerometers located at multiple positions on a building. This model is effectively a computational representation of not only the physical structural system, but also the act of collecting information on that system through the use of sensors. By accounting for interactions between sensor signals collected from the system in different locations, the hope is to infer a representation of the structural connections between locations in the structure or the underlying physics without have any knowledge of the actual structural configuration or dynamics. Assuming that the model learned from a set of data is exclusive to the corresponding physical structural configuration and condition, a change in the model parameters could be indicative of a change in the measured physical structure which might be caused by damage. In order to see if these assumptions might hold true, we test the methodology on data from a model structure in various intact and damaged conditions.

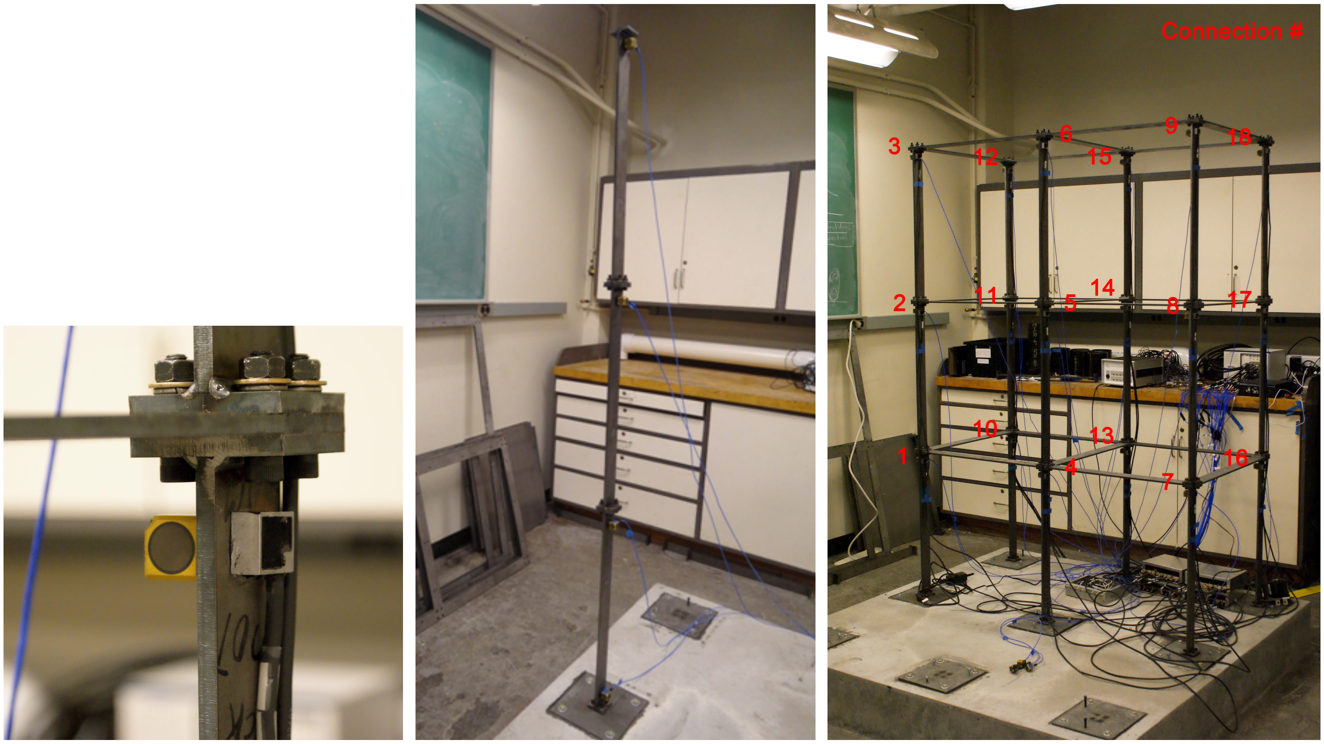
We present experimental setup in Section 2, the Bayesian framework and its modification for a classification problem in Section 3, and experimental results in Section 4. We summarize conclusions in Section 5.

2 EXPERIMENTAL SETUP

Two experimental test structures were used to generate data to test the approach for application on a structure. Both structures are made of modular elements that are based on steel columns that are 60 cm × 5.08 cm × 0.64 cm, and bolted together by 4 bolts at each connection as shown in Fig. 1a as an example of a typical connection. The structures are bolted to a heavy concrete foundation as a reaction mass. They are instrumented with piezoelectric triaxial accelerometers that have a sampling rate of 6000 Hz, and the number used differs for each structure.

The first, simpler structure is a vertical cantilever beam that consists of three steel column elements shown in Fig. 1b. Damage is introduced on one of the two middle bolted connections in either a minor damage case where two of four bolts in the flexible direction are removed, or a major damage case where the four bolts are loosened to only be hand tight. This structure is instrumented with 4 accelerometers, one at each connection, including the connection with the foundation, and at the top of the structure. In order to excite the cantilever beam, it is displaced by approximately 5 cm and then released and allowed to freely vibrate for 10 seconds, during which data was collected. There are 10 test sequences for each damage scenario, and they are summarized in Table 1a.

The second structure is a 3 story 2 bay configuration with a footprint of 120 cm × 60 cm as shown in Fig. 1c. The structure consists of steel columns and beam frames of similar dimensions for each story that are bolted together to form each story. Damage is similarly introduced on the bolted connections with the minor and major damage cases by removing two bolts or loosening all four at connections 1 and 17, which are on opposite corners of the structure, with 1 being on the first story, and 17 being on the second. This structure is instrumented with 18 triaxial accelerometers at each of the connections between elements. For this structure the excitation is a small shaker with a weight of 0.91 kg and a piston weight of 0.17 kg that was attached to the top corner of the structure at connection 18, which provided a random white Gaussian noise excitation in the frequency range of 5 - 350 Hz in the flexible direction. Test measurements lasted for 30 seconds, during which the shaker is always exciting the structure, thus there is no ramp up or unforced section of the data. The damage scenarios are summarized in Table 1b. For each damage scenario, 10 sequences were acquired.



(a) Bolted connection (b) Simple column structure (c) 3 story 2 bay structure

Fig. 1 Details of the experimental setup

Table 1 Test cases and damage scenarios for structural models.

(a) Column structure

Test Case	Damage Scenario
1	Intact column
2	Minor damage, lower joint
3	Major damage, lower joint
4	Minor damage, upper joint
5	Major damage, upper joint

(b) 3 story 2 bay structure

Test Case	Damage Scenario
1	Intact column
2	Minor damage at 17
3	Major damage at 17
4	Minor damage at 1
5	Major damage at 1
6	Major damage at 1 and 17

3 THEORY

We describe the state-space switching interaction model (SSIM) of [6] in Section 3.1 and its modification for the application to classification of time-series in Section 3.2.

