Article:
Bull, L.A., Gardner, P., Rogers, T.J. et al. (2021) Probabilistic inference for structural health monitoring: new modes of learning from data. ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering, 7 (1). 03120003. ISSN: 2376-7642

https://doi.org/10.1061/ajrua6.0001106

PROBABILISTIC INFERENCE FOR STRUCTURAL HEALTH MONITORING: NEW MODES OF LEARNING FROM DATA

Lawrence A. Bull, Paul Gardner, Timothy J. Rogers
Elizabeth J. Cross, Nikolaos Dervilis, Keith Worden
Dept. of Mech. Eng., Univ. of Sheffield, Mappin St., Sheffield, S1 3JD, UK l.a.bull@sheffield.ac.uk

https://doi.org/10.1061/AJRUA6.0001106

ABSTRACT

In data-driven SHM, the signals recorded from systems in operation can be noisy and incomplete. Data corresponding to each of the operational, environmental, and damage states are rarely available *a priori*; furthermore, labelling to describe the measurements is often unavailable. In consequence, the algorithms used to implement SHM should be robust and adaptive, while accommodating for missing information in the training-data – such that new information can be included if it becomes available. By reviewing novel techniques for statistical learning (introduced in previous work), it is argued that probabilistic algorithms offer a natural solution to the modelling of SHM data in practice. In three case-studies, probabilistic methods are adapted for applications to SHM signals — including semi-supervised learning, active learning, and multi-task learning.

Keywords: Structural health monitoring, statistical machine learning, pattern recognition, semi-supervised learning, active learning, multi-task learning, transfer learning

PROBABILISTIC SHM

Under the pattern recognition paradigm associated with Structural Health Monitoring (SHM) (Farrar and Worden 2012), data-driven methods have been established as a primary focus of research. Various machine learning tools have been applied in the literature, for example (Vanik et al. 2000; Sohn et al. 2003; Chatzi and Smyth 2009), and used to infer the health or performance state of the monitored system, either directly or indirectly. Generally, algorithms for regression, classification, density estimation, or clustering learn patterns in the measured signals (available for training), and the associated patterns can be used to infer the state of the system in operation, given future measurements (Worden and Manson 2006).

Unsurprisingly, there are numerous ways to apply machine learning to SHM. Notably (and categorised *generally*), advances have focussed on various probabilistic (e.g. (Vanik et al. 2000; Ou et al. 2017; Flynn and Todd 2010)) and deterministic (e.g. (Bornn et al. 2009; Zhao et al. 2019; Janssens et al. 2017)) methods. Each approach has its advantages; however, considering certain challenges associated with SHM data (outlined in the next section) the current work focusses on probabilistic (i.e. statistical) tools: these algorithms appear to offer a natural solution to some key issues, which can otherwise prevent practical implementation. Additionally, probabilistic methods can lead to predictions *under uncertainty* (Papoulis 1965) – a significant advantage in risk-based applications.

SHM, Uncertainty, and Risk

It should be clear that measured/observed data in SHM will be inherently uncertain, to some degree. Uncertainties can enter via *experimental* sources, including limitations to sensor accuracy, precision or human error; further uncertainties will be associated with the model – machine learning or otherwise – including parametric variability, model discrepancy, and interpolation uncertainty. Considering the implications of *risk*, financially and in terms of safety, uncertainty should be mitigated (during data acquisition), and quantified (within models) as far as possible to inform decision making (Zonta et al. 2014; Cappello et al. 2015). That is, when supporting a financial or safety-critical decision, predictions should be presented with *confidence*: clearly, a certain prediction, which implies a system is safe to use, differs significantly to an *uncertain* prediction, supporting the same decision. If there is no attempt to quantify the associated uncertainties, there is no distinction between these scenarios.

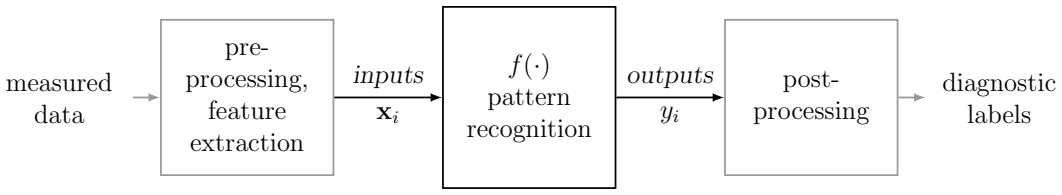


FIG. 1: A *simplified* framework for pattern recognition within SHM.

Various methods can return predictions with confidence (or *credibility*) (Murphy 2012). The current work focusses on probabilistic models, which – under Kolmogorov's axioms (Papoulis 1965) – allow for predictions under well-defined uncertainty, provided the model assumptions are *appropriate*.

A Probabilistic Approach

Discussions in this work will consider the general strategy illustrated in Figure 1. That is, SHM is viewed as a multi-class problem, which categorises measured data into groups, corresponding to the condition of the monitored system. The i^{th} input, denoted by $\mathbf{x}_i$, is defined by a d-dimensional vector of variables, which represents an *observation* of the system, such that $\mathbf{x}_i \in \mathbb{R}^d$. The data *labels* y_i, are used to specify the condition of the system, directly or indirectly. Machine learning is introduced via the pattern recognition model, denoted $f(\cdot)$, and is used to infer relationships between the input and output variables, to inform predictive maintenance.

The inputs $\mathbf{x}_i$ are assumed to be represented by some random vector X (in this case, a continuous random vector), which can take any value within a given feature-space $\mathcal{X}$. The random vector is therefore associated with an appropriate probability density function (p.d.f.), denoted $p(\cdot)$, such that the probability P of X falling within the interval $a < X \leq b$ is, $P\left(a < X \leq b\right) = \int_a^b p\left(\mathbf{x}_i\right) d\mathbf{x}_i$ such that $p\left(\mathbf{x}_i\right) \geq 0$, $\int_{\mathcal{X}} p\left(\mathbf{x}_i\right) d\mathbf{x}_i = 1$. For a discrete classification problem, the labels y_i are represented by a discrete random variable Y, which can take any value from the finite set, $y_i \in \mathcal{Y} = \{1, ..., K\}$. Note: discrete classification is presented in this work, although, SHM is regularly informed by regression models – i.e. y_i is continuous; this is application specific, and most of the motivational arguments remain the same. K is the number of classes defining the (observed) operational, environmental, and health conditions, while $\mathcal{Y}$ denotes the label-space. An appropriate probability mass function (p.m.f.), also denoted $p(.)$, is such that, $P\left(Y = y_i\right) = p(y_i)$ where $0 \leq P\left(Y = y_i\right) \leq 1$, $\sum_{y_i \in Y} P\left(Y = y_i\right) = 1$.

Note: context should make the distinction between p.m.fs and p.d.fs clear. Further details regarding probability theory for pattern recognition can be found in a number of well written textbooks – for example (Murphy 2012; Barber 2012; Gelman et al. 2013).

Layout

Section 2 summarises the most significant challenges for data-driven SHM, while Section 3 suggests probabilistic methods to mitigate these issues. Section 4 introduces theory behind directed graphical models (DGMs), which will be used to formally introduce each method. Section 5 collects four case studies to highlight the advantages of probabilistic inference. Active learning and Dirichlet process clustering are applied to the Z24 bridge data. Semi-supervised learning is applied to data recorded during ground vibration tests of a Gnat aircraft. Multi-task learning is applied simulated and experimental data from shear-building structures.

Note: the applications presented here were introduced in previous work by the authors. The related SHM literature is referenced in the descriptions of each mode of inference.

INCOMPLETE DATA AND MISSING INFORMATION

Arguably, the most significant challenge when implementing pattern recognition for SHM is missing information. Primarily, it is difficult to collect data that might represent damage states or the system in extreme environments (such as earthquakes) *a priori*; data are usually only available for a limited subset of the possible conditions for training algorithms (Farrar and Worden 2012). As a result, conventional methods are restricted to novelty detection, as the information required to inform *multi-class* predictive models (that can localise and classify damage, as well as detect it (Worden and Manson 2006)) is unavailable or not obtained.

For the measurements $\mathbf{x}_i$ that are available – as well as those that are recorded during operation (*in situ*) – *labels* to describe what the signals represent, y_i, are rarely at hand. This missing information is usually due to the cost associated with manually inspecting structures (or data), as well as the practicality of investigating each observation. The absence of labels makes defining and updating (multi-class) machine learning models difficult, particularly in the online setting, as it can become difficult to determine if/when novel valuable information has been recorded, and what it represents (Bull et al. 2019b). For example, consider

streaming data, recorded from a sub-sea pipeline. Comparisons of measured data to the model might indicate novelty; however, without labels, it is difficult to include this new information in a supervised manner: the measurements might represent another operational condition, abnormal wave loads, actual damage, or some other condition.

NEW MODES OF PROBABILISTIC INFERENCE

New modes of probabilistic inference are being proposed to address challenges with SHM data. Specifically, the algorithms focus on probabilistic frameworks to deal with *limited labelled data*, as well as *incomplete measured data*, that only correspond to a subset of the expected conditions *in situ*.

Partially-Supervised Learning

Partially-supervised learning allows multi-class inference in cases where labelled data are limited. Missing label information is especially relevant to practical applications of SHM: while *fully* labelled data are often infeasible, it can be possible to include labels for a limited set (or *budget*) of measurements. Typically, the budget is limited by some expense incurred when investigating the signals; this might include direct costs associated with inspection, or loss of income due to down-time (Bull et al. 2020b).

Generally speaking, partially-supervised methods can be used to perform multi-class classification, while utilising *both* labelled $\mathcal{D}_l$ and unlabelled $\mathcal{D}_u$ signals within a *unifying* training scheme (Schwenker and Trentin 2014) – as such, the training set $\mathcal{D}$ becomes,

$$\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u \tag{1}$$

$$= \{\mathbf{X}, \mathbf{y}\} \cup \tilde{\mathbf{X}} \tag{2}$$

$$\{\mathbf{X}, \mathbf{y}\} \triangleq \{\mathbf{x}_i, y_i\}_{i=1}^{n} \tag{3}$$

$$\tilde{\mathbf{X}} \triangleq \{\tilde{\mathbf{x}}_i\}_{i=1}^{m} \tag{4}$$

Active and *semi-supervised* techniques are suggested – as two variants of partially-supervised learning – to combine/include information from labelled and unlabelled SHM data (Bull et al. 2018; Bull et al. 2019b; Bull et al. 2020b).

