Mechanical Systems and Signal Processing 96 (2017) 239–259

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

# A Bayesian state-space approach for damage detection and classification

Zoran Dzunic, Justin G. Chen, Hossein Mobahi, Oral Büyüköztürk *, John W. Fisher III

*Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA*

ARTICLE INFO

*Article history:*
Received 19 December 2016
Received in revised form 29 March 2017
Accepted 31 March 2017
Available online 22 April 2017

*Keywords:*
Graphical models
Bayesian inference
Structural health monitoring
State-space model
Damage detection

ABSTRACT

The problem of automatic damage detection in civil structures is complex and requires a system that can interpret collected sensor data into meaningful information. We apply our recently developed switching Bayesian model for dependency analysis to the problems of damage detection and classification. The model relies on a state-space approach that accounts for noisy measurement processes and missing data, which also infers the statistical temporal dependency between measurement locations signifying the potential flow of information within the structure. A Gibbs sampling algorithm is used to simultaneously infer the latent states, parameters of the state dynamics, the dependence graph, and any changes in behavior. By employing a fully Bayesian approach, we are able to characterize uncertainty in these variables via their posterior distribution and provide probabilistic estimates of the occurrence of damage or a specific damage scenario. We also implement a single class classification method which is more realistic for most real world situations where training data for a damaged structure is not available. We demonstrate the methodology with experimental test data from a laboratory model structure and accelerometer data from a real world structure during different environmental and excitation conditions.



## 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 were added to the structure they might provide an extra indication of where and when potential damage occurs, 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 discrimination of these features are used to separate data collected from intact and changed structures [2]. Statistical or similar methods are essential for being able to discriminate feature changes as a result of structural changes from measurement or environmental variability.

* Corresponding author.
*E-mail address:* obuyuk@mit.edu (O. Büyüköztürk).

http://dx.doi.org/10.1016/j.ymssp.2017.03.043
0888-3270/

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 for the purposes of structural damage detection and classification. 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 having 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 laboratory model structure in various intact and damaged conditions, as well as on data from a real building under ambient and non-ambient conditions, such as during a fireworks show and a small earthquake. These data consist of short sequences of measurements, such that it is unlikely that changes occur within a single sequence. The problem of damage detection can then be cast as a problem of time-series classification. If prior data from possible damage scenarios is available, then this problem is a standard multi-class classification problem. However, in most real scenarios, only data from an intact structure is available a priori. Then, the problem of damage detection can be seen as a single-class classification problem which we also implement. The primary contribution of this paper in extending the work of Dzunic et al. [6] is the application of the methodology as a structural health monitoring algorithm for damage detection. Building on the work presented in [7] involving multi-class classification, this paper also considers single-class classification as well as the inferred graphical model of a system. Additionally, another goal is to see if the inferred graphical model of the system may represent any physical characteristics of an instrumented structure.

In Theory (Section 2), we first provide background on Bayesian inference and graphical models in Section 2.1, then we describe the switching state-space interaction model of Dzunic et al. [6] in Section 2.2, and finally we develop extensions of this model for time-series classification in Section 2.3 and single-class classification in Section 2.4. We describe the experimental setup of a laboratory model structure and the Massachusetts Institute of Technology's Green building in Section 3. We present three sets of results in Section 4. The results of the interaction analysis, given in Section 4.1, indicate that inferred structures correlate significantly with actual physical structures and prior knowledge. Multi-class classification results, presented in Section 4.2, demonstrate that the classification model can classify time-series obtained under intact and different damage scenarios with high accuracy. Finally, single-class classification results, presented in Section 4.3, demonstrate that the single-class classification model detects with high accuracy time-series obtained under conditions that differ form intact or ambient conditions and that it also predicts the "strength of deviation". We finish with conclusions in Section 5.

## 2. Theory

In this section, we first provide relevant background on graphical models and Bayesian inference in Section 2.1. Then, we describe the state-space switching interaction model (SSIM) of [6] in Section 2.2 and its modifications for the applications to time-series classification in Section 2.3 and single-class classification in Section 2.4.

### 2.1. Background

The relevant background for this paper includes probabilistic graphical models (Bayesian networks and dynamic Bayesian networks in particular) and principles of Bayesian inference. An introduction to the Bayesian approach and Bayesian networks can be found in [8]. An introduction to dynamic Bayesian networks can be found in [9].

#### 2.1.1. Graphical models

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. There are two common types: undirected graphical models (also called Markov random fields) and directed graphical models (Bayesian networks), which use undirected and acyclic directed graphs, respectively, to form such constraints. In both cases, nodes of a graph correspond to the variables which joint distribution is modeled. In an undirected graphical model, a joint probability distribution is assumed to be proportional to a product of nonnegative functions (called potentials) over graph cliques (fully connected subgraphs). In a Bayesian network, a distribution is assumed to be a product of conditional distributions of each variable given its parents in the graph. Examples of both types of graphical models are shown in Fig. 1. In this paper, we use Bayesian networks and their variant, dynamic Bayesian networks. We now describe them in more detail.

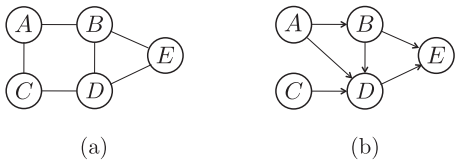


**Fig. 1.** (a) Undirected graphical model example: $P(A,B,C,D,E) \propto f_1(A,B)f_2(A,C)f_3(B,D)f_4(C,D)f_5(B,D,E)$. (b) Directed graphical model example: $P(A,B,C,D,E) = P(A)P(B|A)P(C)P(D|A,B,C)P(E|B,D)$.

*Bayesian networks*. 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 $X_i$. A Bayesian network encodes the following joint probability distribution among variables $X_1, X_2, \ldots, X_N$:

$$p(X_1, X_2, \ldots, X_N) = \prod_{i=1}^{N} p(X_i|pa(X_i)).$$

Conditional distributions $p(X_i|pa(X_i))$ are typically assumed to have some parametric form $p(X_i|pa(X_i),\theta_i)$, in which case learning a Bayesian network means learning parameters $\theta_i$. If, in addition, graph $G$ is unknown, the inference of this graph is commonly referred to as learning the structure of a Bayesian network.

Fig. 2 shows two additional examples of Bayesian networks. In Fig. 2a, $D_1, D_2, \ldots, D_N$ are discrete random variables with values from $\{1,2,\ldots,K\}$ that are drawn independently from a multinomial distribution with parameters $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$, $(\pi_i \geqslant 0, \sum_{i=1}^{K}\pi_i = 1)$, while $\pi$ itself is a random vector drawn from a Dirichlet distribution with parameters $\alpha = (\alpha_1, \alpha_2, \ldots .\alpha_K)$. Then, the overall joint distribution can be written as $p(\pi, D_1, D_2, \ldots, D_N; \alpha) = p(\pi;\alpha)\ \prod_{i=1}^{N} p(D_i|\pi) = Dirichlet(\pi;\alpha)\prod_{i=1}^{N} Mult(D_i;\pi)$. Note that if constant parameters are shown in a graphical model diagram ($\alpha$ in this case), they are written inside a square (as here) or simply without an associated graphical symbol. In Fig. 2b, $X_1, X_2, \ldots, X_N$ are jointly Gaussian univariate random variables with an additional constraint that, for each $i$, $X_i$ is independent of $X_1, \ldots, X_{i-2}$ when conditioned on $X_{i-1}$ (first order Markov assumption): $P(X_1, X_2, \ldots, X_N) = P(X_1)\prod_{i=2}^{N} P(X_i|X_{i-1}) = \mathcal{N}(X_1;\mu_1,\sigma_1^2)\prod_{i=2}^{N}\mathcal{N}(X_i; a_i X_{i-1}, \sigma_i^2)$. Note that this model requires only $2N$ parameters, compared to $N + N^2$ required for a general multivariate Gaussian model.

*Dynamic Bayesian networks*. 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. We will refer to such index as time, although it may not be time-related in general (for example, it can be an index into a genome sequence or a word in a sentence). 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). Let $X_t^i$ denote a random variable that takes the value of signal $i$ at time $t$. Then, if there is an edge from $X_{t_1}^i$ to $X_{t_2}^i, t_2 > t_1$ must hold. Furthermore, edges are often restricted to connect variables at neighboring time points, i.e., they are of the form $X_t^i \rightarrow X_{t+1}^j$. This assumption results in a first-order Markov model over time – signal values at time $t$ are independent of the past given their values at time $t-1$. Let $pa(i,t)$ be the set of parents of signal $i$ at time $t$. Then, the associated conditional probability distributions are of the form $p(X_t^i|X_{t-1}^{pa(i,t)})$, where $X_{t-1}^{pa(i,t)}$ denotes a collection of variables $\{X_{t-1}^v;\ v \in pa(i,t)\}$. In homogenous DBNs, edges (equivalently, parent sets) and conditional distributions are assumed time-invariant. On the other hand, in time-varying DBNs both edges and conditional distributions may vary over time. Fig. 3 shows an example of a time-varying DBN which is piecewise homogenous (switching).

### 2.1.2. Bayesian inference

In contrast to the classical (or frequentist) approach, in which parameters of a statistical model are assumed fixed, but unknown, in the Bayesian approach, parameters are assumed to be drawn from some distribution (called prior distribution

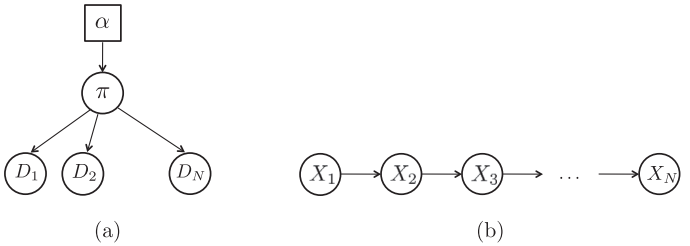


**Fig. 2.** Two examples of Bayesian networks.

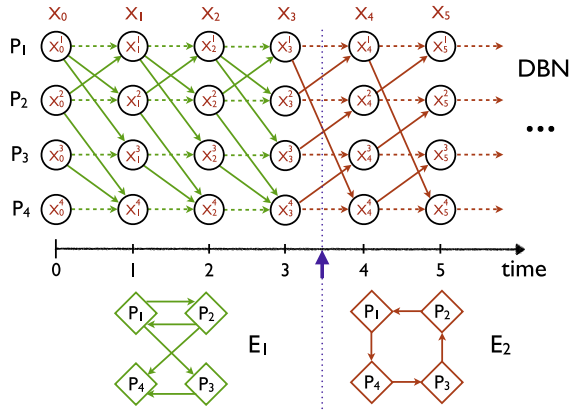


**Fig. 3.** Dynamic Bayesian Network (DBN) representation of switching interaction among four signals. They initially evolve according to interaction graph $E_1$. At time point 4, the interaction pattern changes, and they evolve according to interaction graph $E_2$. Self-edges are assumed.

or simply prior) and therefore treated as random variables. Let $p(X|\theta)$ be a probabilistic model of a phenomenon captured by a collection of variables $X$, with parameters $\theta$, and let $p(\theta;\gamma)$ be the prior distribution of model parameters $\theta$, parametrized by $\gamma$, which are typically called hyperparameters. The prior distribution is often assumed to be known, in which case hyperparameters are treated as constants and are chosen in advance to reflect the prior belief in the parameters $\theta$ (e.g., by a domain expert).