The relevant background for this paper are probabilistic graphical models (Bayesian networks and dynamic Bayesian network in particular) and principles of Bayesian inference. Graphical models are a language that uses graphs to compactly represent families of joint probability distributions among multiple variables that respect certain constraints dictated by a graph. In particular, a Bayesian network (BN) consists of a directed acyclic graph $G = (V, E)$, whose nodes $X_1, X_2, \ldots, X_N$ represent random variables, and a set of conditional distributions $p(X_i|\, pa(X_i))$, $i = 1, \ldots, N$, where $pa(X_i)$ is a set of variables that correspond to the parent nodes (parents) of node X_i. Dynamic Bayesian networks (DBNs) are Bayesian networks that model sequential data, such as time-series. Each signal in a model is represented with a sequence of random variables that correspond to its value at different indices, or discrete time points. Edges are allowed only from a variable with a lower index to a variable with a higher index (i.e., they must "point" forward in time). An introduction to the Bayesian approach and Bayesian networks can be found in [7]. An introduction to dynamic Bayesian networks can be found in [8].

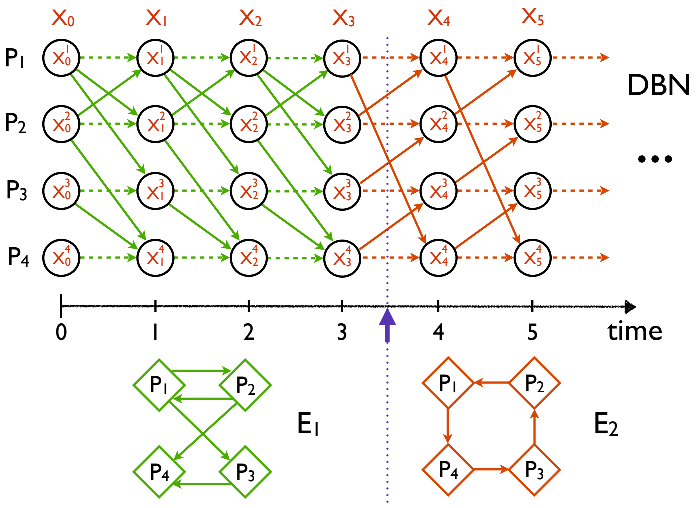


Fig. 2 Dynamic Bayesian Network (DBN) representation of switching interaction among four signals. They initially evolve according to interaction graph E_1. At time point 4, the interaction pattern changes, and they evolve according to interaction graph E_2. Self-edges are assumed.

3.1 State-Space Switching Interaction Model (SSIM)

We assume that N multivariate signals evolve according to a Markov process over discrete time points $t = 0, 1, \ldots, T$. The value of signal i at time point $t > 0$ depends on the value of a subset of signals $pa(i,t)$ at time point $t-1$. We refer to $pa(i,t)$ as a parent set of signal i at time point t. While the preceding implies a first-order Markov process, the approach extends to higher-ordered Markov processes. A collection of directed edges $E_t = \{(v,i);\, i = 1,\ldots,N,\; v \in pa(i,t)\}$ forms a dependence structure (or so-called interaction graph) at time point t, $G_t = (V, E_t)$, where $V = \{1,\ldots,N\}$ is the set of all signals. That is, there is an edge from j to i in G_t if and only if signal i at time point t depends on signal j at time point $t-1$.

Let X_t^i denote a (multivariate) random variable that describes the latent state associated to signal i at time point t. Then, signal i depends on its parents at time t according to a probabilistic model $p(X_t^i|X_{t-1}^{pa(i,t)}, \theta_t^i)$ parametrized by θ_t^i, where $X_{t-1}^{pa(i,t)}$ denotes a collection of variables $\{X_{t-1}^v;\, v \in pa(i,t)\}$. Furthermore, we assume that conditioned on their parents at the previous time point, signals are independent of each other:

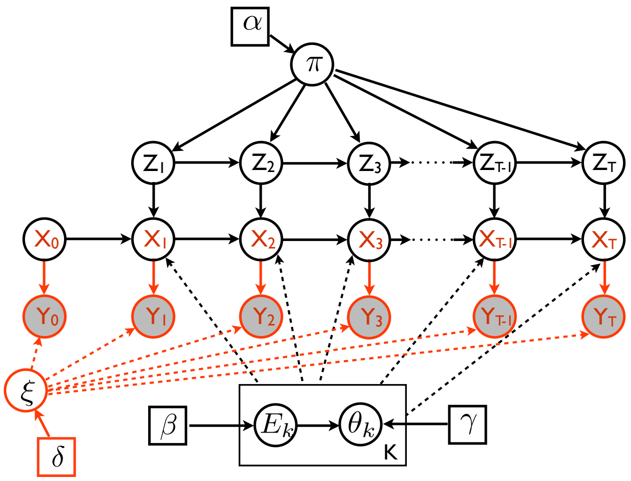
$$p(X_t|X_{t-1}, E_t, \theta_t) = \prod_{i=1}^{N} p(X_t^i|X_{t-1}^{pa(i,t)}, \theta_t^i)\;, \tag{1}$$

where $X_t = \{X_t^i\}_{i=1}^N$ (i.e., X_t is a collection of variables of all signals at time point t) and $\theta_t = \{\theta_t^i\}_{i=1}^N$. Structure E_t and parameters θ_t determine a dependence model at time t, $\mathcal{M}_t = (E_t, \theta_t)$. Finally, we express a joint probability of all variables at all time points, X, as

$$p(X) = p(X_0|\theta_0) \prod_{t=1}^{T} p(X_t|X_{t-1}, E_t, \theta_t) = \prod_{i=1}^{N} p(X_0^i|\theta_0^i) \prod_{t=1}^{T} \prod_{i=1}^{N} p(X_t^i|X_{t-1}^{pa(i,t)}, \theta_t^i)\,. \tag{2}$$

The stochastic process of Eq. 2 can be represented using a dynamic Bayesian network (DBN), such that there is a one-to-one correspondence between the network and the collection of interaction graphs over time, as shown in Figure 2.

In order to learn time-varying interaction from time-series data, we assume that the dependence model switches over time between K distinct models, $\tilde{\mathcal{M}}_k = (\tilde{E}_k, \tilde{\theta}_k), k = 1,\ldots,K$. More formally, for each time point t, $\mathcal{M}_t = \tilde{\mathcal{M}}_k$ for some k, $1 \le k \le K$. One interaction may be active for some period of time, followed by a different interaction over



Multinomials π are sampled from Dirichlet priors parametrized by α as
$(\pi_1, \ldots, \pi_K) \sim \mathcal{D}ir(\alpha_1, \ldots, \alpha_K)$,
$(\pi_{i,1}, \ldots, \pi_{i,K}) \sim \mathcal{D}ir(\alpha_{i,1}, \ldots, \alpha_{i,K})\ \forall i$.

K structures $\tilde{E}_k$ and parameters $\tilde{\theta}_k$ are sampled from the corresponding priors as
$\tilde{E}_k \sim p(E; \beta)$, $\tilde{\theta}_k \sim p(\theta|\tilde{E}_k; \gamma)$, $\forall k$.