Semi-supervised learning

Semi-supervised learning utilises *both* the labelled and unlabelled data to inform a classification *mapping*, $f : \mathcal{X} \mapsto \mathcal{Y}$. Often, a semi-supervised learner will use information in $\mathcal{D}_u$ to further update/constrain a classifier learnt from $\mathcal{D}_l$ (McCallumzy and Nigamy 1998), or, alternatively, partial supervision can be implemented as constraints on a *unsupervised* clustering algorithm (Chapelle et al. 2006). This work focusses on classifier-based methods; however, constraints on clustering algorithms are discussed in later sections.

Arguably, the most simple/intuitive method to introduce unlabelled data is *self-labelling* (Zhu 2005). In this case, a classifier is trained using $\mathcal{D}_l$, which is used to predict labels for the unlabelled set $\mathcal{D}_u$. This defines a new training-set – some labels in $\mathcal{D}$ are the ground truth, from the supervised data, and the others are *pseudo-labels*, predicted by the classifier. Self-labelling is simple, and it can be applied to any supervised method; however, the effectiveness is highly dependent on the method of implementation, and the supervised algorithm within it (Chapelle et al. 2006).

Generative mixture models offer a formal *probabilistic* framework to incorporate unlabelled data (Cozman et al. 2003; Nigam et al. 1998). Generative mixtures apply the cluster assumption: *'if points are in the same cluster, they are likely to be of the same class*'. Note: the cluster assumption does not necessarily imply that each class is represented by a single, compact cluster; instead, the implication is that observations from different classes are unlikely to appear in the same cluster (Chapelle et al. 2006). Through density estimation (Barber 2012), a mixture of base-distributions can be used to estimate the underlying distribution of the data, $p(\mathbf{x}_i, y_i)$, and unlabelled observations can be included in various ways (McCallumzy and Nigamy 1998; Vlachos et al. 2009). For example, the Expectation Maximisation (EM) algorithm (used to learn mixture models in the unsupervised case (Murphy 2012)) can be modified to incorporate labelled observations (Nigam et al. 1998; McCallumzy and Nigamy 1998). Figure 2 demonstrates how a Gaussian mixture, given acoustic emission data (Rippengill et al. 2003), can be improved by considering the surrounding unlabelled examples (via EM).

To summarise, semi-supervised methods allow algorithms to learn from information in the available unlabelled measurements as well as a limited set of labelled data. In practice, semi-supervised inference implies that the cost associated with labelling data could be managed in SHM (Chen et al. 2013; Chen et al.

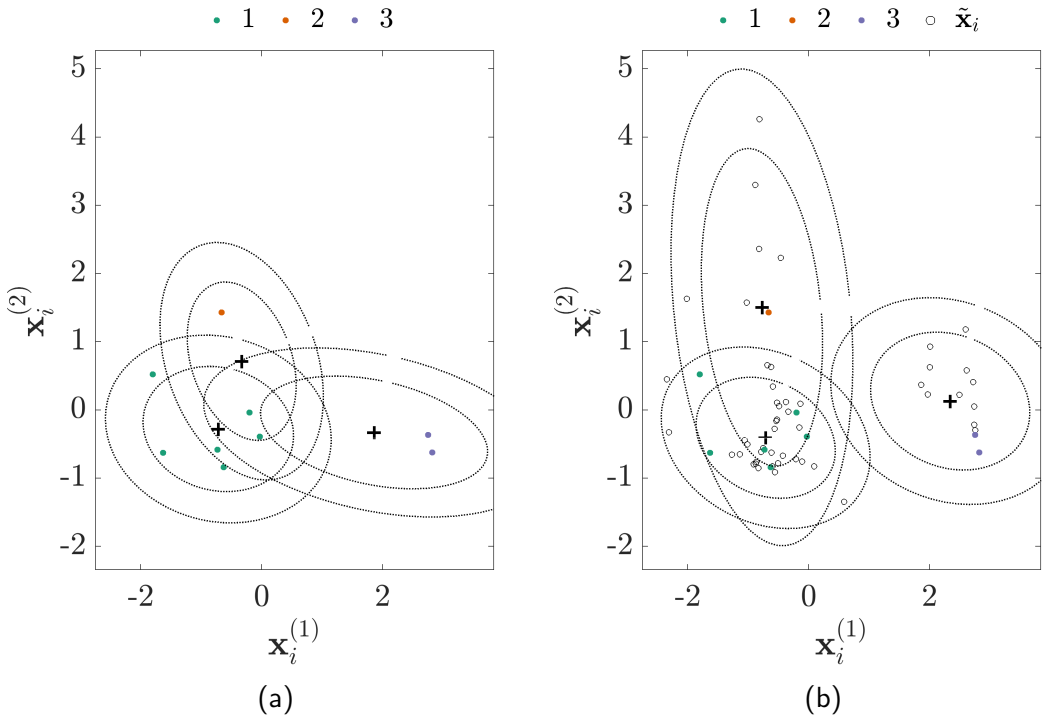


FIG. 2: Semi-supervised GMM for three-class AE data: (a) supervised learning, given the labelled data only, • markers. (b) semi-supervised learning, given the labelled *and* unlabelled data, •/○ markers. Adapted from (Bull 2019).

2014), as the information in a small set of labelled signals is combined with larger sets of unlabelled data (Bull et al. 2019c).

Active Learning

Active learning is an alternative partially-supervised method; the key hypothesis is that an algorithm can provide improved performance, using fewer training labels, if it is allowed to select the data from which it learns (Settles 2012). As with semi-supervised techniques, the learner utilises $\mathcal{D}_l$ and $\mathcal{D}_u$ – however, active algorithms query/annotate the unlabelled data in $\mathcal{D}_u$ to extend the labelled set $\mathcal{D}_l$. Thus, an active learner attempts to define an accurate mapping, $f : \mathcal{X} \mapsto \mathcal{Y}$, while keeping queries to a minimum (Dasgupta 2011); general (and simplified) steps are illustrated in Figure 3.

The critical step for active algorithms is how to select the most informative signals to investigate (Wang et al. 2017; Schwenker and Trentin 2014). For example, *Query by Committee (QBC)* methods build an ensemble/committee of

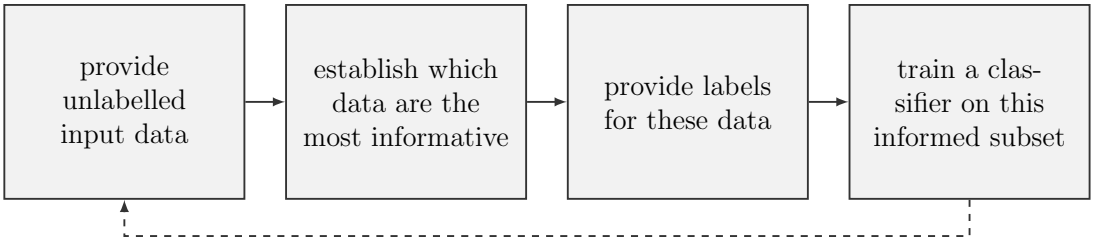


FIG. 3: A general/simplified active learning heuristic.

classifiers using a small, initial (random) sample of labelled data, leading to multiple predictions for unlabelled instances. Observations with the most conflicted label predictions are viewed as informative, thus, they are queried (Wang et al. 2017). On the other hand, *uncertainty-sampling* usually refers to a framework that is based around a single classifier (Kremer et al. 2014; Settles 2012), where signals with the *least confident* predicted label, given the model, are queried. (It is acknowledged that QBC methods can also be viewed as a type of uncertainty sampling.) Uncertainty sampling is (perhaps) most interpretable when considering probabilistic algorithms, as the posterior probability over the class-labels $p(y_i \,|\, \mathbf{x}_i)$ can be used to quantify uncertainty/confidence (Bull et al. 2020c). For example, consider a binary (two-class) problem: intuitively, uncertain samples could be instances whose posterior probability is nearest to 0.5 for both classes. This view can be extended to multiple (> 2) classes using the *Shannon entropy* (MacKay 2003) as a measure of uncertainty; i.e. high entropy (uncertain) signals given the GMM of the acoustic emission data (Rippengill et al. 2003) is illustrated in Figure 4a.

In summary, as label information is limited by cost implications in practical SHM (Bull et al. 2019a), active algorithms can be utilised to automatically administer the label budget, by selecting the most *informative* data to be investigated – such that the performance of predictive models is maximised (Bull et al. 2019d).

Dirichlet Process Mixture Models for Nonparametric clustering

Dirichlet Process (DP) mixture models (Neal 2000) offer another probabilistic framework to deal with limited labels as well as incomplete data *a priori*. The DP is suggested as an (unsupervised) Bayesian algorithm for nonparametric clustering, used to perform inference online such that the need for extensive training-data (before implementing the SHM strategy) is mitigated (Rogers et al.

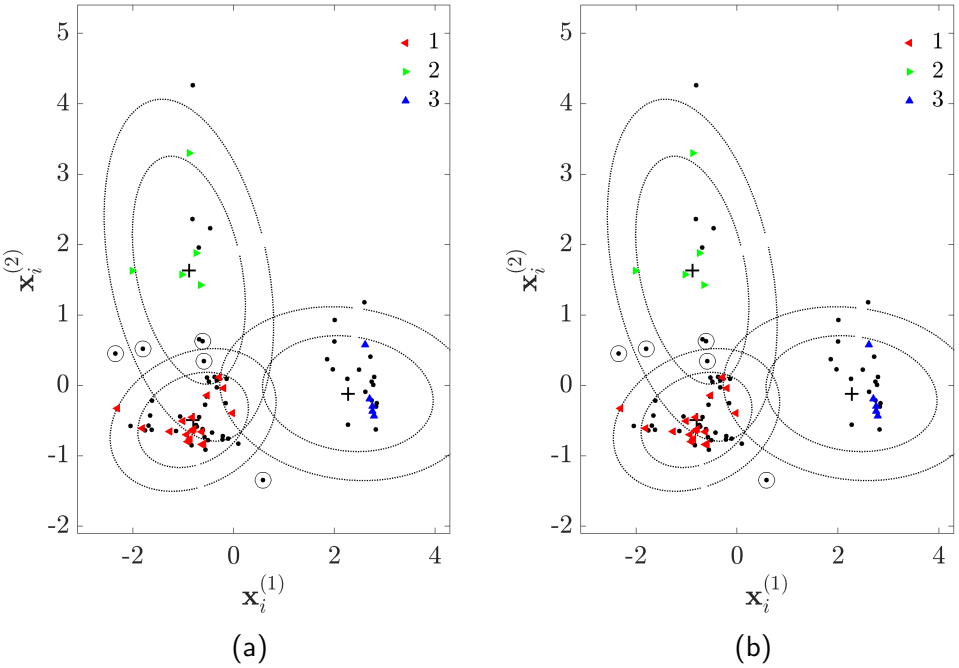


FIG. 4: Uncertainty sampling for the AE data: ► ◄ ▼ markers show the training set, and • markers show the unlabelled data – circles indicate queries by the active learner (a) based on entropy, (b) based on likelihood – adapted from (Bull 2019).