The central computation in Bayesian inference is computing the posterior distribution of parameters $\theta$ given data samples $\mathcal{D} = \{\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_N\}$, namely, $p(\theta|\mathcal{D};\gamma)$. If the samples are independent, the data likelihood is $p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(X = \widetilde{X}_i|\theta)$. The posterior distribution can be computed using the Bayes rule:

$$p(\theta|\mathcal{D};\gamma) = \frac{p(\theta;\gamma)\,p(\mathcal{D}|\theta)}{p(\mathcal{D};\gamma)} = \frac{p(\theta;\gamma)\,p(\mathcal{D}|\theta)}{\int_\theta p(\theta;\gamma)\,p(\mathcal{D}|\theta)\,d\theta}. \tag{1}$$

Note that the denominator $p(\mathcal{D};\gamma)$, the marginal likelihood of data, does not depend on the parameters $\theta$, which are "marginalized out". Therefore, the posterior distribution is proportional to the numerator:

$$p(\theta|\mathcal{D};\gamma) \propto p(\theta;\gamma)\,p(\mathcal{D}|\theta), \tag{2}$$

while the denominator is simply a normalization constant.

Evaluating the numerator above for a specific value of parameters is easy, as it is the product of the prior distribution and the data likelihood terms, which are specified by the model. However, computing the full posterior distribution $p(\theta|\mathcal{D};\gamma)$, or even evaluating it for a specific parameters value (which requires computing the marginal likelihood $p(\mathcal{D};\gamma)$), is in general difficult, as the posterior distribution and the marginal likelihood may not have closed-form analytical expressions. Nonetheless, when the prior distribution, $p(\theta;\gamma)$, is chosen to be a so-called conjugate distribution to the data likelihood distribution, $p(\mathcal{D}|\theta)$, the posterior distribution has the same form as the prior, and, in general, differs from the prior in the value of the hyperparameters, i.e., $p(\theta|\mathcal{D};\gamma) = p(\theta;\gamma')$, where $\gamma'$ is some function of prior hyperparameters, $\gamma$, and the data, $\mathcal{D}$. In this case, computing $\gamma'$ is commonly referred to as "updating" the prior with the data. Not all distributions have a conjugate prior. However, all distributions from the so-called exponential family, which includes a majority of the well-known distributions, have a conjugate prior, and are therefore a convenient choice. Otherwise, simulation (sampling) or some other approximate methods must be employed to represent the posterior.

### 2.2. 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 $\theta_t^i$, 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:

$$p(X_t|X_{t-1},E_t,\theta_t)=\prod_{i=1}^{N}p(X_t^i|X_{t-1}^{pa(i,t)},\theta_t^i), \tag{3}$$

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 $\theta_t$ 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

$$p(X)=p(X_0|\theta_0)\prod_{t=1}^{T}p(X_t|X_{t-1},E_t,\theta_t)=\prod_{i=1}^{N}p(X_0^i|\theta_0^i)\prod_{t=1}^{T}\prod_{i=1}^{N}p(X_t^i|X_{t-1}^{pa(i,t)},\theta_t^i). \tag{4}$$

The stochastic process of Eq. (4) 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 Fig. 3.

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 $k, 1\leqslant k\leqslant K$. One interaction may be active for some period of time, followed by a different interaction over another period of time, and so on, switching between a pool of possible interactions. This is illustrated in Fig. 3. Let $Z_t, 1\leqslant t\leqslant T$, 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 (Eq. (3)) as

$$p(X_t|X_{t-1},Z_t,\tilde{E},\tilde{\theta})=p(X_t|X_{t-1},\tilde{E}_{Z_t},\tilde{\theta}_{Z_t})=\prod_{i=1}^{N}p(X_t^i|X_{t-1}^{\tilde{pa}(i,Z_t)},\tilde{\theta}_{Z_t}^i), \tag{5}$$

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 Eq. (4) 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:

$$p(Z)=p(Z_1)\prod_{t=2}^{T}p(Z_t|Z_{t-1})=\pi_{Z_1}\prod_{t=2}^{T}\pi_{Z_{t-1},Z_t}, \tag{6}$$

where $\pi_{i,j}$ is a transition probability from state $i$ to state $j$ and $\pi_i$ 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 $\xi_t^i$. For simplicity, we assume that the observation model is independent of the state ($\xi_t^i=\xi^i,\ \forall t,i$),

$$p(Y|X,\xi)=\prod_{t=0}^{T}\prod_{i=1}^{N}p(Y_t^i|X_t^i,\xi^i), \tag{7}$$

where $Y=\{Y_t\}_{t=1}^T$ is the observation sequence and $\xi$ 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 2.2.1.

The full SSIM generative model, shown in Fig. 4, incorporates probabilistic models described above along with priors on structures and parameters:

- Multinomials $\pi$ are sampled from Dirichlet priors parametrized by $\alpha$ 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 $Y_t\sim p(Y_t|X_t,\xi)$.

Here, $\beta$ are the hyperparameters of the prior on dependence structure, $p(E;\beta)$, and $\gamma$ 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, $\beta$ is a collection of parameters $\{\beta_E\}$ (one parameter for each structure), such that $\beta_E$ is proportional to the prior probability of $E$:

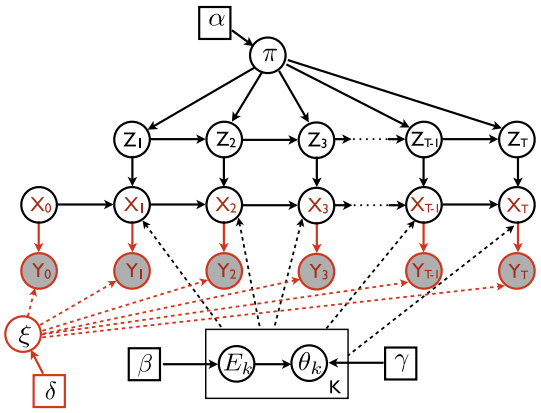


**Fig. 4.** State-space switching interaction model (SSIM).

$$p(E;\beta) = \frac{1}{B}\beta_E \propto \beta_E, \tag{8}$$

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 $\gamma$ 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 [10]. 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 [11–14], which decomposes as a product of priors on parent sets of individual signals and associated parameters:

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

As a result, parent sets can be chosen independently for each signal [15], and the total number of parent sets to consider is $N2^N$, which is exponential in the number of signals. Also, $\beta$ 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

$$p(pa(i);\beta) = \frac{1}{B_i}\beta_{i,pa(i)} \propto \beta_{i,pa(i)}, \tag{10}$$

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

$$p(E,\theta|X;\beta,\gamma) = \prod_{i=1}^{N} p(pa(i)\,|X;\beta)\; p(\theta^i|X,pa(i);\gamma). \tag{11}$$

If, in addition, the number of parents of each signal is bounded by some constant $M$ (a structure with bounded in-degree [12–14]), 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:

$$\begin{aligned} X_t^i &= \widetilde{A}^i_{Z_t} X_{t-1}^{\tilde{pa}(i,Z_t)} + w_t^i, \qquad & w_t^i &\sim \mathcal{N}(0, \widetilde{Q}^i_{Z_t}) \\ Y_t^i &= C^i X_t^i + v^i, & v^i &\sim \mathcal{N}(0, R^i). \end{aligned} \tag{12}$$

$\widetilde{A}^i_k$ and $\widetilde{Q}^i_k$ are the dependence matrix and the noise covariance matrix of signal $i$ in the $k$th dependence model (i.e., $\tilde{\theta}^i_k = (\widetilde{A}^i_k, \widetilde{Q}^i_k)$), 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, $r^{\text{th}}$ 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.

*2.2.1. 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 $\pi$, (4) parameters of $K$ state sequence transition models $(\widetilde{E}, \tilde{\theta})$, and (5) parameters of the observation model $\xi$. 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 [15].

**Algorithm 1.**

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

*2.3. Classification with SSIM*

The SSIM model can simply be extended to multiple sequences, as shown in Fig. 5. Here, $L$ denotes the number of sequences. Each observation sequence $\mathcal{Y}_l = (Y_{l0}, Y_{l1}, \ldots, Y_{lT_l})$ has an associated state sequence $\mathcal{X}_l = (X_{l0}, X_{l1}, \ldots, X_{lT_l})$ and switching sequence $\mathcal{Z}_l = (Z_{l1}, Z_{l2}, \ldots, Z_{lT_l})$, where $l$ is a sequence index and $T_l$ denotes the length of sequence $l$. The inference is still performed as in Algorithm 1, except that steps 1 and 2 are repeated for each sequence separately, while the data needed in steps 3, 4, and 5 (i.e., values of $X, Y$ and $Z$) is pulled from all sequences. We will use $\mathcal{Y} = \{\mathcal{Y}_l\}_{l=1}^{L}, \mathcal{X} = \{\mathcal{X}_l\}_{l=1}^{L}$ and $\mathcal{Z} = \{\mathcal{Z}_l\}_{l=1}^{L}$ to denote collections of observation, latent state and switching sequences, respectively.

In some scenarios, changes in behavior (dependence model) are only expected across different sequences, but not within each sequence. For example, this is the case in a damage detection setup that we exploit, in which short sequences of measurements (e.g., ∼1 min) are recorded far apart from each other (e.g., ∼1 h). Sequences are short enough such that changes within them are unlikely. If switching does not occur within sequences, then each sequence can be assigned a single switching state variable, $Z_l$. We refer to this model as SSIM with multiple homogenous sequences, which is shown in Fig. 6. Since there are no transitions between switching states, this model does not require transition probabilities and parameters of their corresponding Dirichlet priors. Only initial probabilities are needed, and thus $\pi = (\pi_1, \ldots, \pi_K)$ and $\alpha = (\alpha_1, \ldots, \alpha_K)$. In this context, we will refer to sequence switching states as sequence labels. Inference over switching states (labels) in this model is essentially inference over clusters of sequences according to their dependence model (i.e., dynamics).

Classification of sequences can be reduced to the inference in SSIM with multiple homogenous sequences by performing joint inference over training sequences and a test sequence while fixing the labels of training sequences. The probability of any value of the test sequence label is then the frequency of that value in the posterior samples. However, these probabilities can be computed more directly in the following way.

We assume that in a classification problem there are $K$ classes, and, for each class $k \in \{1, 2, \ldots, K\}$, a collection of $N_k^{tr}$ training sequences $\mathcal{Y}_k^{tr} = \{\mathcal{Y}_{kj}^{tr}\}_{j=1}^{N_k^{tr}}$ is given, thus implicitly assuming $Z_{kj}^{tr} = k$ for each $j$. In addition, we will use $\mathcal{Z}_k^{tr} = \{Z_{kj}^{tr}\}_{j=1}^{N_k^{tr}} = \{k\}^{N_k^{tr}}$ to denote a collection of labels associated with training sequences from class $k$, where $\{k\}^{N_k^{tr}}$ denotes a collection of $N_k^{tr}$ values equal to $k$. 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., $P(Z^{test} = k \,|\, \mathcal{Y}^{test}, \{\mathcal{Y}_{k'}^{tr}, \mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K})$, for each $k$. This probability can be computed in the following way[1]:

[1] In this section, hyperparameters are omitted for brevity, but will be reinserted as needed.