Parameters of the observation model are sampled as $\xi^i \sim p(\xi^i; \delta)$, $\forall i$.

Initial values X_0 and Y_0 are generated as $X_0 \sim p(X_0|\theta_0)$ and $Y_0 \sim p(Y_0|X_0, \xi)$.

For each $t = 1, 2, \ldots, T$ (in that order), values of Z_t, X_t and Y_t are sampled as
$Z_t \sim \mathcal{M}ult(\pi_{Z_{t-1},1}, \ldots, \pi_{Z_{t-1},K})$ or
$Z_t \sim \mathcal{M}ult(\pi_1, \ldots, \pi_K)$ if $t = 1$,
$X_t \sim p(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t})$ and $Y_t \sim p(Y_t|X_t, \xi)$.

Fig. 3 State-space switching interaction model (SSIM) and its conditional distributions (generative procedure).

another period of time, and so on, switching between a pool of possible interactions. This is illustrated in Figure 2. Let Z_t, $1 \le t \le T$, be a discrete random variable that represents an index of a dependence model active at time point t; i.e., $\mathcal{M}_t = \tilde{\mathcal{M}}_{Z_t}$, $Z_t \in \{1, \ldots, K\}$. We can now rewrite the transition model (Equation 1) as

$$p(X_t|X_{t-1}, Z_t, \tilde{E}, \tilde{\theta}) = p(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t}) = \prod_{i=1}^{N} p(X_t^i|X_{t-1}^{\tilde{pa}(i,Z_t)}, \tilde{\theta}^i_{Z_t}) \ , \tag{3}$$

where $(\tilde{E}, \tilde{\theta}) = \{(\tilde{E}_k, \tilde{\theta}_k)\}_{k=1}^{K}$ is a collection of all K models and $\tilde{pa}(i, k)$ is a parent set of signal i in $\tilde{E}_k$. We can also rewrite Equation 2 as $p(X|Z, \tilde{E}, \tilde{\theta}) = p(X_0|\theta_0) \prod_{t=1}^{T} p(X_t|X_{t-1}, Z_t, \tilde{E}, \tilde{\theta})$, where $Z = \{Z_t\}_{t=1}^{T}$. To distinguish from signal state, we call Z_t a switching state (at time t) and Z a switching sequence. Furthermore, we assume that Z forms a first order Markov chain:

$$p(Z) = p(Z_1) \prod_{t=2}^{T} p(Z_t|Z_{t-1}) = \pi_{Z_1} \prod_{t=2}^{T} \pi_{Z_{t-1},Z_t} \ , \tag{4}$$

where $\pi_{i,j}$ is a transition probability from state i to state j and π_i is the initial probability of state i.

Finally, we model that the observed value Y_t^i of signal i at time t is generated from its state X_t^i via a probabilistic observation model $p(Y_t^i|X_t^i, \xi_t^i)$ parametrized by ξ_t^i. For simplicity, we assume that the observation model is independent of the state ($\xi_t^i = \xi^i$, $\forall t, i$),

$$p(Y|X, \xi) = \prod_{t=0}^{T} \prod_{i=1}^{N} p(Y_t^i|X_t^i, \xi^i), \tag{5}$$

where $Y = \{Y_t\}_{t=1}^{T}$ is the observation sequence and ξ is the collection of parameters $\{\xi^i\}_{i=1}^{N}$.

The choice of dependence and observations models is application specific and will impact the complexity of some of the inference steps, as discussed in Section 3.1.1.

The full SSIM generative model, shown in Figure 3, incorporates probabilistic models described above along with priors on structures and parameters.

Here, β are the hyperparameters of the prior on dependence structure, $p(E; \beta)$, and γ are the hyperparameters of the prior on dependence model parameters given structure, $p(\theta|E; \gamma)$. We assume that these priors are the same for all

K models. Since the distribution on structure is discrete, in the most general form, β is a collection of parameters $\{\beta_E\}$ (one parameter for each structure), such that β_E is proportional to the prior probability of E:

$$p(E;\beta) = \frac{1}{B}\beta_E \propto \beta_E \, , \tag{6}$$

where $B = \sum_E \beta_E$ is a normalization constant. Note that the prior on parameters, $p(\theta|E;\gamma)$, may depend on the structure and γ is, in general, a collection $\{\gamma_E\}$ of sets of hyperparameters, such that $p(\theta|E;\gamma) = p(\theta;\gamma_E)$.

Learning Bayesian network structures (under reasonable assumptions) is NP hard [9]. The number of possible structures is superexponential in the number of nodes, and, in the worst case, it may be necessary to calculate the posterior of each one separately. The same holds in the case of inference of a dependence structure described above (i.e., a dependence structure of a homogenous DBN). The number of possible such structures is 2^{N^2}.

We employ two fairly general assumptions in order to reduce the complexity of inference over structures. First, we assume a modular prior on structure and parameters [10−13], which decomposes as a product of priors on parent sets of individual signals and associated parameters:

$$p(E,\theta\,|\,\beta,\gamma) = \prod_{i=1}^{N} p(pa(i)\,|\,\beta)\; p(\theta^i\,|\,pa(i);\gamma) \; . \tag{7}$$

As a result, parent sets can be chosen independently for each signal [14], and the total number of parent sets to consider is $N2^N$, which is exponential in the number of signals. Also, β is no longer a collection of parameters per structure, but rather a collection of parameters $\{\beta_{i,pa(i)}\}$ (one parameter for each possible parent set of each signal), such that

$$p(pa(i);\beta) = \frac{1}{B_i}\beta_{i,pa(i)} \propto \beta_{i,pa(i)} \, , \tag{8}$$

where $B_i = \sum_s \beta_{i,s}$ are normalization constants. Modularity is also reflected in the posterior:

$$p(E,\theta\,|X;\beta,\gamma) = \prod_{i=1}^{N} p(pa(i)\,|\,X;\beta)\; p(\theta^i\,|X,pa(i);\gamma) \; . \tag{9}$$

If, in addition, the number of parents of each signal is bounded by some constant M (a structure with bounded in-degree [11−13]), the number of parent sets to evaluate is further reduced to $O(N^{M+1})$, which is polynomial in N.

Linear Gaussian SSIM. Linear Gaussian state-space switching interaction models (LG-SSIM) are an instance of SSIM in which the dependence and observation models of each signal i at each time point t are linear and Gaussian:

$$\begin{aligned} X_t^i &= \tilde{A}^i_{Z_t}\, X_{t-1}^{\tilde{pa}(i,Z_t)} + w_t^i \, , \qquad & w_t^i &\sim \mathcal{N}(0,\tilde{Q}^i_{Z_t}) \\ Y_t^i &= C^i\, X_t^i + v^i \, , & v^i &\sim \mathcal{N}(0,R^i) \, . \end{aligned} \tag{10}$$

$\tilde{A}^i_k$ and $\tilde{Q}^i_k$ are the dependence matrix and the noise covariance matrix of signal i in the k^{th} dependence model (i.e., $\tilde{\theta}^i_k = (\tilde{A}^i_k, \tilde{Q}^i_k)$), while C^i and R^i are the observation matrix and the noise covariance matrix of the observation model of signal i (i.e., $\xi^i = (C^i, R^i)$). We adopt a commonly used matrix normal inverse Wishart distribution as a conjugate prior on the parameters of a linear Gaussian model (more details are given in Appendix A).