2019). As such, unlike partially-supervised methods, labels are always an additional *latent* variable (they are never observed); thus, the ground truth of y_i is not known during inference. Label information has the potential to be incorporated, however; either within the SHM strategy (Rogers et al. 2019), or at the algorithm level to define a semi-supervised DP (Vlachos et al. 2009).

Conveniently, Bayesian properties of the DP allow the incorporation of prior knowledge and updates of belief, given the observed data. The aim is to avoid the need for comprehensive training-data, while retaining flexibility to include any available data formally as prior knowledge. Additionally, as there is a reduction in the number of user-tuned parameters, models can be implemented to perform powerful online learning with minimal *a priori* input/knowledge, in terms of access to data or a physical model (Rogers et al. 2019).

Dirichlet Process Clustering

A popular analogy to describe the DP (for clustering) considers a restaurant with an infinite number of tables (Aldous 1985) (i.e. clusters in $\mathcal{Y}$). Customers –

resembling observations in $\mathfrak{X}$ – arrive and sit at one of the tables (according to some probability) which are either occupied or vacant. As a table becomes more popular, the probability that customers join it increases. The seating arrangement can be viewed to represent a DP mixture. Importantly, the probability that a *new* vacant table is chosen (over an existing table) is defined by a hyperparameter α, associated with the DP. In consequence, α is sometimes referred to as the *dispersion value* – high values lead to an increased probability that new tables (clusters) are formed, while low values lead to less tables, as new tables are less likely to be initiated.

The analogy should highlight a useful property of DP mixtures: the number of clusters K (i.e. tables) does not need to be defined in advance, instead, this is be determined by the model and the data (as well as α) (Vlachos et al. 2009). As a result, the algorithm can be particularly useful when clustering SHM signals online, as the model can adapt and update, selecting the most appropriate value for K as new information becomes available.

To demonstrate, consider a mixture of Gaussian base-distributions; a conventional *finite mixture* (a GMM) requires the number of components K to be defined *a priori*, as in the supervised Gaussian Mixture Model (GMM) with $K = 3$, shown in Figures 2 and 4. As suggested by the analogy, a DP can be interpreted as an *infinite* mixture, such that $K \to \infty$ (Rasmussen 2000); this allows for the probabilistic inference of K through the DP prior. An example DP-GMM for the same AE data (Rippengill et al. 2003) is shown in Figure 5a; the most likely number of components has been automatically found, $K = 3$, given the data and the model for $\alpha = 0.1$. The effect of the *dispersion* hyperparameter α can be visualised in Figure 5b, which shows the posterior-predictive-likelihood of K given the data for various values of α. Considering that $K = 3$, an appropriate hyperparameter range appears to be $0.01 \leq \alpha \leq 0.1$; although, as each class is clearly non-Gaussian, higher values of K are arguably more appropriate to approximate the underlying density of the data. Interestingly, for low values of α, three components appear significantly more likely to describe the data than two (or one).

For SHM in practice, the implementation of the DP for online clustering means that an operator does not need to specify an expected number of normal, environmental or damage conditions (components K) in order to build the model, which can be difficult or impossible to define for a structure in operation (Rogers

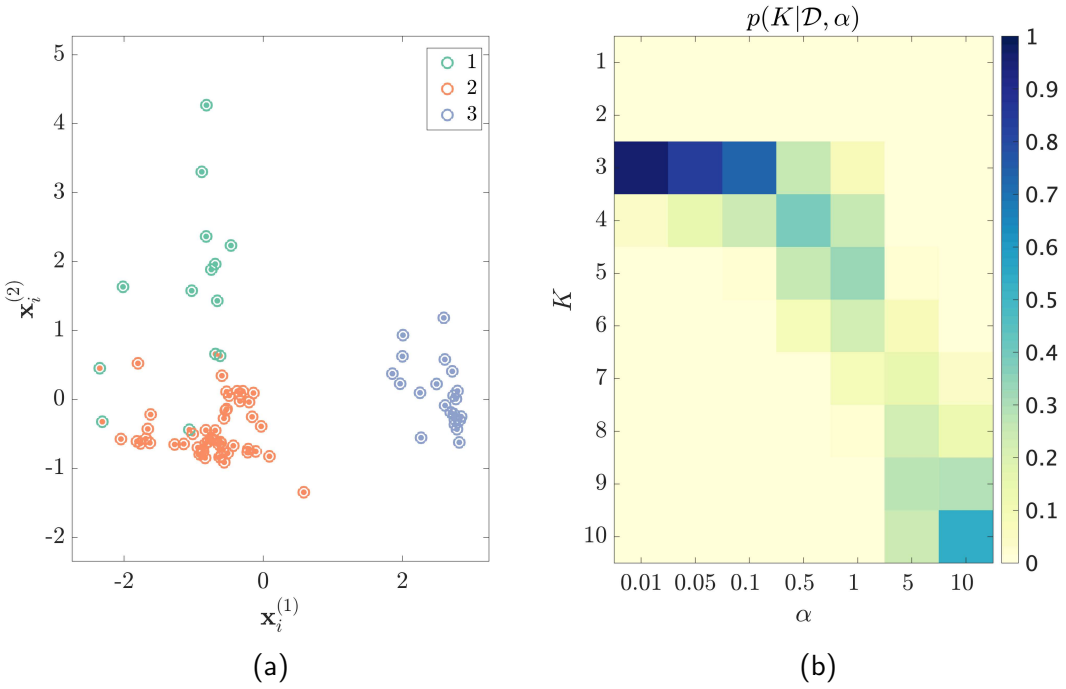


FIG. 5: Unsupervised Dirichlet process Gaussian mixture model for the three-class AE data: (a) unsupervised DP clustering, •/○ markers are the ground-truth/predicted values for y_i. (b) predictive likelihood for the number of clusters K given α, i.e. $p(K|\mathcal{D},\alpha)$.

et al. 2019).

Transfer and Multi-task Learning

Finally, methods for *transfer* (Gao and Mosalam 2018; Gardner et al. 2020d; Jang et al. 2019) and *multi-task* (Wan and Ni 2019; Huang et al. 2019) learning are proposed for inference with incomplete or limited training-data. In general terms, the idea for SHM applications is that valuable information might be transferred or shared, in some sense, between similar systems (via measured and/or simulated data). By considering *shared* information, the performance of predictive models might improve, despite insufficient training observations (Chakraborty et al. 2011; Ye et al. 2017; Dorafshan et al. 2018). For example, consider wind turbines in an offshore wind-farm; one system may have comprehensively labelled measurements, investigated by the engineer, corresponding to a range of environmental effects; other turbines within the farm are likely to experience similar effects, however, the measured signals might be incomplete, with partial labelling or no labels at all.

Various tools (Pan and Yang 2009) offer frameworks to transfer *different aspects* of shared information. For the methods discussed here, it is useful to define two objects (Gardner et al. 2020d):

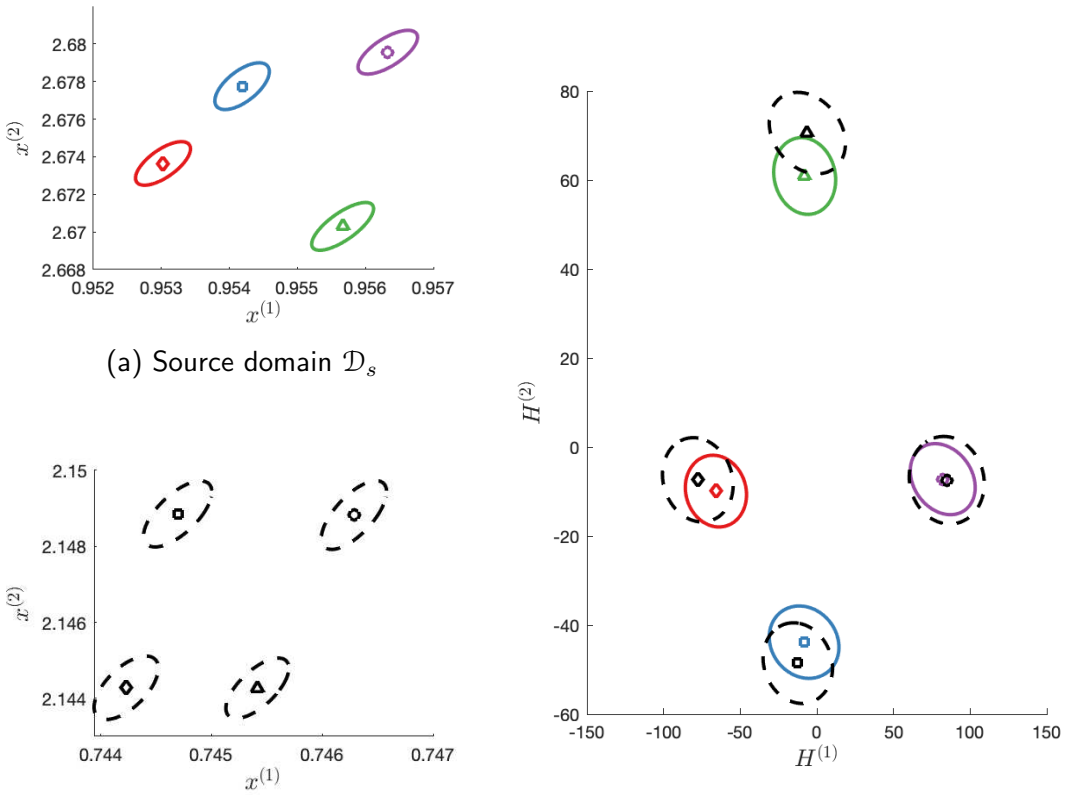
- A **Domain** $\mathcal{D} = \{\mathcal{X}, p(\mathbf{x}_i)\}$ is an object that consists of a feature space $\mathcal{X}$ and a marginal probability distribution $p(\mathbf{x}_i)$ over a finite sample of feature data $\{\mathbf{x}_i\}_{i=1}^n \in \mathcal{X}$.