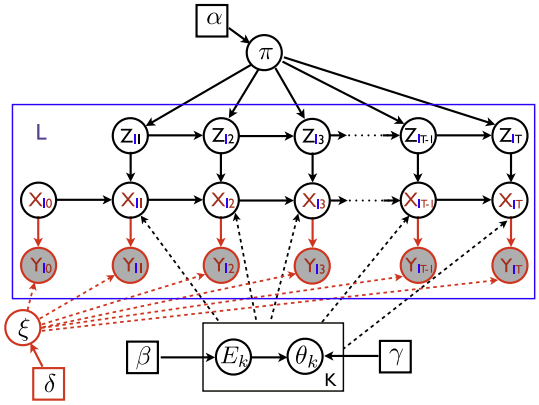


**Fig. 5.** SSIM model with multiple sequences.

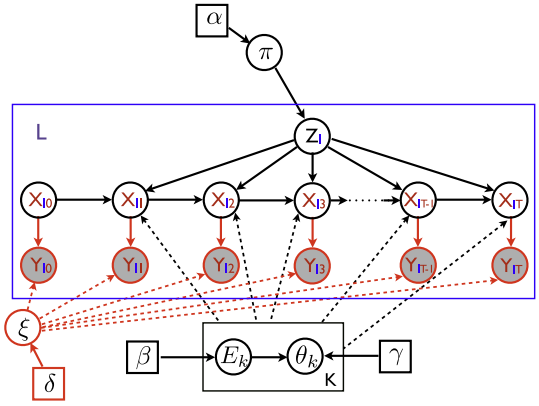


**Fig. 6.** SSIM model with multiple homogenous sequences.

$$
\begin{aligned}
&P(Z^{test}=k\,|\,\mathcal{Y}^{test},\{\mathcal{Y}_{k'}^{tr},\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) \\
&\quad\propto P(Z^{test}=k,\mathcal{Y}^{test}\,|\,\{\mathcal{Y}_{k'}^{tr},\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) \\
&\quad= P(Z^{test}=k\,|\,\{\mathcal{Y}_{k'}^{tr},\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K})\cdot P(\mathcal{Y}^{test}\,|\,Z^{test}=k,\{\mathcal{Y}_{k'}^{tr},\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) \\
&\quad= P(Z^{test}=k\,|\,\{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K})\cdot P(\mathcal{Y}^{test}\,|\,Z^{test}=k,\mathcal{Y}_{k}^{tr},\mathcal{Z}_{k}^{tr}).
\end{aligned}
\tag{13}
$$

The last equality follows from the fact that the test label is independent of the training sequences given training labels, and that the test sequence, assuming it belongs to class $k$, only depends on the training data for that class.

The first term in Eq. (13), $P(Z^{test}=k\,|\,\{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K})$, is the probability of a test sequence belonging to class $k$ before seeing the sequence, given training labels. It can be computed by marginalizing out multinomial parameters $\pi$:

$$
\begin{aligned}
P(Z^{test}=k\,|\,\{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K}) &\equiv P(Z^{test}=k\,|\,\{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K};\alpha) \\
&= \int_{\pi} P(Z^{test}=k\,|\,\pi)\,P(\pi\,|\,\{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K};\alpha)\,d\pi \\
&= \int_{\pi} \pi_k\cdot \mathcal{D}ir(\pi;\alpha_1+N_1^{tr},\alpha_2+N_2^{tr},\ldots,\alpha_K+N_K^{tr})\,d\pi \\
&= \frac{\alpha_k+N_k^{tr}}{\sum_{k'=1}^{K}\alpha_{k'}+N_{k'}^{tr}}.
\end{aligned}
\tag{14}
$$

Note that $P(\pi\,|\,\{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K};\alpha)$ is the posterior distribution of $\pi$ given training labels, which is again a Dirichlet distribution (with updated parameters) due to conjugacy. The final expression is obtained as the expectation of parameter $\pi_k$ with respect to that distribution. For convenience, we will write $P(Z^{test}=k\,|\,\{\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K})\equiv P_{tr}(Z^{test}=k)$.

The second term in Eq. (13), $P(\mathcal{Y}^{test}|Z^{test}=k,\mathcal{Y}_k^{tr},\mathcal{Z}_k^{tr})$, is the marginal likelihood of a test sequence under the class $k$ model, given the training sequences $\mathcal{Y}_k^{tr}$ from that class. It is computed by marginalizing out $k$th model structure and parameters (model averaging):

$$P(\mathcal{Y}^{test}|Z^{test}=k,\mathcal{Y}_k^{tr},\mathcal{Z}_k^{tr})=\sum_{\widetilde{E}_k}\int_{\tilde{\theta}_k}P(\mathcal{Y}^{test}|\widetilde{E}_k,\tilde{\theta}_k)P(\widetilde{E}_k,\tilde{\theta}_k|\mathcal{Y}_k^{tr})\,d\tilde{\theta}_k. \tag{15}$$

The term $P(\widetilde{E}_k,\tilde{\theta}_k|\mathcal{Y}_k^{tr})$ is the posterior distribution of $k$th model structure and parameters given the training sequences $\mathcal{Y}_k^{tr}$, which then serves as a prior for evaluating the test sequence likelihood. For convenience, we will write $P(\mathcal{Y}^{test}|Z^{test}=k,\mathcal{Y}_k^{tr},\mathcal{Z}_k^{tr})\equiv\mathcal{L}_k(\mathcal{Y}^{test}|\mathcal{Y}_k^{tr})$.

Finally, the posterior distribution of the test sequence label, $Z^{test}$, is obtained by normalizing Eq. (13):

$$P(Z^{test}=k|\mathcal{Y}^{test},\{\mathcal{Y}_{k'}^{tr},\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K})=\frac{P_{tr}(Z^{test}=k)\,\mathcal{L}_k(\mathcal{Y}^{test}|\mathcal{Y}_k^{tr})}{\sum_{k'=1}^{K}P_{tr}(Z^{test}=k')\,\mathcal{L}_{k'}(\mathcal{Y}^{test}|\mathcal{Y}_{k'}^{tr})}, \tag{16}$$

and its maximum a posteriori (MAP) estimate is obtained as

$$\widehat{Z}^{test}=\underset{k}{\operatorname{argmax}}\,P(Z^{test}=k|\mathcal{Y}^{test},\{\mathcal{Y}_{k'}^{tr},\mathcal{Z}_{k'}^{tr}\}_{k'=1}^{K})=\underset{k}{\operatorname{argmax}}\,P_{tr}(Z^{test}=k)\,\mathcal{L}_k(\mathcal{Y}^{test}|\mathcal{Y}_k^{tr}). \tag{17}$$

Computing the likelihood in Eq. (15) 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 $P(\widetilde{E}_k,\tilde{\theta}_k|\mathcal{Y}_k^{tr})$ and $P(\mathcal{Y}^{test}|\widetilde{E}_k,\tilde{\theta}_k)$, respectively, and simultaneous marginalization of a state sequence and model structure and parameters is analytically intractable. Instead, this likelihood can be computed via simulation:

$$\mathcal{L}_k(\mathcal{Y}^{test}|\mathcal{Y}_k^{tr})\approx\frac{1}{N_s}\sum_{j=1}^{N_s}P(\mathcal{Y}^{test}|\widehat{E}_j,\hat{\theta}_j),\quad(\widehat{E}_j,\hat{\theta}_j)\sim P(\widetilde{E}_k,\tilde{\theta}_k|\mathcal{Y}_k^{tr}). \tag{18}$$

$N_s$ instances of dependence models, $(\widehat{E}_j,\hat{\theta}_j)$, are sampled from the posterior distribution of the model given training sequences. 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. (15) 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.

### 2.4. Single-class classification with SSIM

In a typical real world structural health monitoring scenario, there is no prior data for a particular type of damage. Even if there has been damage to a structure in the past, it is not likely that exactly the same type of damage will occur in the future, and thus the multi-class classification procedure described in Section 2.3 cannot be applied. On the other hand, data from an intact structure can be recorded easily. Damage detection then becomes a single-class classification problem, in which the goal is to detect whether new data sequences belong to the existing, intact case, or deviate from it and potentially indicate damage.

In the SSIM framework, as a benefit of the Bayesian approach, single-class classification can be simply reduced to multi-class classification by assuming that there are two classes ($K=2$), that the first class indicates the intact scenario, and that there is no data for the second (damage) class ($\mathcal{Y}_2^{tr}=\varnothing,\mathcal{Z}_2^{tr}=\varnothing,N_2^{tr}=0$). Eq. (13) can now be written as:

$$P(Z^{test}=k|\mathcal{Y}^{test},\mathcal{Y}_1^{tr},\mathcal{Z}_1^{tr})\propto P(Z^{test}=k|\mathcal{Z}_1^{tr})\cdot P(\mathcal{Y}^{test}|Z^{test}=k,\mathcal{Y}_k^{tr},\mathcal{Z}_k^{tr}), \tag{19}$$

where, from Eq. (14),

$$P(Z^{test}=k|\mathcal{Z}_1^{tr})\equiv P_{tr}(Z^{test}=k)=\begin{cases}\frac{\alpha_1+N_1^{tr}}{\alpha_1+N_1^{tr}+\alpha_2}, & k=1\\ \frac{\alpha_2}{\alpha_1+N_1^{tr}+\alpha_2}, & k=2\end{cases}, \tag{20}$$

and, from Eq. (15),

$$\begin{aligned}P(\mathcal{Y}^{test}|Z^{test}=k,\mathcal{Y}_k^{tr},\mathcal{Z}_k^{tr})&=\begin{cases}\mathcal{L}_1(\mathcal{Y}^{test}|\mathcal{Y}_1^{tr}), & k=1\\ \mathcal{L}_2(\mathcal{Y}^{test}), & k=2\end{cases}\\ &=\begin{cases}\sum_{\widetilde{E}_1}\int_{\tilde{\theta}_1}P(\mathcal{Y}^{test}|\widetilde{E}_1,\tilde{\theta}_1)P(\widetilde{E}_1,\tilde{\theta}_1|\mathcal{Y}_1^{tr})\,d\tilde{\theta}_1, & k=1\\ \sum_{\widetilde{E}_2}\int_{\tilde{\theta}_2}P(\mathcal{Y}^{test}|\widetilde{E}_2,\tilde{\theta}_2)P(\widetilde{E}_2,\tilde{\theta}_2)\,d\tilde{\theta}_2, & k=2\end{cases}.\end{aligned} \tag{21}$$

Here, $P(\widetilde{E}_2, \tilde{\theta}_2 \,|\, \mathcal{Y}_2^{tr} = \varnothing) = P(\widetilde{E}_2, \tilde{\theta}_2)$ is simply the prior probability of structure and parameters for class 2 (damage scenario), while $\mathcal{L}_2(\mathcal{Y}^{test})$ is the marginal likelihood of the test sequence under that prior.