Latent autoregressive LG-SSIM. The model above implies a first order Markov process. However, it extends to a higher, r^{th} order process by defining a new state at time t as $X'_t = [X_t\, X_{t-1}\, \ldots\, X_{t-r+1}]$, i.e., by incorporating a history of length r as a basis for predicting a state at time $t+1$. We will refer to this model as a latent autoregressive (AR) LG-SSIM of AR order r, since the autoregressive modeling is done in the latent space.

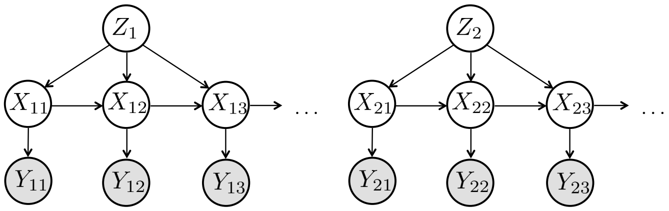


Fig. 4 Part of the SSIM model modified for the application to multiple homogenous sequences.

3.1.1 INFERENCE IN SSIM AND LG-SSIM

Exact inference for the SSIM is generally intractable, and one need to resort to approximate methods. An efficient Gibbs sampling procedure is developed in [6] and shown in Algorithm 1. The procedure alternates between sampling of (1) latent state sequence X, (2) latent switching sequence Z, (3) parameters of switching sequence dependence models π, (4) parameters of K state sequence transition models $(\tilde{E}, \tilde{\theta})$, and (5) parameters of the observation model ξ. In each step, a corresponding variable is sampled from the conditional distribution of that variable given other variables (i.e., the rest of the variables are assumed fixed at that step).

This procedure is particularly efficient when the dependence model and the observation model distributions have conjugate priors, such as in LG-SSIM, as steps 4 and 5 are reduced to performing conjugate updates. In addition, an efficient message-passing algorithm for batch sampling of the state sequence X (step 1) is developed in [6]. On the other hand, steps 2 and 3 are independent of these choice, and thus inherent to SSIM in general. Step 3 is simply a conjugate update of a Dirichlet distribution, while an efficient message passing algorithm for batch sampling of the switching sequence Z is shown in [14].

Algorithm 1 SSIM Gibbs sampler	Algorithm in LG-SSIM
1. $X \sim p(X \mid Z, Y, \tilde{E}, \tilde{\theta}, \xi)$	Gaussian-MP block sampling
2. $Z \sim p(Z \mid X, \tilde{E}, \tilde{\theta}, \pi)$	discrete-MP block sampling
3. $\pi \sim p(\pi \mid Z; \alpha)$	conjugate update
4. $\tilde{E}, \tilde{\theta} \sim p(\tilde{E}, \tilde{\theta} \mid Z, X; \beta, \gamma)$	conjugate update
5. $\xi \sim p(\xi \mid X, Y; \delta)$	conjugate update

3.2 Classification with SSIM

The data collected from the two structures consists of multiple sequences taken under intact and different damage scenarios. Since there is no change in conditions during recording of a single sequence, the SSIM model is modified to produce a single switching label on the entire sequence. The modified part of the model is shown in Fig. 4. Each observation sequence $\mathcal{Y}_i = (Y_{i0}, Y_{i1}, \ldots, Y_{iT_i})$ has an associated state sequence $\mathcal{X}_i = (X_{i0}, X_{i1}, \ldots, X_{iT_i})$ and a switching label Z_i, where i is a sequence index and T_i denotes the length of sequence i. We consider the problem of classification of sequences according to the structure condition. Intact structure and each of the damage scenarios are associated with different class labels. The switching label of sequence i, Z_i, indicates its class membership.

Each classification problem has the following form. There are K classes, and a training sequences $\mathcal{Y}_k^{tr}$ is given for each class $k \in \{1, 2, \ldots, K\}$, thus implicitly assuming $Z_k^{tr} = k$. Note that in case there are multiple training sequences from a single class, one can simply assume that $\mathcal{Y}_k^{tr}$ denotes a collection of such sequences. Given a test sequence $\mathcal{Y}^{test}$ and the training data, the goal is to find the probability distribution of the test sequence label, i.e.,

$P(Z^{test} = k \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Y}_2^{tr}, \ldots, \mathcal{Y}_K^{tr})$, for each k. This probability can be computed in the following way:[*]

$$\begin{aligned}
&P(Z^{test} = k \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Y}_2^{tr}, \ldots, \mathcal{Y}_K^{tr}) \\
&\propto P(Z^{test} = k, \mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr}, \mathcal{Y}_2^{tr}, \ldots, \mathcal{Y}_K^{tr}) \\
&= P(Z^{test} = k \,|\, \mathcal{Y}_1^{tr}, \mathcal{Y}_2^{tr}, \ldots, \mathcal{Y}_K^{tr})\, P(\mathcal{Y}^{test} \,|\, Z^{test} = k, \mathcal{Y}_1^{tr}, \mathcal{Y}_2^{tr}, \ldots, \mathcal{Y}_K^{tr}) \\
&\propto P(\mathcal{Y}^{test} \,|\, Z^{test} = k, \mathcal{Y}_k^{tr})
\end{aligned} \tag{11}$$

assuming that, given the training data, the prior probability of a test sequence belonging to class k (prior to seeing the actual test sequence) is uniform, i.e., $P(Z^{test} = k \,|\, \mathcal{Y}_1^{tr}, \mathcal{Y}_2^{tr}, \ldots, \mathcal{Y}_K^{tr}) \propto const$. Therefore, the probability of the test sequence belonging to class k is proportional to its likelihood under the class k model, given the training sequence $\mathcal{Y}_k^{tr}$ from that class. We will commonly write $P(\mathcal{Y}^{test} \,|\, \mathcal{Y}_k^{tr})$ for this likelihood, thus implicitly assuming conditioning on $Z^{test} = k$. It is computed by marginalizing out model structure and parameters (model averaging):

$$P(\mathcal{Y}^{test} \,|\, \mathcal{Y}_k^{tr}) = \sum_{\tilde{E}_k} \int_{\tilde{\theta}_k} P(\mathcal{Y}^{test} \,|\, \tilde{E}_k, \tilde{\theta}_k)\, P(\tilde{E}_k, \tilde{\theta}_k \,|\, \mathcal{Y}_k^{tr})\, d\tilde{\theta}_k \,. \tag{12}$$