- A **Task** $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ is a combination of a label space $\mathcal{Y}$ and a predictive model/ function $f(\cdot)$.

Domain adaptation is one approach to transfer learning, following a framework which maps the distributions from feature/label spaces (i.e. $\mathcal{X}/\mathcal{Y}$) associated with *different* structures into a shared (more *consistent*) space. The observations are typically *labelled* for one structure only, therefore, a predictive model $f(\cdot)$ can be learnt, such that label information is *transferred* between domains. The domain with labelled data is referred to as the *source* domain $\mathcal{D}_s$ – shown in Figure 6a – while the unlabelled data correspond to the *target* domain $\mathcal{D}_t$ – shown in Figure 6b. Importantly, a classifier $f(\cdot)$ applied in the projected latent space of Figure 6c should generalise to the target structure, despite missing label information.

Multi-task learning considers shared information from an alternative perspective. As with domain adaptation, knowledge from *multiple domains* is used to improve tasks (Pan and Yang 2009); however, in this case, each domain is weighted equally (Zhang and Yang 2018). The goal is, therefore, to generate an improved predictive function $f(\cdot)$ across multiple tasks by utilising *labelled* feature data from several different *source domains*. This approach to inference is particularly useful when labelled training-data are insufficient across multiple tasks or systems. By considering the shared knowledge across various labelled domains, the amount of the training-data can, in effect, be increased.

This work suggests *kernelised Bayesian transfer learning* (KBTL) (Gönen and Margolin 2014) to model shared information. KBTL is a particular form of multi-task learning, which can be viewed as a method for *heterogeneous* transfer; i.e. at least one feature space $\mathcal{X}_j$ for a domain $\mathcal{D}_j$ is not the same dimension as another feature space $\mathcal{X}_k$ (in the set of domains), such that $d_j \neq d_k$ (Gardner et al. 2020d). KBTL is a probabilistic method that performs two tasks: 1) finding a



(a) Source domain $\mathcal{D}_s$

(b) Target domain $\mathcal{D}_t$

(c) Latent space

FIG. 6: Visualisation of knowledge transfer via domain adaptation. Ellipses represent clusters of data – coloured according to labels. (a) and (b) are the source and target domains respectively, in their original sample spaces. (c) shows the source and target data mapped into a shared, more consistent latent space.

shared latent subspace for each domain and 2) inferring a discriminative classifier in the shared latent subspace in a Bayesian manner. It is assumed that there is a relationship between the feature space and the label space for each domain, and that all domains provide knowledge that will improve the predictive function $f(\cdot)$ for all domains (Gardner et al. 2020d).

In practice, methods such as KBTL should be particularly useful for SHM, as the (labelled) training-data are often insufficient or incomplete across structures. If, through multi-task/transfer learning, tasks from *different* structures can be considered together, this should increase the amount of information available to train algorithms. In turn, this should increase the performance of predictive models, utilising the *shared* information between systems.

DIRECTED GRAPHICAL MODELS

It will be useful to introduce basic concepts behind *directed graphical models* (DGMs), as these will be used to (visually) introduce each probabilistic algorithm. The terminology here follows that of (Murphy 2012). Generally speaking, DGMs can be used to represent the joint distribution of the variables in a statistical model by making assumptions of *conditional independence*. For these ideas to make sense, the *chain rule* is needed; that is, the joint distribution of a probabilistic model can be represented as follows, using any ordering of the variables $\{X_1, X_2 \ldots, X_V\}$:

$$\begin{aligned} p(X_{1:V}) &= p(X_1)p(X_2 \,|\, X_1)p(X_3 \,|\, X_1, X_2) \ldots p(X_V \,|\, X_{1:V-1}) \\ X_{1:V} &\triangleq \{X_1, X_2 \ldots, X_V\} \end{aligned} \tag{5}$$

In practice, a problem with expression (5) is that it becomes difficult to represent the conditional distribution $p(X_V \,|\, X_{1:V-1})$ as V gets large. Therefore, to efficiently approximate large joint distributions, assumptions of conditional independence (6) are critical. Specifically, conditional independence is denoted with $\perp$, and it implies that,

$$A \perp B \,|\, C \iff p(A, B \,|\, C) = p(A \,|\, C)\, p(B \,|\, C) \tag{6}$$

Considering these ideas, nodes in a graphical model can be used to represent variables, while edges represent conditional dependencies. For example, for the AE data (in Figures 2, 4, or 5a), one can consider a random vector $\mathbf{x}_i$ to describe the (two-dimensional) measured features $\mathbf{x}_i = \left\{x_i^{(1)}, x_i^{(2)}\right\}$, and a random variable y_i to represent the class label $\{1, 2, 3\}$. As a result, the joint distribution of an appropriate model might be $p\left(\mathbf{x}_i, y_i\right)$. To simply matters, the features can be considered to be independent (an invalid but often acceptable assumption), i.e. $x_i^{(1)} \perp x_i^{(2)} \,|\, y_i$. This leads to the following approximation of distribution of the model (for a single observation):

$$p\left(\mathbf{x}_i, y_i\right) = p\left(x_i^{(1)} \mid y_i\right) p\left(x_i^{(2)} \mid y_i\right) p\left(y_i\right) \tag{7}$$

An appropriate distribution function $p(\cdot)$ can now be assigned to each of these densities (or masses). The DGM resulting from (7) is plotted in in Figure 7a. In many cases, the features in $\mathbf{x}_i$ are the *observed* variables (measured), while the labels y_i are the *latent* (or hidden) variables that one wishes to infer. To visualise this, the observed and latent variables are shown by shaded/unshaded nodes respectively in Figure 7a. For high-dimensional feature vectors (e.g. $d >> 2$), plates can be used to represent conditionally-independent variables and avoid a cluttered graph, as shown in Figure 7b. Another plate with $i = \{1, \ldots, n\}$ is included to represent *independent and identically distributed* data, with n observations. The DGM now represents the whole dataset, which is a matrix of observed variables $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, and the vector of labels, denoted $\mathbf{y} = \{y_1, \ldots, y_n\}$. This assumptions implies that each sample was drawn independently from the same underlying distribution, such that the order in which data arrive makes no difference to the belief in the model, i.e. the likelihood of the dataset is,

$$p\left(\mathbf{X}, \mathbf{y}\right) = \prod_{i=1}^{n} p\left(x_i^{(1)} \mid y_i\right) p\left(x_i^{(2)} \mid y_i\right) p\left(y_i\right) \tag{8}$$

The corresponding DGM can be used to describe a (maximum likelihood) Naïve Bayes classifier – a simplified version of the generative classifiers applied later in this work.

CASE STUDIES

Semi-supervised, active, and multi-task learning, as well as DP clustering, are now demonstrated in case studies. A brief overview of the theory for each algorithm is provided, with the corresponding DGMs; for details behind each algorithm, the reader is referred to the SHM application papers (Bull et al. 2019b; Bull et al. 2020b; Rogers et al. 2019; Gardner et al. 2020d; Gardner et al. 2020a).

Active learning with Gaussian Mixture Models

A generative classifier is used to demonstrate probabilistic active learning. In this example – originally shown in (Bull et al. 2020b) – a Gaussian mixture model (GMM) is used to monitor streaming data from a motorway bridge, as if the signals were recorded online. The model defines a multi-class classifier, to aid both damage detection and identification, while limiting the number of (costly) system inspections.

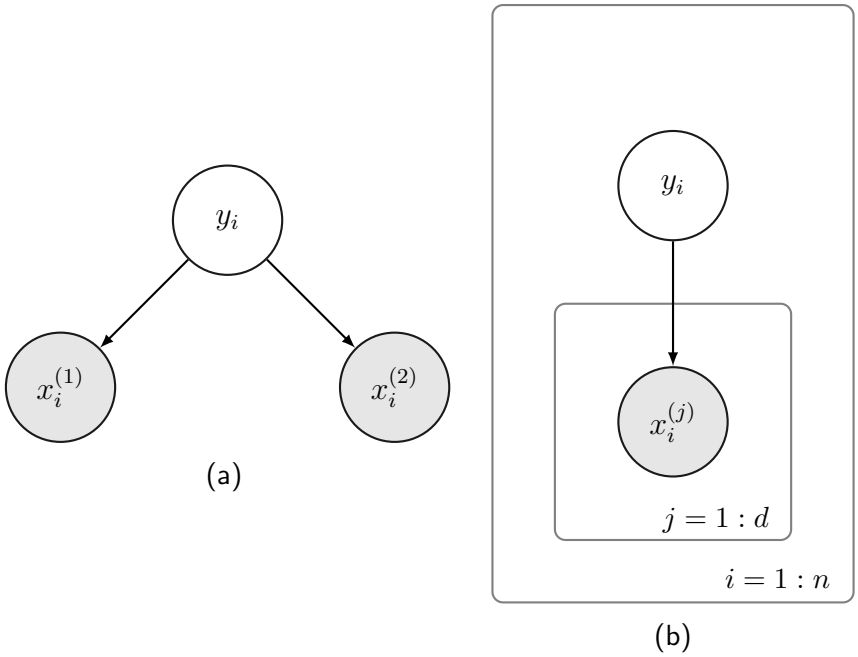


FIG. 7: Examples of directed graphical models (DGMs) based on the AE data. Shaded and unshaded nodes represent observed/latent variables respectively; arrows represent conditional dependencies; boxes represent plates.

The directed graphical model

As the data are being approximated by a Gaussian mixture model, when a new class k is discovered from the streaming data (following inspection), it is assigned a Gaussian distribution – Gaussian clusters like this can be visualised for the AE data in Figure 2. Note: the first DGM is explained in detail, to introduce the theory that is used throughout. The conditional distribution of the observations $\mathbf{x}_i$ given label $y_i = k$ is, therefore,

$$p\left(\mathbf{x}_i \mid y_i = k\right) = \mathcal{N}\left(\mathbf{x}_i\,;\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{9}$$

(Semicolon notation ; is used to indicate that a function is parameterised by the variables that follow – this is distinct from bar notation | which implies a conditional probability.) k is used to index the class group, given the number of observed clusters at that time $k \in \{1, ..., K\}$. As such, $\boldsymbol{\mu}_k$ is the mean (centre) and $\boldsymbol{\Sigma}_k$ is the covariance (scatter) of the cluster of data $\mathbf{x}_i$ with label k, for K Gaussian base-distributions.