Finally, Eq. (16) can now be specialized to:

$$P(Z^{test} = 1 \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr}) = \frac{P_{tr}(Z^{test} = 1)\, \mathcal{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr})}{P_{tr}(Z^{test} = 1)\, \mathcal{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr}) + P_{tr}(Z^{test} = 2)\, \mathcal{L}_2(\mathcal{Y}^{test})},$$
$$P(Z^{test} = 2 \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr}) = \frac{P_{tr}(Z^{test} = 2)\, \mathcal{L}_2(\mathcal{Y}^{test})}{P_{tr}(Z^{test} = 1)\, \mathcal{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr}) + P_{tr}(Z^{test} = 2)\, \mathcal{L}_2(\mathcal{Y}^{test})}, \tag{22}$$

which are the probabilities of a given test sequence being "intact" or "damaged", respectively. The higher values of $P(Z^{test} = 2 \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr})$ mean that the dynamics of the test sequence deviates more from the dynamics of intact (training) sequences, which we relate to a higher probability of damage.

In practice, one may want to act upon the knowledge of damage probability. The simplest rule would be to use a threshold, $\epsilon_{dam}$, such that a further investigation of damage is required if this probability exceeds the threshold, i.e., if $P(Z^{test} = 2 \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr}) \geqslant \epsilon_{dam}$. A more sophisticated rule could be that different actions are taken for different levels of damage probability (i.e., when exceeding different thresholds). By rewriting the formula for damage probability as

$$P(Z^{test} = 2 \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr}) = \frac{\frac{P_{tr}(Z^{test}=2)}{P_{tr}(Z^{test}=1)} \cdot \frac{\mathcal{L}_2(\mathcal{Y}^{test})}{\mathcal{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr})}}{1 + \frac{P_{tr}(Z^{test}=2)}{P_{tr}(Z^{test}=1)} \cdot \frac{\mathcal{L}_2(\mathcal{Y}^{test})}{\mathcal{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr})}}, \tag{23}$$

we can see that it depends on the ratio of likelihoods of the test sequence under the intact and prior models, $\frac{\mathcal{L}_2(\mathcal{Y}^{test})}{\mathcal{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr})}$, and on the ratio of damage and intact probabilities prior to seeing a test sequence, $\frac{P_{tr}(Z^{test}=2)}{P_{tr}(Z^{test}=1)}$. The first ratio may depend on the choice of the dependence model (linear Gaussian in this paper) and its hyperparameters (prior on structure and paramaters), but, assuming that these are appropriate/reasonable choices, it most importantly depends on the test sequence itself and how it differs from training sequences. On the other hand, the second ratio, $\frac{P_{tr}(Z^{test}=2)}{P_{tr}(Z^{test}=1)} = \frac{\alpha_2}{\alpha_1 + N_1^{tr}}$, depends only on the prior parameters $\alpha_1$ and $\alpha_2$ and on the number of training sequences. By controlling parameters $\alpha_1$ and $\alpha_2$, this ratio can be set to an arbitrary value (assuming fixed training data). Note that $\alpha_1$ and $\alpha_2$ are pseudo-counts of intact and damaged sequences that reflect our prior belief in the probability of intact versus damage scenario. Intuitively, one should expect a low probability of damage, and thus $\alpha_2 \ll \alpha_1$. On the other hand, the prior probability of damage can be set higher than expected (e.g., $\alpha_2 \approx \alpha_1$), which would reflect the "fear" of damage and increase the posterior probability that a test sequence belongs to a damage scenario. That would simply mean that a larger number of test sequences would "alarm" for damage. Note however the same effect could be achieved by decreasing the "alarm" threshold, $\epsilon_{dam}$. Note also that, instead of using the posterior probability of damage, $P(Z^{test} = 2 \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr})$, to indicate the possibility of damage, one can equivalently use the ratio of posterior probabilities of damage and intact scenarios:

$$\frac{P(Z^{test} = 2 \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr})}{P(Z^{test} = 1 \,|\, \mathcal{Y}^{test}, \mathcal{Y}_1^{tr}, \mathcal{Z}_1^{tr})} = \frac{P_{tr}(Z^{test} = 2)}{P_{tr}(Z^{test} = 1)} \cdot \frac{\mathcal{L}_2(\mathcal{Y}^{test})}{\mathcal{L}_1(\mathcal{Y}^{test} \,|\, \mathcal{Y}_1^{tr})}, \tag{24}$$

and devise rules based on the value of this ratio (e.g., ratio of 1 is equivalent to the damage probability of 0.5).

## 3. Experimental setup

Two different experimental setups were used to test the approach. An experimental laboratory model structure was used to generate data with an intact or damaged structure, as a basis application on a real building. Data from an instrumented building at the Massachusetts Institute of Technology (MIT) was used for seeing if the approach could distinguish between different excitation and environmental conditions in a real structure.

### 3.1. Laboratory model structure

The laboratory model structure is a 3 story 2 bay configuration with a footprint of 120 cm × 60 cm and 240 cm tall, as shown in Fig. 7a. It consists of steel columns and beam frames that have elements with dimensions of 60 cm × 5.08 cm × 0.64 cm, and bolted together by 4 bolts at each connection, an example of which is shown in Fig. 7b. Damage is similarly introduced on the bolted connections with the minor and major damage cases by removing two bolts or loosening all four bolts 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 piezoelectric triaxial accelerometers sampling at 6000 Hz at each of the connections between elements. To excite the structure, a small shaker with a weight of 0.91 kg and a piston weight of 0.17 kg 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 s,

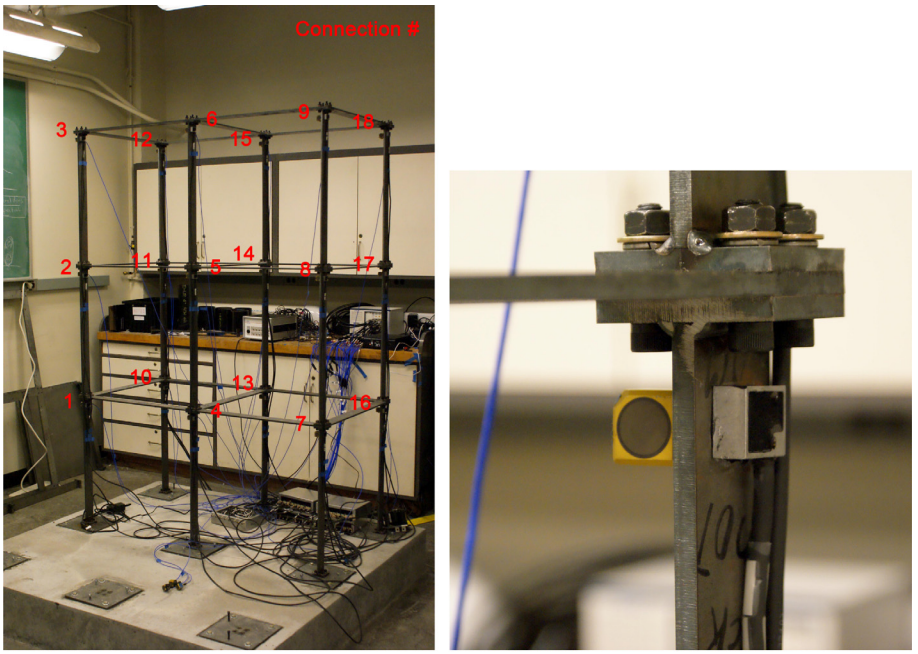


**Fig. 7.** Details of the experimental setup showing (a) the 3 story 2 bay structure and (b) a bolted connection.

**Table 1**
Test cases and damage scenarios for structural models.

*(a) Laboratory model structure*

| Test case | Damage scenario |
|---|---|
| 1 | Intact model structure |
| 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 |

*(b) MIT Green Building excitation or environmental conditions*

| Test case | Date | excitation/env. condition |
|---|---|---|
| 1 | 5/14/2012 | Unknown event |
| 2–3 | 6/22/2012 | Ambient |
| 4–6 | 7/4/2012 | Fireworks |
| 7 | 10/16/2012 | Earthquake |
| 8–10 | 4/15/2013 | Ambient |
| 11–16 | 10/7/2013 | Windy day |

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 1a. For each damage scenario, 10 sequences were acquired.

### 3.2. MIT Green Building

The Green Building is a 21 story building on the campus of MIT that has been instrumented by an accelerometer system, used as a testbed for system identification and structural health monitoring studies [16]. The building itself is shown in Fig. 8a and the locations and directions of the 36 uniaxial accelerometers are shown in Fig. 8b. Data from these accelerometers was used to test the methodology in a different situation from the experimental structure, where there is no known damage or change in the structure between the different data collections. Instead, the excitation and environmental conditions for the structure vary greatly. They are summarized in Table 1b. The excitation conditions vary from typical ambient vibrations, to a day with 20 mph sustained winds, to a 4.0 magnitude earthquake located approximately 100 miles away. The

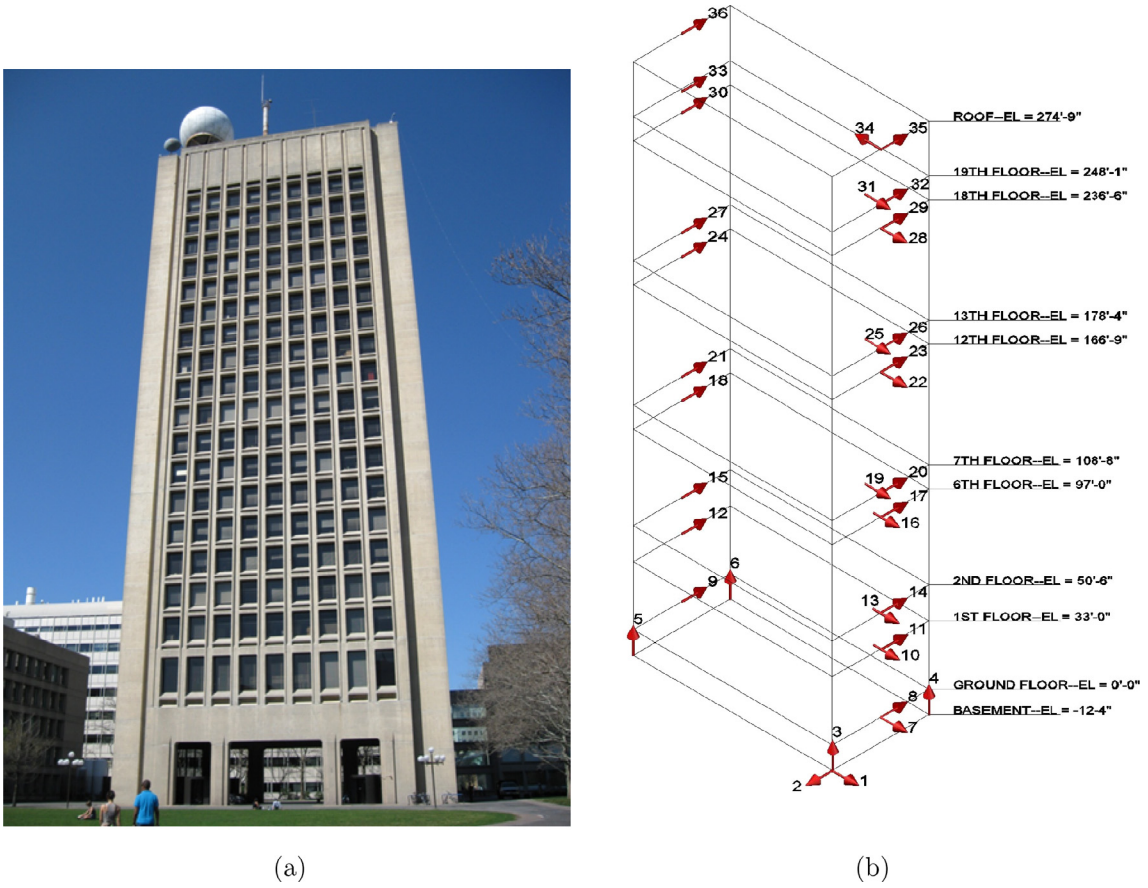


**Fig. 8.** (a) MIT Green Building with (b) sensor locations.

measurements were made in the months of April to October, and with air temperatures typical of Spring, Summer and Fall, with temperature effects potentially inducing small changes in the structure due to internal stresses from differential thermal expansion of materials. The goal with processing this data is to use the ambient excitation data as a baseline for the structure and detect when an anomalous event or excitation occurs, while not triggering false positives during similar ambient excitations, while under different environmental conditions. We subdivided the test cases into several sequences of 30,000 sample length. Some of the sequences are longer than the others, so there are multiple sequences for some of the test cases. The test cases belonging to each excitation and/or environmental condition are given.