The term $P(\tilde{E}_k, \tilde{\theta}_k \,|\, \mathcal{Y}_k^{tr})$ is the posterior distribution of model structure and parameters given the training sequence $\mathcal{Y}_k^{tr}$), which then serves as a prior for evaluating the test sequence likelihood. The posterior distribution of the test sequence label, Z^{test}, is then obtained by normalizing the likelihoods of the test sequence against training sequences:

$$P(Z^{test} = k \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Y}_2^{tr}, \ldots, \mathcal{Y}_K^{tr}) = \frac{P(\mathcal{Y}^{test} \,|\, \mathcal{Y}_k^{tr})}{\sum_{k'} P(\mathcal{Y}^{test} \,|\, \mathcal{Y}_{k'}^{tr})} \,, \tag{13}$$

and the test sequence is classified according to the label that maximizes this distribution:

$$\hat{Z}^{test} = \arg\max_k P(Z^{test} = k \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Y}_2^{tr}, \ldots, \mathcal{Y}_K^{tr}) = \arg\max_k P(\mathcal{Y}^{test} \,|\, \mathcal{Y}_k^{tr}) \,. \tag{14}$$

Computing the likelihood in Eq. 12 in closed form is intractable in general. The latent training and test state sequences, $\mathcal{X}_k^{tr}$ and $\mathcal{X}^{test}$, need to be marginalized out to compute $P(\tilde{E}_k, \tilde{\theta}_k \,|\, \mathcal{Y}_k^{tr})$ and $P(\mathcal{Y}^{test} \,|\, \tilde{E}_k, \tilde{\theta}_k)$, respectively, and simultaneous marginalization of a state sequence and model structure and parameters is analytically intractable. Instead, this likelihood can be computed via simulation:

$$P(\mathcal{Y}^{test} \,|\, \mathcal{Y}_k^{tr}) \approx \frac{1}{N_s} \sum_{j=1}^{N_s} P(\mathcal{Y}^{test} | \, \hat{E}_j, \hat{\theta}_j) \,, \quad (\hat{E}_j, \hat{\theta}_j) \sim P(\tilde{E}_k, \tilde{\theta}_k \,|\, \mathcal{Y}_k^{tr}) \,. \tag{15}$$

N_s instances of dependence models, $(\hat{E}_j, \hat{\theta}_j)$, are sampled from the posterior distribution of the model given training sequence. The test sequence likelihood is evaluated against each of the sampled models, and then averaged out. On the other hand, in an approximate model which assumes no observation noise (i.e., $\mathcal{X}_i \equiv \mathcal{Y}_i$), the likelihood in Eq. 12 can be computed in closed form by updating the conjugate prior on dependence structure and parameters with the training data and then evaluating the likelihood of the test data against thus obtained posterior.

4 RESULTS

The basic goals here are to detect changes in a structure due to damage, provide some probabilistic description of the occurrence of damage, and try to infer the physical structure without any knowledge of the actual structure.

We consider the problem of classification of sequences according to the structure condition, as described in Section 3.2. In each dataset, there are 10 sequences of each class. We perform 10 rounds of classification. In round j, sequence j from each class is included in the training set, while the other 9 sequences of each class are used for testing. Classification results are then averaged over all 10 rounds.

[*] In this section, hyperparameters are omitted from equations for brevity.

We employ a latent-AR LG-SSIM model for classification. We find that AR order 5 is sufficient to produce good classification result, although there is a slight advantage by further increasing this order. Hyperparameter values are either estimated from data or set in a general fashion (e.g., implying a broad prior distribution). In all experiments, we assume presence of a self edge for each node in the dependence structure. The bound on the number of additional allowed parents is set to 3 (maximum) in the single column case. In the 3 story 2 bay structure data, however, we found that the best classification results are obtained when no additional parents (other than self) are allowed. Explaining this result requires further investigation.

We compared the classification results obtained by the full SSIM model and an approximate model which assumes no observation noise (Section 3.2) and found that on the datasets presented here the full model performs only slightly better, but at the significant additional computational cost (mainly due to step 1 in the inference algorithm). Therefore, we present here detailed results obtained using the approximate model.

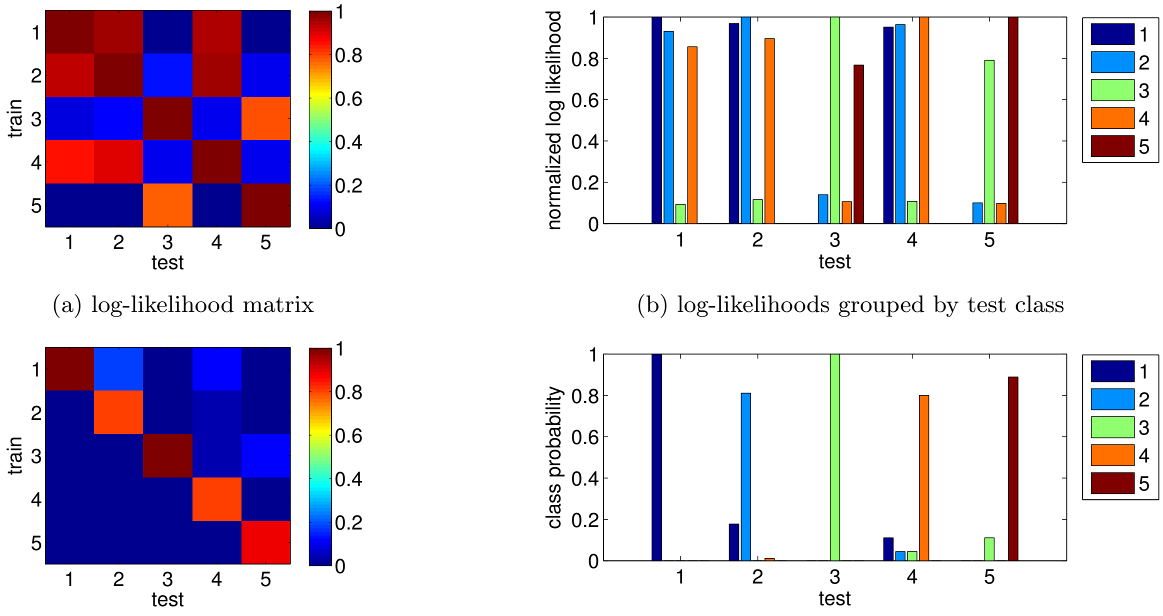
4.1 Single column results

First, for each pair of classes i and j, we compute the average log-likelihood of a test sequence from class i given a training sequence from class j (the average is over all pairs of sequences from classes i and j). Note that the average log-likelihoods do not account for the variability within a class and thus can only partially predict classification results. However, they can be considered as a measure of (asymmetric) similarity between classes. In particular, the comparison of log-likelihoods of a test class given different training classes is useful to indicate its possible “confusion” with other classes. The log domain is chosen to bring likelihoods closer to each other for the purpose of illusrtation, since the differences in likelihoods are huge in their original domain.