A discrete random variable is used to represent the labels y_i, which is cate-

gorically distributed, parameterised by a vector of *mixing proportions* $\boldsymbol{\lambda}$,

$$p\left(y_i\right) = \mathrm{Cat}(y_i\,;\,\boldsymbol{\lambda}) \tag{10}$$

the mixing proportions can be viewed as a histogram over the label values, such that $\boldsymbol{\lambda} = \{\lambda_1, ..., \lambda_K\}$ and $p(y_i = k) = P\left(y_i = k\right) = \lambda_k$.

The collected parameters of the model (from each component) are denoted by $\boldsymbol{\theta}$, such that $\boldsymbol{\theta} = \{\boldsymbol{\Sigma}, \boldsymbol{\mu}, \boldsymbol{\lambda}\} = \{\boldsymbol{\Sigma}_i, \boldsymbol{\mu}_i, \boldsymbol{\lambda}_i\}_{i=1}^K$; therefore, the joint distribution of the model could be written as,

$$p\left(\mathbf{x}_i, y_i\,;\,\boldsymbol{\theta}\right) = p\left(\mathbf{x}_i \,|\, y_i\,;\,\boldsymbol{\theta}\right) p(y_i\,;\,\boldsymbol{\theta}) \tag{11}$$

However, to consider a more *complete* model, a Bayesian approach is adopted. That is, the parameters $\boldsymbol{\theta}$ themselves are considered to be random variables, and, therefore, they are included in the joint distribution (rather than simply parametersing it),

$$\begin{aligned} p\left(\mathbf{x}_i, y_i, \boldsymbol{\theta}\right) &= p\left(\mathbf{x}_i \,|\, y_i, \boldsymbol{\theta}\right) p(y_i \,|\, \boldsymbol{\theta}) p\left(\boldsymbol{\theta}\right) && (12)\\ &= p\left(\mathbf{x}_i \,|\, y_i, \boldsymbol{\Sigma}, \boldsymbol{\mu}\right) p\left(\boldsymbol{\Sigma}, \boldsymbol{\mu}\right) p(y_i \,|\, \boldsymbol{\lambda}) p\left(\boldsymbol{\lambda}\right) && (13) \end{aligned}$$

This perspective has various advantages; importantly, it allows for the incorporation of prior knowledge regarding the parameters via the *prior distribution* $p\left(\boldsymbol{\theta}\right)$. Additionally, when implemented correctly, Bayesian methods lead to robust, self-regularising models (Rasmussen and Ghahramani 2001).

To provide analytical solutions, it is convenient to assign conjugate (prior) distributions over the parameters $p\left(\boldsymbol{\theta}\right) = p\left(\boldsymbol{\Sigma}, \boldsymbol{\mu}\right) p(\boldsymbol{\lambda})$. Here it is assumed that $\{\boldsymbol{\Sigma}, \boldsymbol{\mu}\}$ are independent from $\boldsymbol{\lambda}$, to define two conjugate pairs; one associated with the observations $\mathbf{x}_i$ and another with the labels y_i. For the mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$, a conjugate (hierarchical) prior is the Normal Inverse Wishart (NIW) distribution,

$$p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \mathrm{NIW}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\,;\,\mathbf{m}_0, \kappa_0, \nu_0, \mathbf{S}_0) \tag{14}$$

This introduces the *hyperparameters* $\{\mathbf{m}_0, \kappa_0, \nu_0, \mathbf{S}_0\}$ associated with the prior, which can be interpreted as follows: $\mathbf{m}_0$ is the prior mean for the location of each

class $\boldsymbol{\mu}_k$, and κ_0 determines the strength of the prior; $\mathbf{S}_0$ is (proportional to) the prior mean of the covariance, $\boldsymbol{\Sigma}_k$, and ν_0 determines the strength of that prior (Murphy 2012). Considering that the streaming data will be normalised (online), it is reasonable that hyperparemeters are defined such that the prior belief states that each class is represented by a zero-mean and unit-variance Gaussian distribution.

For the mixing proportions, the conjugate prior is a Dirichlet (Dir) distribution, parameterised by $\boldsymbol{\alpha}$, which encodes the prior belief of the mixing proportion (or weight) of each class. In this case, each class is assumed equally weighted *a priori* for generality – although, care should be taken when setting this prior, as it is application specific, particularly for streaming data (Bull et al. 2019b).

$$p(\boldsymbol{\lambda}) = \mathrm{Dir}(\boldsymbol{\lambda}\,;\,\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} {\lambda_k}^{\alpha_k - 1} \tag{15}$$

$$\boldsymbol{\alpha} \triangleq \{\alpha_1, \ldots, \alpha_k\} \tag{16}$$

With this information, the joint distribution of the model $p(\mathbf{x}_i, y_i, \boldsymbol{\theta})$ can be approximated, such that $p(\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathbf{x}_i, y_i, \boldsymbol{\theta})$. The associated DGM can be drawn, including conditional dependences and hyperparameters, for n (supervised) training data in Figure 8.

Having observed the labelled training data $\mathcal{D}_l = \{\mathbf{X}, \mathbf{y}\}$, the posterior distributions can be defined by applying Bayes' theorem to each conjugate pair – where $\mathbf{X}_k$ denotes the observations $\mathbf{x}_i \in \mathbf{X}$ with the labels $y_i = k$,

$$p\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \mid \mathbf{X}_k, \right) = \frac{p\left(\mathbf{X}_k \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) p\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{p(\mathbf{X}_k)} \tag{17}$$

$$p\left(\boldsymbol{\lambda} \mid \mathbf{y}\right) = \frac{p(\mathbf{y} \mid \boldsymbol{\lambda}) p\left(\boldsymbol{\lambda}\right)}{p(\mathbf{y})} \tag{18}$$

In general terms, while the prior $p(\boldsymbol{\theta})$ was the distribution over the parameters *before* any data were observed, the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D}_l)$ describes the parameters given the training data (i.e. conditioned on the training data). Conveniently, each of these have analytical solutions (Barber 2012; Murphy 2012).

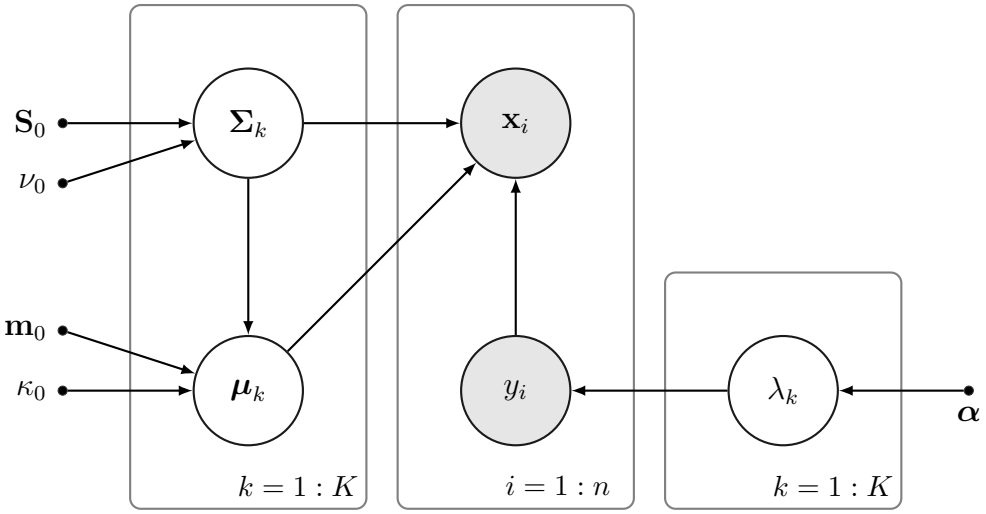


FIG. 8: Directed graphical model for the GMM $p(\mathbf{x}_i, y_i, \boldsymbol{\theta})$ over the *labelled* data $\mathcal{D}_l$. As training data are supervised, both $\mathbf{x}_i$ and y_i are observed variables. Shaded and white nodes are the observed and latent variables respectively; arrows represent conditional dependencies; dots represent constants (i.e. hyperparameters). Adapted from (Bull 2019).

Active sampling

To use the DGM to query informative data recorded from the motorway bridge, an initial model is learnt given a small sample of data recorded at the beginning of the monitoring regime. In this case, it should be safe to assume the labels $y_i = 1$, which corresponds to the normal condition of the structure. As new (unlabelled) measurements arrive online, denoted $\tilde{\mathbf{x}}_i$, the model can be used to predict the labels *under uncertainty*. The predictive equations are found by marginalising (integrating) out the parameters from the joint distribution (for each conjugate pair),

$$p(\tilde{\mathbf{x}}_i \,|\, \tilde{y}_i = k, \mathcal{D}_l) = \int\int p(\tilde{\mathbf{x}}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \underbrace{p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \,|\, \mathcal{D}_l)}_{\text{Eq.(17)}} \, d\boldsymbol{\mu}_k d\boldsymbol{\Sigma}_k \tag{19}$$

$$p(\tilde{y}_i \,|\, \mathcal{D}_l) = \int p(\tilde{y}_i \,|\, \boldsymbol{\lambda}) \underbrace{p(\boldsymbol{\lambda} \,|\, \mathcal{D}_l)}_{\text{Eq.(18)}} \, d\boldsymbol{\lambda} \tag{20}$$

Again, due to conjugacy, these have analytical solutions (Murphy 2012). The posterior predictive equations (19) and (20) can be combined to define the pos-

terior over the label estimates given unlabelled observations of the bridge,

$$p(\tilde{y}_i \,|\, \tilde{\mathbf{x}}_i, \mathcal{D}_l) = \frac{p(\tilde{\mathbf{x}}_i \,|\, \tilde{y}_i, \mathcal{D}_l)\; p(\tilde{y}_i \,|\, \mathcal{D}_l)}{p(\tilde{\mathbf{x}}_i \,|\, \mathcal{D}_l)} \tag{21}$$

Considering the predictive distribution (21), labels that appear most uncertain can be investigated by the engineer. This observation is now labelled $\{\mathbf{x}_i, y_i\}$, thus extending the (supervised) training set $\mathcal{D}_l$. Two measures of uncertainty are considered: a) the marginal likelihood of the new observation given the model (the denominator of Equation (21)) and b) the entropy of the predicted label, given by,

$$H(\tilde{y}_i) = -\sum_{k=1}^{K} p(\tilde{y}_i = k \,|\, \tilde{\mathbf{x}}_i, \mathcal{D}_l) \log p(\tilde{y}_i = k \,|\, \tilde{\mathbf{x}}_i, \mathcal{D}_l) \tag{22}$$

Queries with high entropy consider data at the boundary between two existing classes, while queries given low likelihood will select data that appear unlikely given the current model estimate. Visual examples of data that would be selected given these measures are shown in Figure 4a for high entropy, and Figure 4b for low likelihood.