## 4. Results

We present three types of results: on interaction analysis, multi-class classification, and single-class classification. The goal of interaction analysis is to understand how an inferred structure among signals correlates with physical properties of a building and how it may differ under different damage or environmental conditions. The goal of multi-class classification experiments is to understand how well the model can differentiate between an intact structure and different types of damage scenarios in an ideal case when damage scenarios are known in advance and training data from these scenarios is available. Finally, the goal of single-class classification experiments is to understand the ability of the model to detect anomalous behavior when only data from an intact (normal) scenario is available a priori, which most closely resembles typical structural health monitoring applications in the real world.

We employ the latent-AR LG-SSIM model in all experiments. We find that an AR order 5 is sufficient to produce good results, 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.

We compared the classification results obtained by the full SSIM model and an approximate model which assumes no observation noise (Section 2.3) 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. Interaction analysis

#### 4.1.1. 3-story 2-bay structure

We analyze the results of inference over dependence structure among signals from different sensors on the **3-story 2-bay structure**. The number of parents of each node is bounded to 4, including the assumed self-dependency (therefore, 3 additional parents are allowed). Each data sequence is split into 18 subsequences that are 10,000 samples long. For each class, the posterior distribution over edges is computed on 180 subsequences that belong to that class (10 original sequences, 18 subsequences each) and then averaged out. The averaging is performed to get a stable result, since the posterior distribution fluctuates across subsequences. A visualization of the parent and child relationships for the intact structure is shown in Fig. 9. Colors represent the node the relationship originates from, and the width of the line represents the edge probability (wider is more likely). Specifically, relationships are plotted if their probability is higher than 0.3, and in Fig. 9a, the parents of the nodes are plotted, while in Fig. 9b the children of nodes are plotted. The nodes are vaguely arranged in the physical shape of the structure, and we can see that a lot of the same possible relationships in the physical structure, such as the columns, the beams, and the cross beams between the two sides of the structure, also show up in the inferred dependence structure.

Edge posteriors are also visualized as a matrix where the rows are the parents, and the columns are the child nodes, shown in Fig. 10a. We see that there are two quadrants where the relationships are strong, the 1–9 parent child relationships, and the 10–18 parent child relationships, which correspond to the two sides of the structure. Within these quadrants, we see that there are strong relationships in groups of three, 1–3 for example, suggestive of the columns in the structure. We also see that there are relationships between nodes separated by three, such as 1:4:7, and similar for all the other nodes, which are suggestive of the beams that connect the nodes in the same story of the structure. Then, the other strong relationship is between the two sides of the structure, 1:10, 2:11, etc. which is seen as an off diagonal.

The results of inference for the other damage scenarios are also shown, and they mostly resemble the structure for the intact scenario. Looking at Fig. 10b instead, where for the damage cases, the difference from the intact scenario is shown, a couple of differences become more obvious. For both of the minor damage scenarios, the differences are minimal. However for major damage at node 1, we see that node 1 is now less likely a parent of nodes 2, 3, and 10. For example, the most likely parents of node 2 in the intact structure are nodes 1, 5 and 11, but for major damage at node 1, node 1 is replaced by node 3 on this list. Note that sensor 1 is actually slightly below the joint, so the damaged joint stands between nodes 1 and 2. For major damage at node 17, node 13 is much less often a parent of node 15, and the same for node 14, being a parent of node 13, all nodes that are physically close to node 17. Also, the dependence of node 18 on nodes 16 and 17 is reduced, as well as the dependence of node 17 on node 18. Note that the damaged joint stands between node 18 on one side and nodes 16 and 17 on the other side. Similarly, the dependence of node 11 on node 17, between which the shortest path goes through the damaged joint, becomes less likely. Finally, in the dual major damage scenario at both 1 and 17, both these effects are seen in the inferred structure.

#### 4.1.2. Green Building

We use a subsequence of length 10,000 from the 6/22/2012 ambient recording to infer the dependence structure among sensor signals from the Green Building. An AR order of 5 was used, with a maximum of 3 additional parents allowed. We plot a visualization of the parent and child relationships in Fig. 11. The color in these plots shows the direction of the sensor in the building, with red for E-W, blue for N-S, and green for vertical sensors. We see that there are many relationships between the

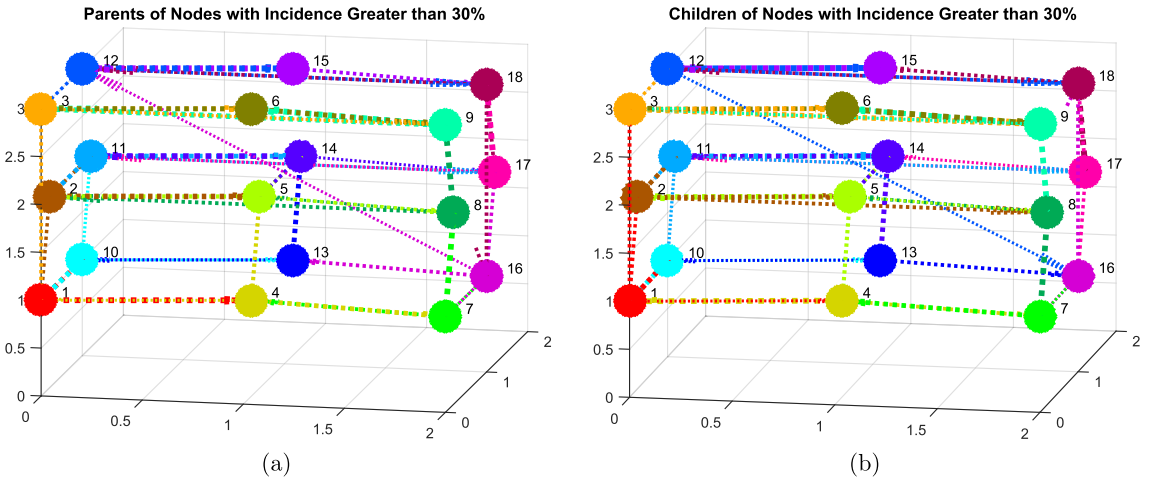


**Fig. 9.** 3D visualization of (a) node parent and (b) node children relationships with probability above 0.3.

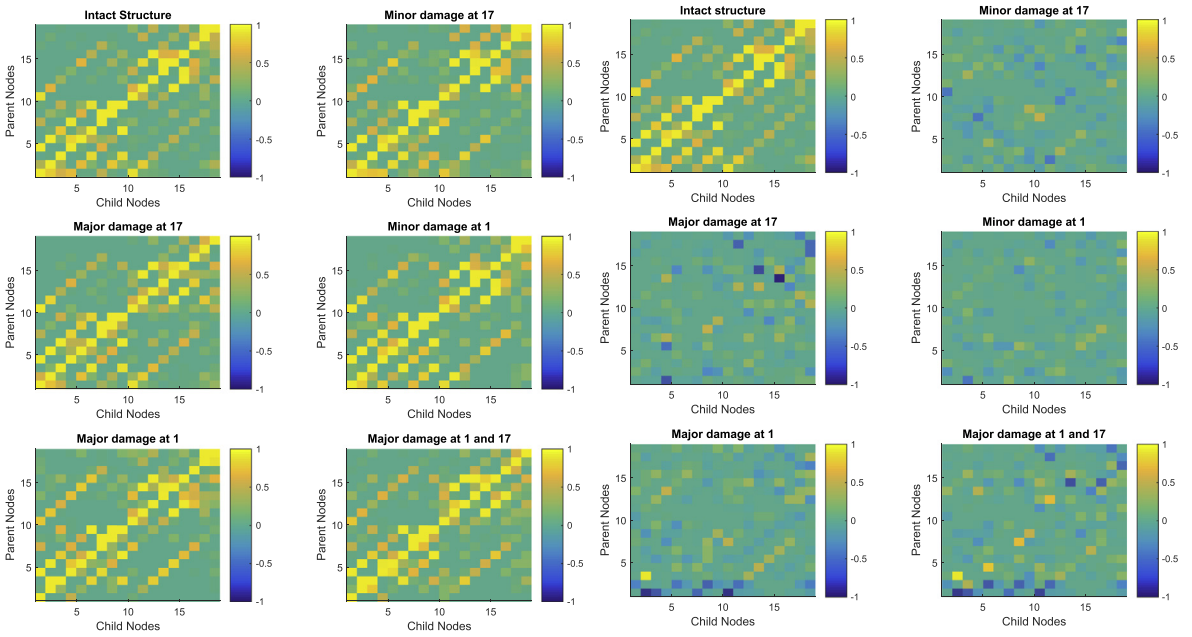


**Fig. 10.** (a) Probability of inferred parent-child relationships for intact and damaged cases obtained by averaging over many tests when number of parents is limited to 4. (b) Probability of edges in intact case and difference from intact case for damaged scenarios.

sensors in the same direction, and fewer between sensors in different directions. Most relationships are between the sensors that are located close to each other. There is also a fair number of relationships across the structure for the NS sensors.

A particularly interesting observation is the lack of relationships between the vertical sensors except for the pairs of 3–4 and 5–6. This may be explained by the rocking behavior found in the building [16], where sensors 3 and 4 move in phase, in opposition to sensors 5 and 6.

These relationships are also visualized in a matrix shown in Fig. 12. The sensors are grouped into vertical sensors, EW sensors, and then NS sensors, as given in the axis labels.

### *4.2. Multi-class classification*

We consider the problem of classification of sequences according to the structure condition, as described in Section 2.3. This problem is not directly applicable to real civil structures, as either damage has never occurred or it is unlikely that exactly the same damage scenario will occur in the future. However, it tells us how well the algorithm can distinguish not only damage from intact, but also different damage scenarios from each other. It is also worth noting that in some other damage detection problems, such as with machine parts, classification may actually be a realistic approach, as there may only be a handful of types of damages that typically occur and data from such scenarios may be available.