The resulting class-class log-likelihood matrix is shown in Fig. 5a. For the purpose of visualization, each column is normalized to contain values between 0 and 1, which does not change the relative comparison of values within a column. A different visualization of the same log-likelihood matrix is shown in Fig. 5b, in which each group of bars corresponds to a single test class, while bars within a group correspond to different training classes. Clearly, the average log-likelihood of each class is the highest when conditioned on sequences from the same class (diagonal entries). This suggests that the model indeed captures important features pertained to each class via posterior distribution of parameters. However, for some classes, the log-likelihood is also relatively high when conditioned on some of the classes other than itself. For example, the intact class (1) and the two minor damage classes (2 and 4) are the closest to each other in that sense. Also, the two major damage classes (3 and 5) are close to each other, although less than the previous three classes. On the other hand, there is a significantly higher separation between the low- and high-damage classes, and, as we will see next, a sequence from one of these groups is rarely misclassified as belonging to a class from the other group.

Classification results are shown in Figs. 5c and 5d. Again, these are two different visualizations of the same results. For each pair of classes, test class i and training class j, the frequency of classifying a test sequence from class i as belonging to class j is shown. Therefore, each column in the matrix in Fig. 5c, as well as each group of bars in Fig. 5d, must sum to one. Overall, sequences are classified correctly most of the times (high diagonal values). Sequences from the two minor damage classes (2 and 4) are occasionally misclassified as belonging to the intact class (1), while sequences from the two major damage classes (3 and 5) are never misclassified as belonging to one of the low-damage classes and occasionally misclassified as belonging to the other major damage class.

Finally, we analyze classification accuracy as a function of training and test sequence lengths. Fig. 6a shows the overall classification accuracy (averaged over all classes) for three different training sequence lengths, $1,000$, $5,000$ and $10,000$, and ten test sequence lengths ranging from $1,000$ to $10,000$. Interestingly, for a fixed training sequence length, classification accuracy increases as the test sequence length increases only until it becomes equal to the training sequence length, after which it start decreasing. This result suggests that the properties of these time-series data change over time. Namely, subsequence for training and testing are always extracted starting at the same time in all sequences. Therefore, when training and test sequences are of the same length, they are aligned with respect to where they are in the measurement process (assuming that different sequences are measured under the same or very similar conditions). However, when the test sequence length increases beyond the training sequence length, test sequences start to increasingly incorporate parts of the process that was not included in training. Similarly, when



(a) log-likelihood matrix

(b) log-likelihoods grouped by test class

(c) classification frequencies matrix

(d) classification frequencies grouped by test class

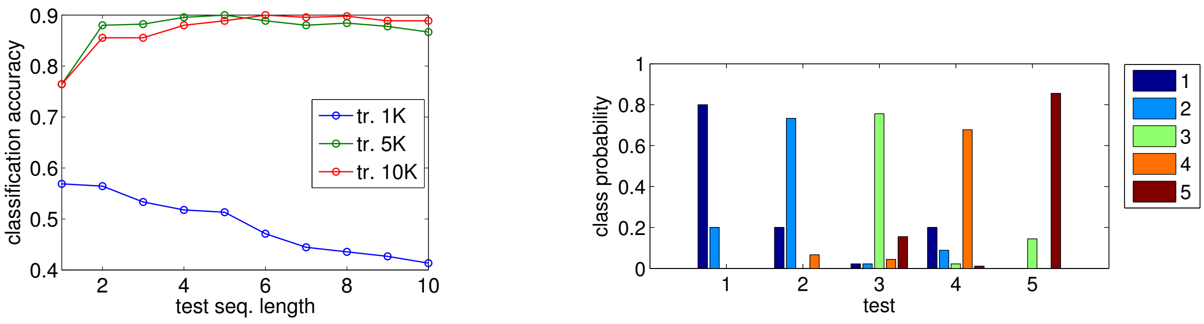
Fig. 5 Column structure data class-class log-likelihoods are shown as (a) matrix and (b) bar groups. Similarly, classification frequencies are shown as (c) matrix and (d) bar groups.

test sequences are shorter than training sequences, training sequences include characteristics of a broader window of the process than is tested. This also can explain why the classification results are overall not better when the training sequence length is $10,000$ than when it is $5,000$. Likely, a window of $10,000$ is too broad and the additional amount of data, the second $5,000$ samples, does not help, since it differs in behavior than the first $5,000$ time samples. Naturally, there is a tradeoff between this behavior and the sequence length. For example, $1,000$ is too short, and the results with that length are clearly much worse. The phenomenon explained here could be attributed to the nature of excitation used in this setup, which is free vibration. The results with the shaker excitation, shown below, do not follow this pattern and behave as with one's expectations – more test or training data consistently yields higher accuracy. Lastly, Fig. 6b shows classification results for training and test sequence lengths equal to $5,000$ and $1,000$, respectively, which could be compared to the results in Fig. 5d, in which both lengths are $5,000$.

4.2 3-story 2-bay structure results

We present the same set of results on the 3-story 2-bay structure data. Average log-likelihoods between all pairs of classes are shown as a matrix in Fig. 7a and as bars grouped by test class in Fig. 7b. Again, these log-likelihoods are normalized such that each column in the matrix are between 0 and 1. As with the single column structure, the average log-likelihood of a sequence of one class is the highest when conditioned on a sequence from that same class (diagonal elements), and the highest confusion is between the low-damage classes, namely, the intact class, 1, and the two minor damage classes, 2 and 4. The lesser major damage classes, 3 and 5, seem to be occasionally confused as classes with either smaller or higher damage relative to them. Finally, the greater major damage class, 6, is most similar to the lesser major damage classes.

Classification results in terms of frequencies (fraction of times a sequence from one class is classified as belonging to another class) are shown as a matrix in Fig. 7c and as bars grouped by test class in Fig. 7d. Sequences from major damage classes (3, 5 and 6) are classified almost perfectly. On the other hand, some confusion between the three low-damage classes (1, 2 and 4) is present. In particular, sequences from the class that corresponds to a minor damage at node 17 are often misclassified as belonging to the intact class. This could possibly be attributed to the closeness of this node to the shaker.