Figure 9 demonstrates how streaming SHM signals might be queried using these uncertainty measures. The (unlabelled) data arrive online, in batches of size B; the data that appear most uncertain (given the current model) are investigated. The number of investigations per batch q_b is determined by the label budget, which, in turn, is limited by cost implications. Once labelled by the engineer, these data can be added to $\mathcal{D}_l$ and used to update the classification model.

Z24 bridge dataset

The Z24 bridge was a concrete highway bridge in Switzerland, connecting the villages of Koppigen and Utzenstorf. Before its demolition in 1998, the bridge was used for experimental SHM purposes (de Roeck 2003). Over a twelve-month period, a series of sensors were used to capture dynamic response measurements, to extract the first four natural frequencies of the structure. Air/deck temperature, humidity and wind speed were also recorded (Peeters and de Roeck 2001). There are a total of 3932 observations in the dataset.

Before demolition, different types of damage were artificially introduced, starting from observation 3476 (Dervilis et al. 2014). The natural frequencies and deck

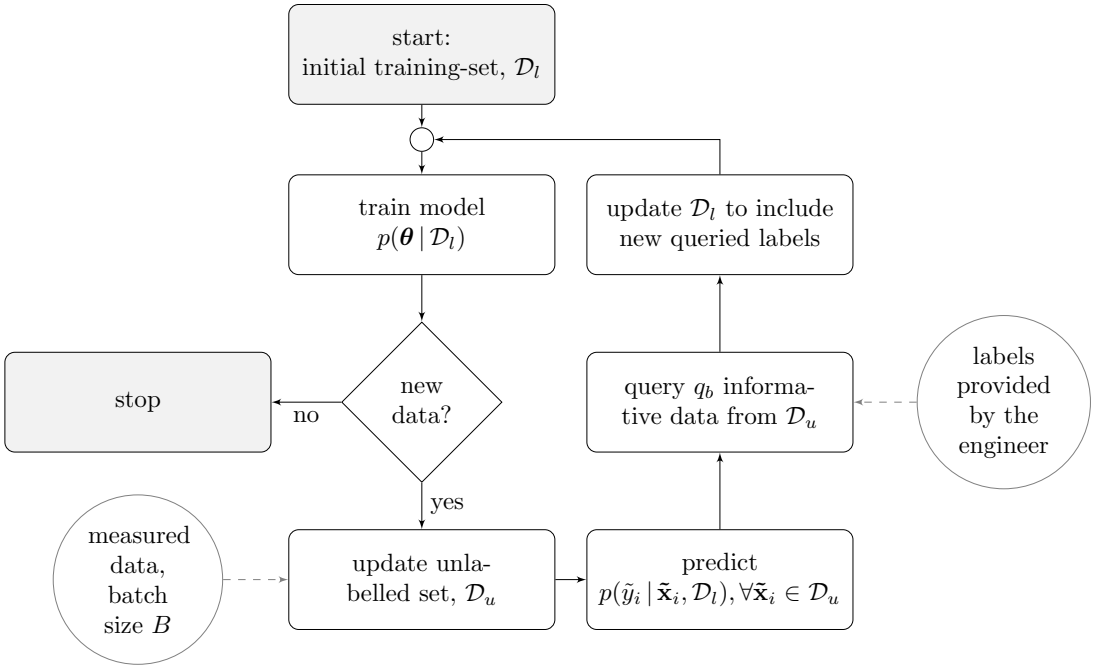


FIG. 9: Flow chart to illustrate the online active learning process – adapted from (Bull et al. 2019b).

temperature are shown in Figure 10. Visible fluctuations in the natural frequencies can be observed in Figure 10, for $1200 \leq n \leq 1500$, while there is little variation following the introduction of damage at observation 3476. It is believed that the asphalt layer in the deck experienced very low temperatures during this time, leading to increased structural stiffness.

In the analysis, the four natural frequencies are the observation data, such that $\mathbf{x}_i \in \mathbb{R}^4$. The damage data are assumed to represent their own class, from observation 3476. Outlying observations within the remaining dataset are determined using the robust Minimum Covariance Determinant (MCD) algorithm (Rousseeuw and Driessen 1999; Dervilis et al. 2014). In consequence, a three-class classification problem is defined, according to the colours in Figure 10: normal data (blue), outlying data due to environmental effects (green), and damage (red), corresponding to $y_i \in \{1, 2, 3\}$ respectively.

Clearly, it is undesirable for an engineer to investigate the bridge following each data acquisition. Therefore, if active learning can provide an improved classification performance, compared to passive learning (random sampling) with the same sample budget, this demonstrates the relevance of active methods to SHM.

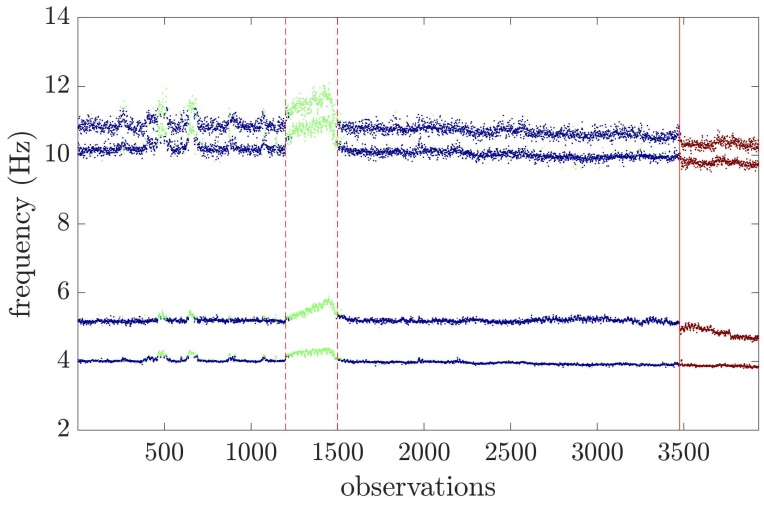


FIG. 10: Z24 bridge data, time history of natural frequencies, colours represent three classes of data: normal data (blue), outlying data due to environmental effects (green), and damage (red).

Results: Active learning

The model is applied *online* to the frequency data from the Z24 bridge. To provide a online performance metric, the dataset is divided into two equal subsets: one is used for training and querying by the active learner $\{\mathcal{D}_l, \mathcal{D}_u\}$, the other is used as a distinct/independent test set. The f_1 score is used as the performance metric (throughout this work); this is a weighted average of precision and recall (Murphy 2012), with values between 0 and 1; a perfect score corresponds to $f_1 = 1$. Precision (P) and recall (R) can be defined in terms of numbers of true positives (TP), false positives (FP) and false negatives (FN) for each class, $k \in Y$ (Murphy 2012),

$$P_k = \frac{TP_k}{TP_k + FP_k} \tag{23a}$$

$$R_k = \frac{TP_k}{TP_k + FN_k} \tag{23b}$$

The (macro) f_1 score is then defined by (Murphy 2012),

$$f_{1,k} = \frac{2P_k R_k}{P_k + R_k} \tag{24a}$$

$$f_1 = \frac{1}{K} \sum_{k \in Y} f_{1,k} \tag{24b}$$

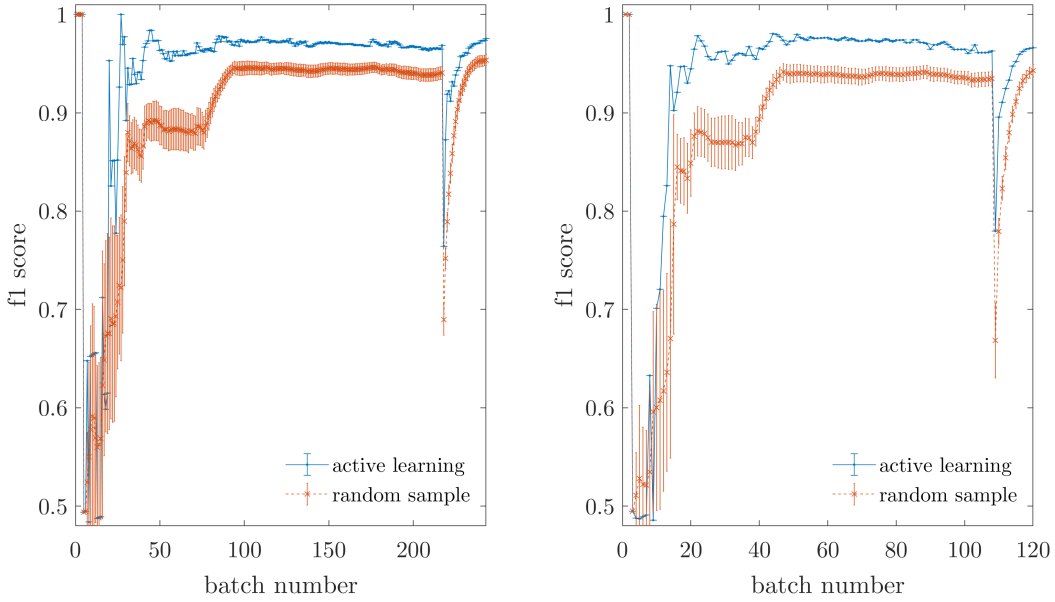


FIG. 11: Online classification performance (f_1 score) for the Z24 data, for query budgets of (a) 25%; (b) 12.5% of the total dataset – adapted from (Bull et al. 2019b).