#### *4.2.1. 3-story 2-bay structure*

Recall that 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. To reduce computation, a subsequence of length 5000 is used from each sequence, except in the experiments that test the effect of training and test sequence lengths. Although the results with longer sequences may be slightly better, they are not qualitatively different.

Interestingly, we found that the best classification results are obtained when no additional parents (other than self) are allowed. Explaining this result requires further investigation. On the other hand, classification result on a single-column structure constructed from similar elements and which is excited by displacing it and let to vibrate freely are not degraded when additional parents are included [7].

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

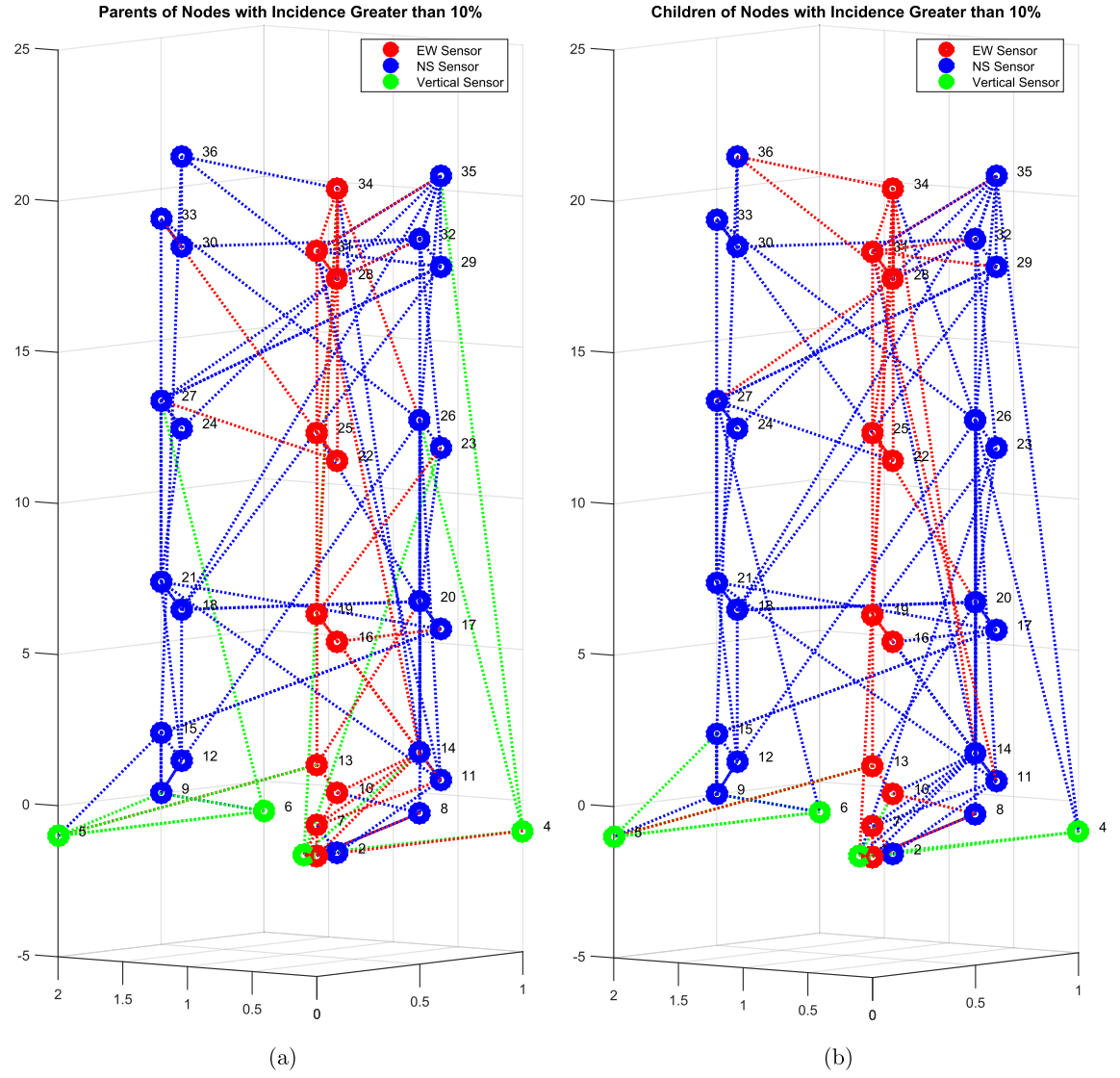


**Fig. 11.** 3D visualization of Green Building (a) node parents and (b) node children relationships with incidence over 10%.

to bring likelihoods closer to each other for the purpose of illustration, since the differences in likelihoods are huge in their original domain.

The resulting class-class log-likelihood matrix is shown in Fig. 13a. 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. 13b, 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 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 are shown in Fig. 13c and d. 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. 13c, as well as each group of bars in Fig. 13d, must sum to one. Overall, sequences are classified correctly most of the times (high diagonal values). 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

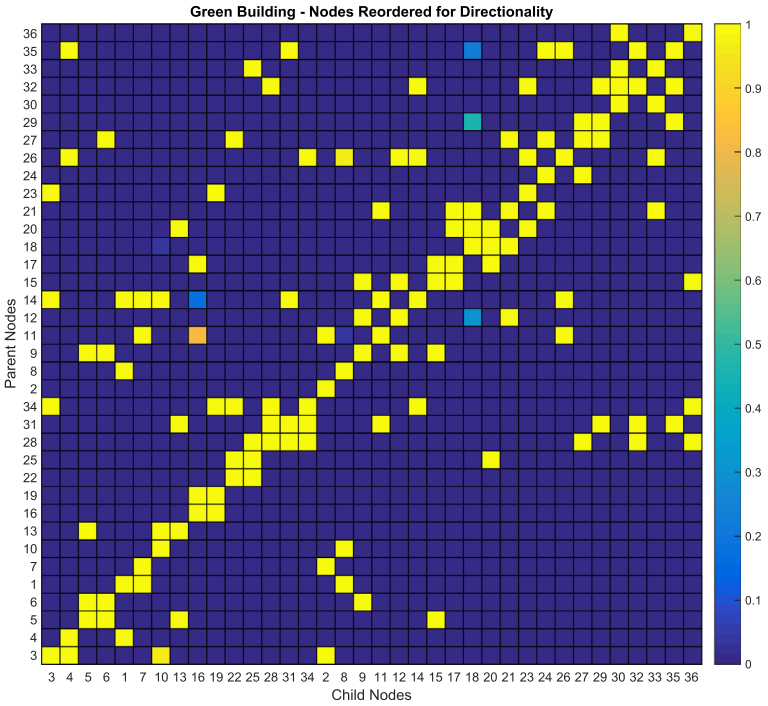


**Fig. 12.** Matrix visualization of node incidence for Green Building. The sensors are grouped into vertical sensors, EW sensors, and then NS sensors, as given in the axis labels. Concentration of high probability edges around the diagonal shows that many relationships are between the sensors in the same direction and close to each other.

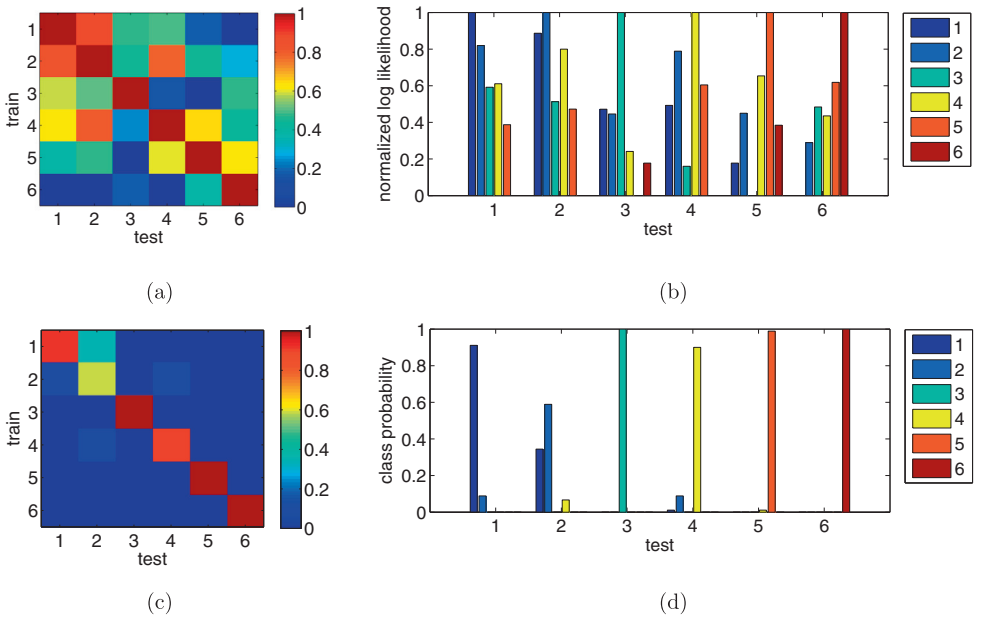


**Fig. 13.** 3 story 2 bay structure data class-class log-likelihoods are shown as a (a) matrix and (b) bars grouped by test class. Similarly, classification frequencies are shown as a (c) matrix and (d) bars grouped by test class.

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.

The overall classification accuracy as a function of training and test sequence lengths is shown in Fig. 14a. Three different training sequence lengths were used, 1000, 5000 and 10,000, while the test sequence length is varied from 1000 to 10,000.

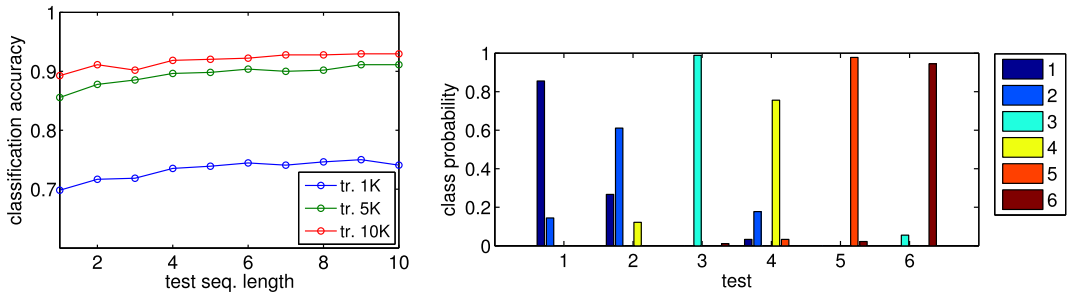


**Fig. 14.** (a) Overall classification accuracy on 3 story 2 bay structure data as a function of training and test sequence lengths. (b) Classification frequencies when training and test sequence lengths are $5K$ and $1K$, respectively.

Note that 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. This is in contrast to the results on a single-column structure from [7], which show that for a fixed length of a training sequence classification accuracy increases with test sequence length until test sequence length reaches training sequence length, after which classification accuracy decreases. This behavior could be attributed to the nature of excitation used in that setup, which is free vibration – i.e., there is no constant excitation over time and different parts of a same sequence may vary in behavior. Finally, for comparison with the results in Fig. 13d, in which both lengths are 5000, Fig. 14b shows classification results on the 3 story 2 bay structure data when training and test sequence lengths are 5000 and 1000, respectively.