(a)

(b)

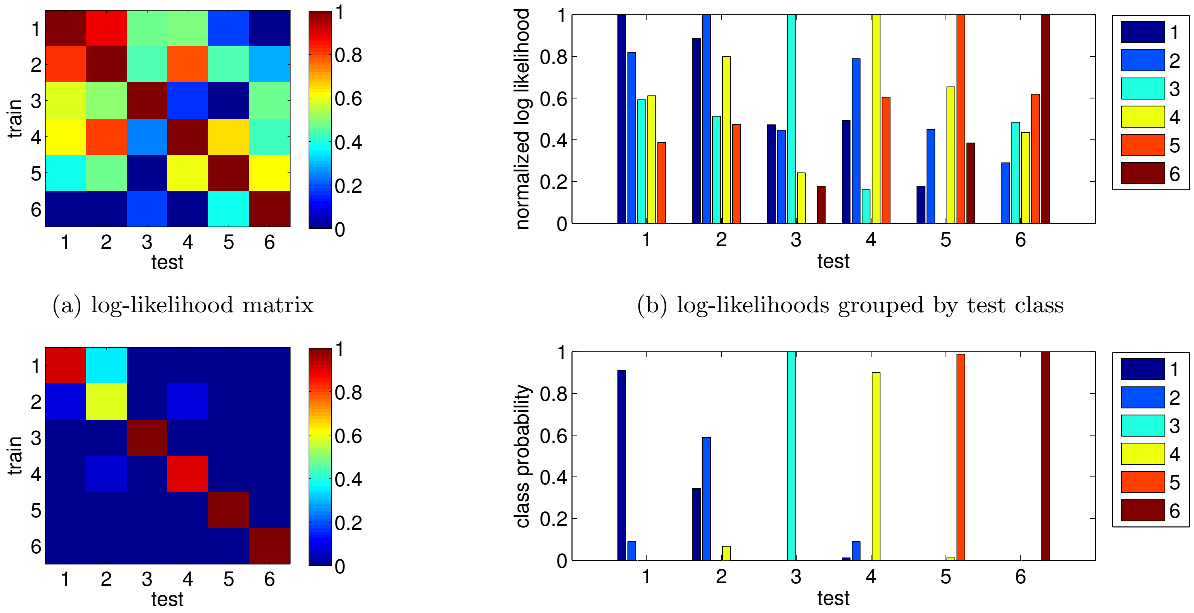
Fig. 6 (a) Overall classification accuracy on column structure data as a function of training and test sequence lengths. (b) Classification frequencies (by test class) when training and test sequence lengths are $5K$ and $1K$, respectively.

Table 2 Most likely parent sets of each node when the number of additional parents is bounded to 1, 2, and 3. Each node is assumed to be its own parent. Only additional parents are shown.

node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1-bound	10	11	6	7	14	3	8	2	3	1	2	15	14	5	12	13	18	12
2-bound	7 10	5 11	6 9	1 7	2 8	3 9	1 8	2 9	3 6	1 2	2 17	15 18	7 14	5 11	12 18	13 17	11 18	12 16
3-bound	4 7 10	1 5 11	2 6 9	1 7 16	2 8 17	3 9 15	1 4 8	2 5 9	3 6 15	1 2 7	2 10 17	9 15 18	14 15 16	5 11 18	3 12 18	10 13 17	11 15 18	12 15 16

The overall classification accuracy as a function of training and test sequence lengths is shown in Fig. 8a. Three different training sequence lengths were used, $1,000$, $5,000$ and $10,000$, while the test sequence length is varied from $1,000$ to $10,000$. Unlike with the single column structure results, classification accuracy on the 3 story 2 bay structure data consistently improves with the increased length of either training or a test sequence. This trend suggests that there is likely no significant variability in the dynamics of a sequence over time, and, consequently, longer sequences represent effectively more data. This is an expected behavior, since excitation provided by the shaker is uniform over time. Finally, for comparison with the results in Fig. 7d, in which both lengths are $5,000$, Fig. 8b shows classification results when training and test sequence lengths are $5,000$ and $1,000$.

We also analyze the the results of inference over dependence structures. The most likely parent set of each node obtained by structure inference on a single sequece (from intact class) is shown in Table 2. Three different results are shown, in which the number of additional parents (other than self, which is assumed) is bounded to 1, 2 and 3. As can be seen, these data favor larger parent sets and the most likely parent set for each node in each scenario exhausts the constraint. Clearly, each node is most often best explained by nearby nodes. The exact meaning of these results and they correspond to the physics of the problem requires further investigation. Besides explaining data and properties of a physical structure, one possible application of dependence structure analysis is in recovering the topology of a physical structure when it is unknown. Another interesting observation is that, while by increasing the number of allowed parents, the new "best" parent set of a node is most commonly a superset of the previous one, this is not always the case. For example, the best parent of size 1 of node 5 is 14. However, node 14 is not included in its best parent sets of sizes 2 and 3. This reiterates the need for dependence inference at the structure level rather than at the pairwise level – simply adding nodes from k most likely pairwise relationships does not result in the mostly likely parent set of size k.



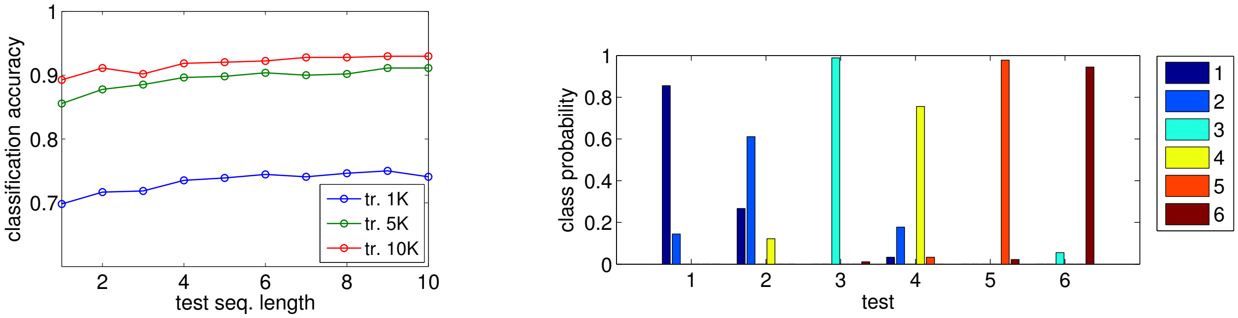
(a) log-likelihood matrix

(b) log-likelihoods grouped by test class

(c) classification frequencies matrix

(d) classification frequencies grouped by test class

Fig. 7 3 story 2 bay structure data class-class log-likelihoods are shown as (a) matrix and (b) bar groups. Similarly, classification frequencies are shown as (c) matrix and (d) bar groups.



(a)

(b)

Fig. 8 (a) Overall classification accuracy on 3 story 2 bay structure data as a function of training and test sequence lengths. (b) Classification frequencies when training and test sequence lengths are $5K$ and $1K$, respectively.

5 CONCLUSION

In this paper we have presented an approach using Bayesian inference on a state-space switching interaction model to detect and classify changes indicative of damage in a model structure. Data was collected with accelerometers from two laboratory models, a steel cantilever column and a larger 3 story 2 bay steel structure, and analyzed using the damage detection approach by obtaining the log-likelihoods of test sequences given a different training sequence. For both structures, test data were classified correctly to their respective damage scenarios or the intact structure case with relatively high accuracy. It was also found that generally longer test and training sequences provide better classification accuracy. Inference was done over the dependence structures in the model, and it was determined that the parents of a node were most likely physically connected to that node, possibly providing information about the physical structure. Future work involves testing the approach on more structural configurations and damage scenarios, classification using the edges in the dependency graph, and use as a single-class classifier, but with these preliminary results this approach joins the growing arsenal of effective damage detection methods for statistical based structural health monitoring.