Figure 11 illustrates improvements in classification performance when active learning is used to label 25% and 12.4% of the measured data. Active learning is compared to the *passive* learning benchmark, where the same number of data are labelled according to a random sample, rather than uncertainty measures. Throughout the monitoring regime, if the GMM is used to select the training data, the predictive performance increases. Most notably, drops in the f_1 score (corresponding to new classes being discovered) are less significant when active learning is used to select data; particularly when class two (environmental effects) is introduced. This is because new classes are *unlikely* given the current model, i.e. uncertainty measure (a). Intuitively, novel classes are discovered sooner via uncertainty sampling. For a range of query budgets and additional SHM applications refer to (Bull et al. 2019b). Code and animations of uncertainty sampling for the Z24 data are available at `https://github.com/labull/probabilistic_active_learning_GMM`.

Semi-supervised updates to Gaussian Mixture Models

While active learning considered the unlabelled data $\mathcal{D}_u$ for querying, the observations only contribute to the model once labelled; in other words, once included in the labelled set $\mathcal{D}_l$. A semi-supervised model, however, can consider both the labelled *and* unlabelled data when approximating the parameters. Therefore, $\boldsymbol{\theta}$ is estimated given *both* labelled and unlabelled observations, such that the posterior becomes $p(\boldsymbol{\theta} \mid \mathcal{D}_l, \mathcal{D}_u)$. This is advantageous for SHM, *unlabelled* observations can also contribute to the model estimate; reducing the dependance on costly supervised data.

Continuing the probabilistic approach, the original DGM in Figure 8 can be updated (relatively simply) to become semi-supervised – shown in Figure 12. The inclusion of $\mathcal{D}_u$ introduces another latent variable $\tilde{y}_i$, and, as a result, obtaining the posterior distribution over the parameters becomes less simple. One solution adopts an expectation maximisation (EM) approach (Dempster et al. 1977). The implementation here involves finding the maximum *a posteriori* (MAP) estimate of the parameters $\hat{\boldsymbol{\theta}}$ (the mode of the full posterior distribution), while maximising the likelihood of the model. Specifically, from the joint distribution, and using Bayes' theorem, the MAP estimate of the parameters $\boldsymbol{\theta}$ given the labelled and unlabelled subsets is,

$$\begin{aligned} \hat{\boldsymbol{\theta}} \mid \mathcal{D} &= \operatorname{argmax}_{\boldsymbol{\theta}} \left\{ \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})} \right\} \\ &= \operatorname{argmax}_{\boldsymbol{\theta}} \left\{ \frac{p(\mathcal{D}_u \mid \boldsymbol{\theta}) p(\mathcal{D}_l \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D}_u, \mathcal{D}_l)} \right\} \\ \mathcal{D} &\triangleq \mathcal{D}_u \cup \mathcal{D}_l \end{aligned} \tag{25}$$

Again, it is assumed that the data are i.i.d, so that $\mathcal{D}_l$ and $\mathcal{D}_u$ can be factorised. Thus, the marginal likelihood of the model (the denominator of equation (25)), considers both the labelled and unlabelled data – this is referred to as the *joint likelihood*, and it is the value that is maximised while inferring the parameters of the model through EM.

The EM algorithm iterates E and M steps until convergence in the joint (log) likelihood. During each E-step, the parameters are fixed, and the unlabelled observations are classified using the current model estimate $p\left(\tilde{\mathbf{y}} \mid \tilde{\mathbf{X}}, \mathcal{D}\right)$. The M-step corresponds to finding the $\hat{\boldsymbol{\theta}}$, given the predicted labels from the E step

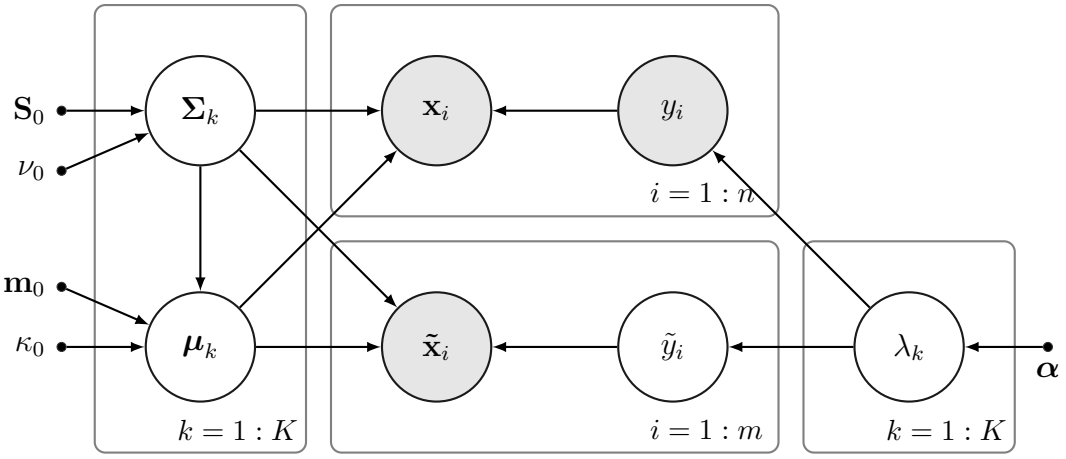


FIG. 12: DGM of the semisupervised GMM, given the labelled $\mathcal{D}_l$ and unlabelled data $\mathcal{D}_u$. For the unsupervised set, $\tilde{\mathbf{x}}_i$ is the only observed variable, while $\tilde{y}_i$ is a latent variable. Adapted from (Bull 2019).

Algorithm 1: *Semi-supervised EM for a Gaussian Mixture Model*

Input : Labelled data $\mathcal{D}_l$, unlabelled data $\mathcal{D}_u$
Output: Semi-supervised MAP estimates of $\hat{\boldsymbol{\theta}} = \left\{\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\right\}$

Initilise $\hat{\boldsymbol{\theta}}$ using the labelled data, $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} \{p(\boldsymbol{\theta} \,|\, \mathcal{D}_l)\}$;
while *the joint log-likelihood* $\log \{p(\mathcal{D}_l, \mathcal{D}_u)\}$ *improves* **do**
    E-step: use the current model $\hat{\boldsymbol{\theta}} \mid \mathcal{D}$ to estimate class-membership for the unlabelled data $\mathcal{D}_u$, i.e. $p\left(\tilde{\mathbf{y}} \mid \tilde{\mathbf{X}}, \mathcal{D}\right)$;
    M-step: update the MAP estimate of $\hat{\boldsymbol{\theta}}$ given the component membership for *all* observations $\hat{\boldsymbol{\theta}} := \text{argmax}_{\boldsymbol{\theta}} \{p(\boldsymbol{\theta} \,|\, \mathcal{D}_l, \mathcal{D}_u)\}$;
end

and the absolute labels for the supervised data. This involves some minor modifications to the conventional MAP estimates, such that the contribution of the unlabelled data is shared between classes, weighted according to the posterior distribution $p\left(\tilde{\mathbf{y}} \mid \tilde{\mathbf{X}}, \mathcal{D}\right)$ (Barber 2012; Bull et al. 2020b). Pseudo-code is provided in Algorithm 1; Matlab code for the semi-supervised GMM is also available at `https://github.com/labull/semi_supervised_GMM`.

Semi-supervised learning with the Gnat aircraft data

A visual example of improvements to a GMM via semi-supervision was shown in Figure 2. To quantify potential advantages for SHM, the method is also applied to experimental data from aircraft experiments, originally presented in (Bull

et al. 2020b). For details behind the Gnat aircraft data, refer to (Manson et al. 2003). Briefly, during the tests, the aircraft was excited with an electrodynamic shaker and band-limited white noise. Transmissibilty data were recorded using a network of sensors distributed over the wing. Artificial damage was introduced by sequentially removing one of nine inspection panels in the wing. 198 measurements were recorded for the removal of each panel, such that the total number of (frequency domain) observations is 1782. Over the network of sensors, nine transmissibilties were recorded (Manson et al. 2003). Each transmissibility was converted to a one-dimensional novelty detector, with reference a distinct set of normal data, where all the panels were intact (Worden et al. 2008). Therefore, the data represent a nine-class classification problem, one class for the removal of each panel, such that $y_i = \{1, \dots, 9\}$. The measurements are nine-dimensional $\mathbf{x}_i \in \mathbb{R}^9$, each feature is a novelty index, representing one of nine transmissibilities.

When applying semi-supervised learning, 1/3 of the total data were set aside as an independent test-set. The remaining 2/3 were used for training, i.e. $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$. Of the training data $\mathcal{D}$, the number of labelled observations n was increased (in 5% increments) until all the observations are labelled. The results are compared to standard supervised learning for the same budget n.

The changes in the classification performance through semi-supervised updates are shown in Figure 13; inclusion of the unlabelled data consistently improves the f_1 score. For very low proportions of labelled data $< 1.26\%$ ($m \gg n$), semi-supervised updates can decrease the predictive performance, this is likely due to the unlabelled data outweighing the labelled instances in the likelihood cost function. Notably, the maximum increase in the f_1 score is 0.0405, corresponding to a 3.83% reduction in the classification error for 2.94% labelled data. Such improvements to the classification performance for for low proportions of labelled data should highlight significant advantages for SHM, reducing the dependence on large sets of costly supervised data.

Dirichlet Process Clustering of Streaming Data

Returning to the streaming data recorded from the Z24 bridge, an alternative perspective considers that labels are not needed to *infer* the model. In this case, an *unsupervised* algorithm could be used to cluster data online, and labels could be assigned to the resulting clusters *outside* of the inference, within the wider SHM scheme – as suggested by (Rogers et al. 2019). However, if y_i is unobserved for the purposes of inference, the number of class components K becomes an

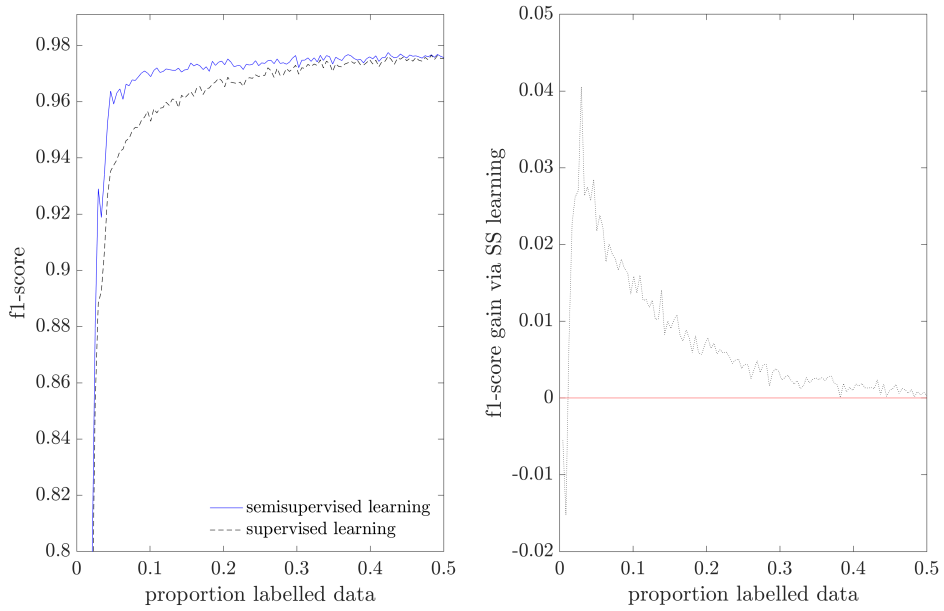


FIG. 13: Classification performance (f_1 score) for the supervised GMM vs. the semi-supervised GMM. Left: f_1 for an increasing proportion of labelled data. Right: the gain in f_1 score through semi-supervised updates, the red line highlights zero-gain. Adapted from (Bull et al. 2020b).

additional latent variable, unlike the GMM from previous case studies.