### 4.3. Single-class classification

#### 4.3.1. 3-story 2-bay structure

As in the evaluation of multi-class classification above, subsequences of length 5000 are used for training and testing. For each training and each test sequence, the value $\frac{\mathcal{L}_2(\cdot)^{x^{test}})}{\mathcal{L}_1(\mathcal{Y}^{x^{test}}|\mathcal{Y}_1^{x_{tr}})}$ is computed, where labels 1 and 2 correspond to intact and damage classes, as in Section 2.4. The test sequence is classified as anomalous if this value is above some threshold $\epsilon_{dam}$. Note that this is equivalent to using Eq. (24), since the ratio $\frac{P_{tr}(Z^{test}=2)}{P_{tr}(Z^{test}=1)}$ is determined by the prior and can be absorbed into the threshold.

A receiver operating characteristic (ROC) curve [17], which represents the rate of true positives as a function of the rate of false positives, is computed separately for each damage scenario by varying the value of the threshold $\epsilon_{dam}$. Cross-validation is used to increase the number of training-test pairs. In each round, one sequence from intact scenario is considered as a training sequence, while the remaining 9 intact sequences and all 10 sequences from the chosen damage scenario are treated as test sequences. The number of false positives and the number true positives are computed as a function of $\epsilon_{dam}$ and aggregated over all rounds (i.e., over all choices of a training sequence).

Thus computed ROC curves for all damage classes are shown in Fig. 15. ROC curves are "perfect" for all major damage scenarios, in that there is a threshold for which all test sequences are correctly classified (i.e., the value $\frac{\mathcal{L}_2(\cdot)^{x^{test}})}{\mathcal{L}_1(\mathcal{Y}^{x^{test}}|\mathcal{Y}_1^{x_{tr}})}$ is below the threshold for all intact test sequences and it is above the threshold for all sequences from the damage case). Note that the ROC curves for scenarios 3 and 5 are not visible in Fig. 15 because they are overlayed by the curve for scenario 6. The ROC curve for the case of minor damage at node 1 (scenario 4) is close to perfect, while the worst result is for the case of minor damage at node 17 (scenario 2). This is not surprising, given that we already found in the previous section that most errors in a classification setting occur when an intact sequence is misclassified as belonging to scenario 2, and vice versa, i.e., that sequences from these two classes are most similar to each other.

In addition, for each damage scenario, a point that corresponds to the threshold value $\epsilon_{dam} = 1$ is shown in Fig. 15 with an 'x' mark. This threshold value corresponds to the posterior probability of a test sequence being damaged equal to 0.5, under the assumption that the prior probabilities of a test sequence being intact or damaged are equal. Note that there are no false positives in any of the scenarios. In other words, the posterior probability that a sequence is damaged is never higher than 0.5 for intact sequences. On the other hand, for the major damage scenarios, this probability is above 0.5 for almost all damaged sequences. However, only about 60% of damaged sequences have posterior probability of damage above 0.5 in case of the minor damage at node 1, and less than 15% of damaged sequences are classified as damaged by this rule in the case of the minor damage at node 17.

If one wants to devise a threshold rule in practice, the threshold that corresponds to the posterior probability of damage of 0.5 is not necessarily the right choice. From Fig. 15 we can see that, in the case of minor damages, this rule would classify a sequence as damaged only when it's very certain of it. If one wants to be less conservative and detect more damaged cases

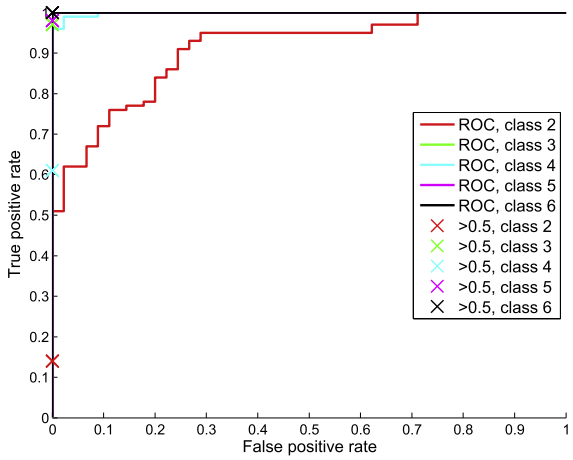


**Fig. 15.** ROC curves for each damage scenario on 3-story 2-bay structure data. Points on the curves that correspond to sequences with the posterior probability of damage above 0.5 being classified as damage are marked with an 'x'.

(at the expense of false positives), the threshold should be set to a lower value. However, choosing that number may not be as intuitive as one may expect. The likelihood of a sequence depends on its length approximately exponentially since the likelihoods of variables at each time point are multiplied together.[2] The ratio $\frac{\mathcal{L}_2(\mathcal{Y}^{test})}{\mathcal{L}_1(\mathcal{Y}^{test}\mid\mathcal{Y}_1^{tr})}$, which is used to discriminate damaged from intact sequences, approaches 0 or infinity exponentially with sequence length, depending on whether the test sequence is more likely under the posterior model given the training sequence or under the prior model. In an ideal case, if the model perfectly matches the data, one could simply "trust" these probabilities – i.e., if the model tells that the probability of damage is 1, that would indeed mean that there is almost certainly damage, and, similarly, sequences with posterior probability of damage close to 0 would almost certainly correspond to an intact structure. However, due to the fact that the statistical model is only an approximation to the physical model, some sequences from a damage scenario may actually have low posterior probability of damage or some sequences from intact scenario may have high probability of damage. From the results above, we see that the former is the case for the 3-story 2-bay structure data.

One approach to compensate for the effects of sequence length and model mismatch is to adjust the threshold to account for them. However, that is a very hard problem, as it is difficult to quantify these effects precisely (or even approximately). Instead, we take a data driven approach to choosing a threshold. Since the assumption is that the data from a damage scenario is not available a priori, we can only use the data from the intact scenario. Specifically, we assume that one intact sequence is used as a training sequence, 8 intact sequences are used for tuning, and the remaining intact sequence is used for testing (along with all ten sequences from a damage scenario that is tested). First, the value of the ratio $\frac{\mathcal{L}_2(\mathcal{Y}^{test})}{\mathcal{L}_1(\mathcal{Y}^{test}\mid\mathcal{Y}_1^{tr})}$ is computed for all tuning sequences. Let $L_1^{tune},\ldots,L_8^{tune}$ denote these values. A threshold is computed as a function of these values, which is then applied to classify the test sequences. This is repeated for all possible choices of a training sequence and tuning sequences among intact data, and the results are aggregated (which we refer to as "cross-validation" in this context). It remains to discuss how to choose the threshold $\epsilon_{dam}$ as a function of values $L_1^{tune},\ldots,L_8^{tune}$. One possibility is to use the maximum of these values, which would result in low false positive rates, and, if the damage sequences are relatively different from intact sequences, would result in a large true positive rate. More generally, if these values are sorted such that $L_1^{tune}>L_2^{tune}>\ldots>L_8^{tune}$, then, choosing a threshold that is between $i^{\text{th}}$ and $(i+1)^{st}$ value would approximately result in the false positive rate of $i/8$. Therefore, the false positive rate can be controlled even though the corresponding rate of true positives is not known a priori. Another approach is to assume that these values come from a Gaussian distribution and compute their empirical mean and standard deviation. The threshold can then be set as $\mathrm{E}L_i^{tune}+\lambda\sigma L_i^{tune}$, for some value of $\lambda$ (e.g., $\lambda=2$ would correspond to taking two standard deviations away from the mean). Fig. 16 shows the tradeoff between the rates of true positives and false positives for these two approaches in the case of minor damage at node 17 (scenario 2). Fig. 16a shows the tradeoff points when the threshold is set to $L_1^{tune},\ldots,L_8^{tune}$, respectively, assuming that these values are sorted in the decreasing order. Fig. 16b shows the tradeoff points for various values of $\lambda$ when he threshold is set to $\mathrm{E}L_i^{tune}+\lambda\sigma L_i^{tune}$. Note that the points in both figures do not necessarily fall on the ROC curve because thresholds are a function of tuning sets and are therefore not necessarily uniform across all training-tuning sets.

[2] Technically, this is the case for a specific value of model structure and parameters, and the overall likelihood is obtained by summing/integrating over possible values of structures and parameters, weighted by their prior.

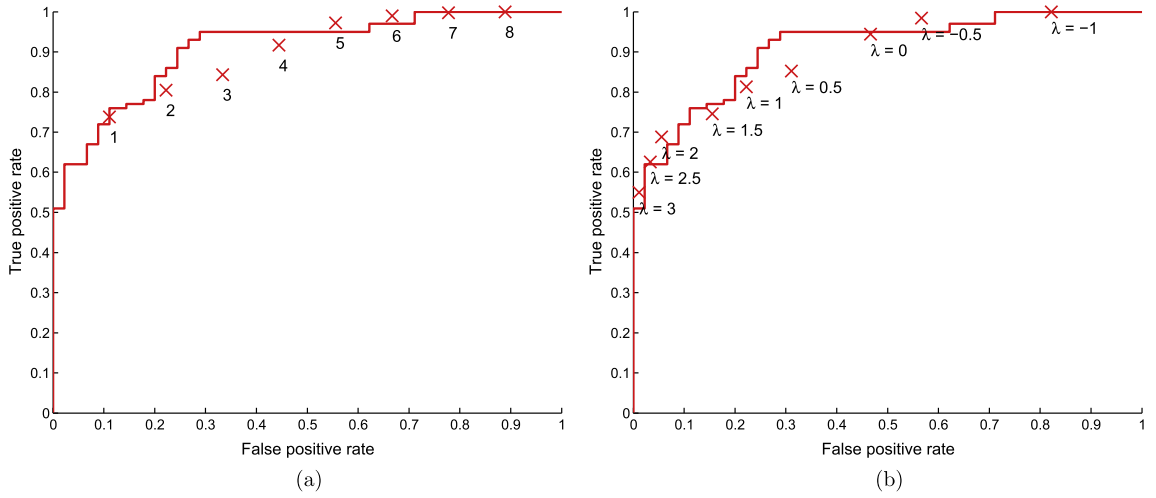


**Fig. 16.** Points of tradeoff between the rates of true positives and false positives when: (a) The threshold is set to $L_1^{tune} > L_2^{tune} > \ldots > L_8^{tune}$, respectively. (b) The threshold is set to $EL_i^{tune} + \lambda\sigma L_i^{tune}$ for different values of $\lambda$.