ACKNOWLEDGEMENTS

The authors acknowledge the support provided by Royal Dutch Shell through the MIT Energy Initiative, and thank chief scientists Dr. Dirk Smit, Dr. Sergio Kapusta, project managers Dr. Keng Yap and Dr. Yile Li, and Shell-MIT liaison Dr. Jonathan Kane for their oversight of this work. We also acknowledge the help of Dr. Michael Feng and Draper Laboratory for providing experimental equipment, and James Long, Reza Mohammadi Ghazi, and Young-Jin Cha for help in collecting the experimental data.

A MATRIX NORMAL INVERSE WISHART PRIOR

Here, we consider a linear Gaussian model of a multivariate signal X_t,

$$X_t = A\,X_{t-1} + w_t\,, \qquad w_t \sim \mathcal{N}(0, Q)\,, \tag{16}$$

with parameters A (transition matrix) and Q (noise covariance matrix).

We assume that $\Theta = (A, Q)$ follows a matrix-normal inverse-Wishart distribution, which is a conjugate prior to the dependence model $\mathcal{N}(X_t; AX_{t-1}, Q)$:

$$p(A, Q; M, \Omega, \Psi, \kappa) = \mathcal{MN}\text{-}\mathcal{IW}(A, Q; M, \Omega, \Psi, \kappa) = \mathcal{MN}(A; M, Q, \Omega)\,\mathcal{IW}(Q; \Psi, \kappa)\,. \tag{17}$$

It is a product of (1) the matrix-normal distribution

$$\mathcal{MN}(A; M, Q, \Omega) = \frac{\exp\left(-\frac{1}{2}\operatorname{tr}\left[\Omega^{-1}(A-M)^T Q^{-1}(A-M)\right]\right)}{(2\pi)^{dl/2}|\Omega|^{d/2}|Q|^{l/2}}\,, \tag{18}$$

where d and l are the dimensions of matrix A ($A_{d\times l}$), while $M_{d\times l}$, $Q_{d\times d}$ and $\Omega_{l\times l}$ are the mean, the column covariance and the row covariance parameters; and (2) the inverse-Wishart distribution

$$\mathcal{IW}(Q; \Psi, \kappa) = \frac{|\Psi|^{\kappa/2}}{2^{\kappa d/2}\Gamma_d(\kappa/2)}\,|Q|^{-(\kappa+d+1)/2}\exp(-\frac{1}{2}\operatorname{tr}(\Psi Q^{-1}))\,, \tag{19}$$

where d is the dimension of matrix Q ($Q_{d\times d}$) and $\Gamma_d()$ is a multivariate gamma function while κ and $\Psi_{d\times d}$ are the degree of freedom and the inverse scale matrix parameters. Note how the two distributions are coupled. The matrix normal distribution of the dependence matrix A depends on the covariance matrix Q, which is sampled from the inverse Wishart distribution.

Due to conjugacy, the posterior distribution of parameters A and Q given data sequence $X_0, X_1, \ldots, X_T$ is also a matrix-normal inverse-Wishart distribution:

$$p(A, Q|X_{0:T}; M, \Omega, \Psi, \kappa) = \mathcal{MN}\text{-}\mathcal{IW}(A, Q; M', \Omega', \Psi', \kappa') = \mathcal{MN}(A; M', Q, \Omega')\,\mathcal{IW}(Q; \Psi', \kappa')\,, \tag{20}$$

where

$$
\begin{aligned}
\Omega' &= \left(\Omega^{-1} + \sum_{t=0}^{T-1} X_t X_t^T\right)^{-1} \\
M' &= \left(M\Omega^{-1} + \sum_{t=1}^{T} X_t X_{t-1}^T\right)\Omega' \\
\kappa' &= \kappa + T \\
\Psi' &= \Psi + \sum_{t=1}^{T} X_t X_t^T + M\Omega^{-1}M^T - M'\Omega'^{-1}M'^T\,.
\end{aligned}
\tag{21}
$$

REFERENCES

[1] **Brownjohn, J. M.**, *Structural health monitoring of civil infrastructure*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, Vol. 365, No. 1851, pp. 589–622, 2007.

[2] **Sohn, H., Farrar, C. R., Hemez, F. M., Shunk, D. D., Stinemates, D. W., Nadler, B. R.** and **Czarnecki, J. J.**, A review of structural health monitoring literature: 1996-2001, Los Alamos National Laboratory Los Alamos, NM, 2004.

[3] **Beck, J. L.** and **Katafygiotis, L. S.**, *Updating models and their uncertainties. I: Bayesian statistical framework*, Journal of Engineering Mechanics, Vol. 124, No. 4, pp. 455–461, 1998.

[4] **Vanik, M. W., Beck, J.** and **Au, S.**, *Bayesian probabilistic approach to structural health monitoring*, Journal of Engineering Mechanics, Vol. 126, No. 7, pp. 738–745, 2000.

[5] **Flynn, E. B.** and **Todd, M. D.**, *A Bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing*, Mechanical Systems and Signal Processing, Vol. 24, No. 4, pp. 891–903, 2010.

[6] **Dzunic, Z.** and **Fisher III, J.**, *Bayesian Switching Interaction Analysis Under Uncertainty*, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pp. 220–228, 2014.

[7] **Heckerman, D.**, *A Tutorial on Learning With Bayesian Networks*, Tech. Rep. MSR-TR-95-06, Microsoft Research, March 1995.

[8] **Ghahramani, Z.**, *Learning dynamic Bayesian networks*, *Adaptive processing of sequences and data structures*, pp. 168–197, Springer, 1998.

[9] **Chickering, D. M.**, *Learning Bayesian Networks is NP-Complete*, *Learning from Data: Artificial Intelligence and Statistics V*, edited by Fisher, D. and Lenz, H., pp. 121–130, Springer-Verlag, 1996.

[10] **Buntine, W.**, *Theory Refinement on Bayesian Networks*, *Proceedings of the Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-91)*, pp. 52–60, Morgan Kaufmann, San Mateo, CA, 1991.

[11] **Cooper, G. F.** and **Dietterich, T.**, *A Bayesian method for the induction of probabilistic networks from data*, *Machine Learning*, pp. 309–347, 1992.

[12] **Heckerman, D., Geiger, D.** and **Chickering, D. M.**, *Learning Bayesian Networks: The Combination of Knowledge and Statistical Data*, *Machine Learning*, pp. 197–243, 1995.

[13] **Friedman, N.** and **Koller, D.**, *Being Bayesian about Bayesian Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks.*, Machine Learning, Vol. 50, No. 1–2, pp. 95–125, 2003, full version of UAI 2000 paper.

[14] **Siracusa, M. R.** and **Fisher III, J. W.**, *Tractable Bayesian Inference of Time-series Dependence Structure*, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2009.