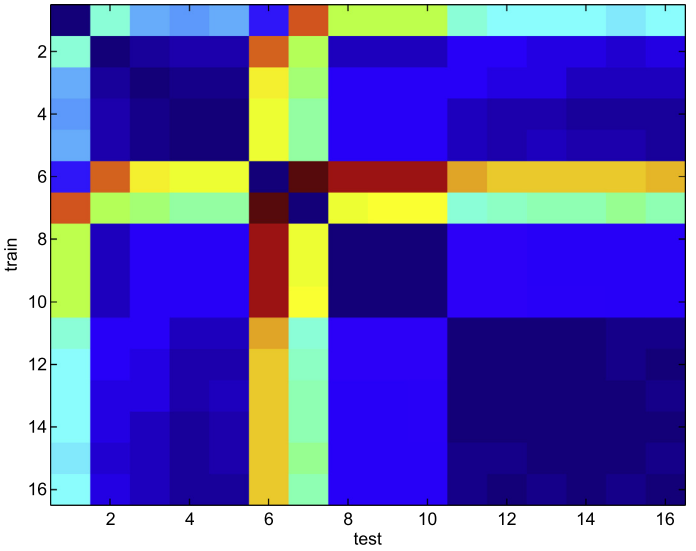


**Fig. 17.** Matrix of the log-likelihood ratios, $\log\frac{\mathcal{L}_2(\mathcal{Y}^{test})}{\mathcal{L}_1(\mathcal{Y}^{test}|\mathcal{Y}_1^{tr})}$, between Green Building data sequences, normalized to be between 0 and 1. The value at row $i$ and column $j$ corresponds to the ratio computed when sequence $i$ is considered as a training sequence and sequence $j$ as a test sequence. The correspondence between sequence indices and events is: 5/14/2012 Unknown Event (1), 6/22/2012 Ambient Event (2–3), Fireworks (4–6), Earthquake (7), 4/15/2013 Ambient Event (8–10), and Windy Day (11–16). Note that the events that are the most similar to each other are the events in ambient conditions, windy conditions, but also the first two sequences for the fireworks event, which were recorded before the fireworks actually started. On the other hand, the last sequence in the fireworks test case, the earthquake, and the 5/14/2012 event test cases all have significantly higher likelihood ratios with respect to the ambient cases. These results suggest that we can likely classify when the structure has been excited in a significantly different way than typical ambient conditions.

### *4.3.2. Green Building*

Fig. 17 shows the matrix of the logarithm of likelihood ratios, $\log\frac{\mathcal{L}_2(\mathcal{Y}^{test})}{\mathcal{L}_1(\mathcal{Y}^{test}|\mathcal{Y}_1^{tr})}$, normalized to be between 0 and 1 for the visualization purpose. The value at row $i$ and column $j$ corresponds to the ratio computed when sequence $i$ is considered as a training sequence and sequence $j$ as a test sequence. Recall from Eq. (24) that this ratio can be used to discriminate sequences that behave differently from the training sequence. The higher the value of the ratio is, the more likely it is that the test sequence will be labeled differently from the training sequence.

We can see that the events that are the most similar to each other are the events in ambient conditions, windy conditions, but also the first two sequences for the fireworks event. For the fireworks event, when the recording was made during the Boston July 4th fireworks show, only the last sequence of the three occurs during when the fireworks are being set off. The

first two sequences are of the normal ambient structure, and thus they have low likelihood ratio with respect to the other ambient structure test cases. The windy condition measurements are not as dissimilar from the ambient measurements as we would have expected as winds were sustained at 20mph with gusts at higher speeds. The accelerations measured however are likely similar to ambient conditions with slightly higher magnitudes, as the winds are random excitations.

The last sequence in the fireworks test case, the earthquake, and the 5/14/2012 event test cases all have significantly higher likelihood ratios with respect to the ambient cases. What's interesting is that the fireworks sequence is similar to the 5/14/2012 event, but both are dissimilar to the earthquake case. The 5/14/2012 event was when the recording system was triggered to record because accelerations exceeded a preset threshold, however there is no known event that corresponds to it. The time-series looks like a single impulse, possibly suggesting similar behavior induced in the structure to the series of impulses from the fireworks sequences. The third fireworks sequence seems to be the most dissimilar from all the other sequences.

What these results tell us is that we can likely classify when the structure has been excited in a significantly different way than typical ambient conditions. The differences between random ambient excitations and impulse excitations or earthquake excitations are clearly visible. We do not evaluate the performance of the single-class classification formally, as we did with the laboratory data using ROC curves, since the number of recorded sequences for the Green Building is not that large. However, it is clear from the likelihood ratio matrix in Fig. 17 that using any of the ambient sequences as a training sequence and a reasonable threshold rule (e.g., use other ambient sequences as tuning data and take the highest ratio among them as a threshold) would perfectly classify the sequences from the earthquake and the 5/14/2012 event and the third fireworks sequence as non-ambient. Sequences from the windy condition would also be classified as non-ambient in most cases, except when the second sequence of the 6/22/2012 ambient recording is used for training. In that case, the sequences from the windy condition have lower likelihood ratio than those from the 4/15/2013 ambient event. If the latter ones are used for tuning, the former ones would be classified as ambient. Also, note that the two ambient recordings are slightly different from each other, which could possibly be attributed to the temperature difference of 40°F between these two recordings. This suggests that acquiring more ambient recordings over time and in different conditions would be useful to understand the variation in them and how that relates to the classification problem.

## 5. Conclusion

In this paper we presented an approach using Bayesian inference on a state-space switching interaction model to detect and classify changes indicative of either damage or a change in excitation in structures. We applied the methodology developed in Dzunic et al. [6] to data from two structures, a laboratory model and the MIT Green Building as a test for structural health monitoring. Inference was done over dependence structures in the model, and it was determined that the parents of a node were most likely physically connected by a structural element to that node, possibly providing information about the physical structure between sensor locations. Damage detection was accomplished by obtaining the log-likelihoods of test sequences given a different training sequence. Test data was classified correctly to their respective damage scenarios or the baseline structure case with relatively high accuracy. Classification was also accomplished in a single class methodology more realistic for structural health monitoring where training data from a damaged structure may not be available. Future work involves testing the approach on more varied structural configurations, damage scenarios, and in environmental conditions. Continuous vibration records from a structure would be useful for further testing to explore the switching interaction model. Another approach for classification might involve using the edges in the dependency graph and should also be tested.

## Acknowledgements

The authors acknowledge the support provided by Royal Dutch Shell through the MIT Energy Initiative, and thank chief scientists Dr. Dirk Smit and Dr. Sergio Kapusta, project managers Dr. Keng Yap and Dr. Lorna Ortiz-Soto, 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 Dr. Young-Jin Cha for assistance in collecting the experimental data.

## Appendix A. Matrix normal inverse Wishart prior

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

$$X_t = AX_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, Q), \tag{A.1}$$

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 $\mathcal{N}(X_t; AX_{t-1}, Q)$:

$$p(A, Q; M, \Omega, \Psi, \kappa) = \mathcal{MN} - \mathcal{IW}(A, Q; M, \Omega, \Psi, \kappa) = \mathcal{MN}(A; M, Q, \Omega)\mathcal{IW}(Q; \Psi, \kappa). \tag{A.2}$$

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

$$\mathcal{MN}(A;M,Q,\Omega)=\frac{\exp\left(-\frac{1}{2}\mathrm{tr}\left[\Omega^{-1}(A-M)^{T}Q^{-1}(A-M)\right]\right)}{(2\pi)^{dl/2}|\Omega|^{d/2}|Q|^{l/2}}, \tag{A.3}$$

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

$$\mathcal{IW}(Q;\Psi,\kappa)=\frac{|\Psi|^{\kappa/2}}{2^{\kappa d/2}\Gamma_{d}(\kappa/2)}|Q|^{-(\kappa+d+1)/2}\exp(-\frac{1}{2}\mathrm{tr}(\Psi Q^{-1})), \tag{A.4}$$

where $d$ is the dimension of matrix $Q$ ($Q_{d\times d}$) and $\Gamma_{d}()$ is a multivariate gamma function, while $\kappa$ 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:

$$p(A,Q|X_{0:T};M,\Omega,\Psi,\kappa)=\mathcal{MN}-\mathcal{IW}(A,Q;M',\Omega',\Psi',\kappa')=\mathcal{MN}(A;M',Q,\Omega')\mathcal{IW}(Q;\Psi',\kappa'), \tag{A.5}$$

where

$$\begin{aligned}
\Omega' &= \left(\Omega^{-1}+\sum_{t=0}^{T-1}X_tX_t^T\right)^{-1}\\
M' &= \left(M\Omega^{-1}+\sum_{t=1}^{T}X_tX_{t-1}^T\right)\Omega'\\
\kappa' &= \kappa+T\\
\Psi' &= \Psi+\sum_{t=1}^{T}X_tX_t^T+M\Omega^{-1}M^T-M'\Omega'^{-1}M'^T.
\end{aligned} \tag{A.6}$$

## References


[1] J.M. Brownjohn, Structural health monitoring of civil infrastructure, Philos. Trans. Roy. Soc. A: Math. Phys. Eng. Sci. 365 (1851) (2007) 589–622.
[2] H. Sohn, C.R. Farrar, F.M. Hemez, D.D. Shunk, D.W. Stinemates, B.R. Nadler, J.J. Czarnecki, A Review of Structural Health Monitoring Literature: 1996–2001, Los Alamos National Laboratory Los Alamos, NM, 2004.
[3] J.L. Beck, L.S. Katafygiotis, Updating models and their uncertainties. i: Bayesian statistical framework, J. Eng. Mech. 124 (4) (1998) 455–461.
[4] M.W. Vanik, J. Beck, S. Au, Bayesian probabilistic approach to structural health monitoring, J. Eng. Mech. 126 (7) (2000) 738–745.
[5] E.B. Flynn, M.D. Todd, A bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing, Mech. Syst. Signal Process. 24 (4) (2010) 891–903.
[6] Z. Dzunic, J. Fisher III, Bayesian switching interaction analysis under uncertainty, in: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, 2014, pp. 220–228.
[7] Z. Dzunic, J.G. Chen, H. Mobahi, O. Buyukozturk, J.W. Fisher III, A bayesian state-space approach for damage detection and classification, Dynamics of Civil Structures, vol. 2, Springer, 2015, pp. 171–183.
[8] D. Heckerman, A tutorial on learning with bayesian networks, Tech. Rep. MSR-TR-95-06, Microsoft Research, March 1995.
[9] Z. Ghahramani, Learning dynamic bayesian networks, in: Adaptive Processing of Sequences and Data Structures, Springer, 1998, pp. 168–197.
[10] D.M. Chickering, Learning Bayesian networks is NP-Complete, in: D. Fisher, H. Lenz (Eds.), Learning from Data: Artificial Intelligence and Statistics V, Springer-Verlag, 1996, pp. 121–130.
[11] W. Buntine, Theory refinement on bayesian networks, in: Proceedings of the Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-91), Morgan Kaufmann, San Mateo, CA, 1991, pp. 52–60.
[12] G.F. Cooper, T. Dietterich, A bayesian method for the induction of probabilistic networks from data, in: Machine Learning, 1992, pp. 309–347.
[13] D. Heckerman, D. Geiger, D.M. Chickering, Learning bayesian networks: the combination of knowledge and statistical data, in: Machine Learning, 1995, pp. 197–243.
[14] N. Friedman, D. Koller, Being Bayesian about Bayesian network structure: a Bayesian approach to structure discovery in Bayesian networks, Machine Learn. 50 (1–2) (2003) 95–125, full version of UAI 2000 paper.
[15] M.R. Siracusa, J.W. Fisher III, Tractable bayesian inference of time-series dependence structure, in: Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, 2009.
[16] M. Çelebi, N. Toksöz, O. Büyüköztürk, Rocking behavior of an instrumented unique building on the mit campus identified from ambient shaking data, Earthq. Spectra 30 (2) (2014) 705–720.
[17] T. Fawcett, An introduction to ROC analysis, Pattern Recogn. Lett. 27 (8) (2006) 861–874.