As aforementioned, the Dirichlet Process Gaussian Mixture Model (DPGMM) is one solution to this problem. The DPGMM allows for the probabilistic selection of K through the a Dirichlet process prior. Initially, this involves defining a GMM in a Bayesian manner, using the same priors as before; however, by following (Rasmussen 2000), it is possible to take the limit $K \to \infty$ to form an infinite Gaussian mixture model. Surprisingly, this concept can be shown through another simple modification to the first DGM in Figure 8, leading to Figure 14. The generative equations remain the same as (9), (10), (14), and (15).

A collapsed Gibbs sampler can be used to perform efficient online inference over this model (Neal 2000). Although potentially faster algorithms for variational inference exist (Blei et al. 2006), it can be more practical to implement the Gibbs sampler when performing inference online. The nature of the Gibbs sampling solution is that each data point is assessed conditionally in the sampler, this allows the addition of new points online, rather than batch updates (Rogers et al. 2019).

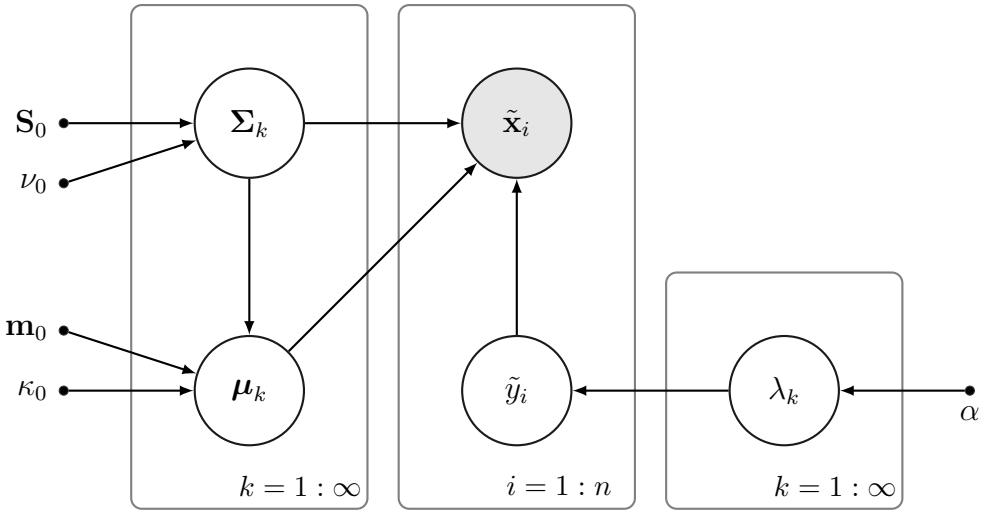


FIG. 14: DGM for the infinite Gaussian mixture model.

Within the Gibbs sampler, only components $k = \{1, \ldots, K+1\}$ need to be considered to cover the full set of possible clusters (Rasmussen 2000). As with the GMM, there are two conjugate pairs in the model; therefore, the predictive equations remain analytical (leading to a *collapsed* Gibbs sampler). In brief/general terms: while fixing the parameters, the Gibbs scheme determines the likelihood of an observation $\tilde{\mathbf{x}}_i$ being sampled from an existing cluster $k = \{1, \ldots, K\}$, or an (as of yet) unobserved cluster $k = K+1$ (i.e. the prior). Given the posterior over the $K+1$ classes, the cluster assignment $\tilde{y}_i$ is sampled, and the model parameters are updated accordingly. This process is iterated until convergence.

Applications to the Z24 bridge data

In terms of monitoring the streaming Z24 data, any new observations that relate to existing clusters will update the associated parameters. If a new cluster is formed, indicating novelty, this triggers an alarm. In this case, the cluster must contain at least 50 observations to indicate novelty; for details refer to (Rogers et al. 2019). Upon investigating the structure, an appropriate description can be assigned to the unsupervised cluster index (outside of the inference). As before, the Z24 data are normalised in an online manner, thus, the hyperparemeters of the prior $p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ encode this knowledge. The choice of the dispersion value α, defining $p(\boldsymbol{\lambda})$, is more application dependent – as discussed in the restaurant analogy, this determines the likelihood that new clusters will be generated. In (Rogers et al. 2019), sensible values for online SHM applications were found to be between $0 < \alpha < 20$; for the Z24 data, this is set to $\alpha = 10$.

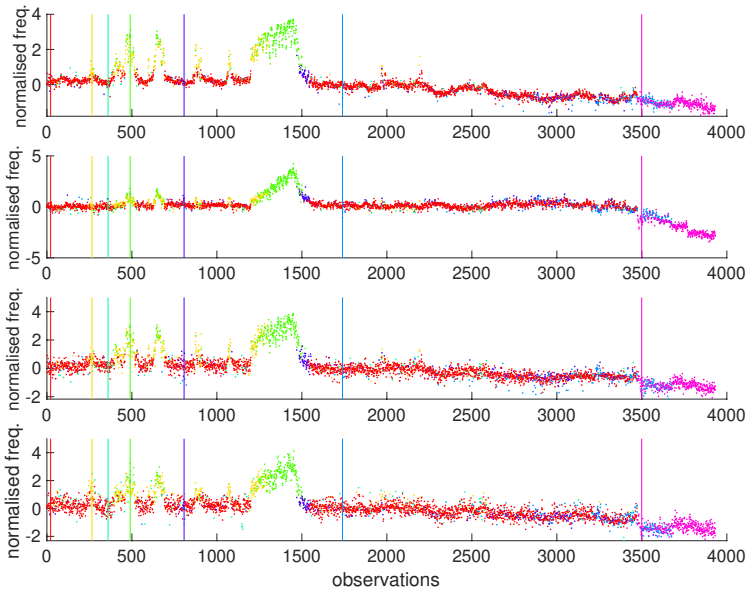


FIG. 15: Figure showing online DP clustering applied to the Z24 bridge data using the first four natural frequencies as the features. Vertical lines indicate that a new cluster has been formed. Adapted from (Rogers et al. 2019).

As with the active GMM, a small set of data from the start of the monitoring regime make up an initial training set. Figure 15 shows the algorithm progress for the streaming data. A normal condition cluster (red) is quickly established. As the temperature cools, three more cluster are created (orange, cyan and green) corresponding to the progression of freezing of the deck. Two additional clusters are also created: dark blue around point 800 and light blue close to point 1700. From inspection of the feature space (Rogers et al. 2019), it is hypothesised that the light blue cluster corresponds to a shift and rotation in the normal condition; therefore, this leads to another *normal* cluster. As the corresponding normal data are now non-Gaussian, they are better approximated by two mixture components. Finally, the magenta cluster is created following two observations of damage, showing the ability of the DPGMM implementation to detect a change in behaviour corresponding to damage, as well as environmental effects.

The DPGMM has automatically inferred seven clusters given the data and the model. While three classes were originally defined (as in the active and semi-supervised case), this representation is equally interpretable following system inspections to describe each component. Additionally, the DPGMM is likely

to better approximate the underlying density, as each class of data can be described by a number of Gaussian components, rather than one. That is, in this case: three clusters describe the normal condition (blues and red), three clusters cover various environmental effects (orange, cyan and green), and one represents the damage condition (magenta).

The results shown on the Z24 data demonstrate the ability of the online DP algorithm to deal with recurring environmental conditions while remaining sensitive to damage. The DPGMM is incorporated into an SHM system for online damage detection, and it is shown to categorise multiple damaged and undamaged states, while automatically inferring an appropriate number of mixture components K in the mixture model. The method requires little user input, and it updates online with simple feedback to the user as to when inspection is likely required. If desired, the unsupervised clusters can be assigned meaningful descriptions, to be interpreted by the end user.

Multi-task learning

In the final case study, supervised data from different structures (each represented by their own domain) are considered simultaneously to improve the performance of an SHM task. In the following example, each domain $\mathcal{D}_t$ corresponds to supervised training data recorded from a different system; the task $\mathcal{T}$ corresponds to a predictive SHM model. By considering the data from a group (or population) of *similar* structures in a latent space, the amount of training data can (in effect) be increased. Multi-task learning should be particularly useful in SHM, where training data are often incomplete for individual systems. If a predictive model can be improved by considering the data collected from various *similar* structures, this should highlight the potential benefit of multi-task learning.

Kernelised Bayesian transfer learning

Referring back to task $\mathcal{T}$ and domain $\mathcal{D}$ objects, it is assumed that there are T (binary) classification tasks over the heterogeneous domains $\{\mathcal{D}_t\}_{t=1}^{T}$. In other words, the label space $\mathcal{Y}$ is consistent across all tasks (in this case, normal or damaged), while the feature space $\mathcal{X}_t$ can change dimensionality, potentially leading to $d_t \neq d_{t'}$. For each task, there is an i.i.d. training set of observations $\mathbf{X}_t$ and labels $\mathbf{y}_t$, where $\mathbf{X}_t = \left\{\mathbf{x}_i^{(t)} \in \mathbb{R}^{d_t}\right\}_{i=1}^{n_t}$ and $\mathbf{y}_t = \left\{y_i^{(t)} \in \{-1, +1\}\right\}_{i=1}^{n_t}$. Each domain has a task specific kernel function k_t to determine the similarities between observations and the associated kernel matrix $\mathbf{K}_t[i,j] = k_t\left(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}\right)$,

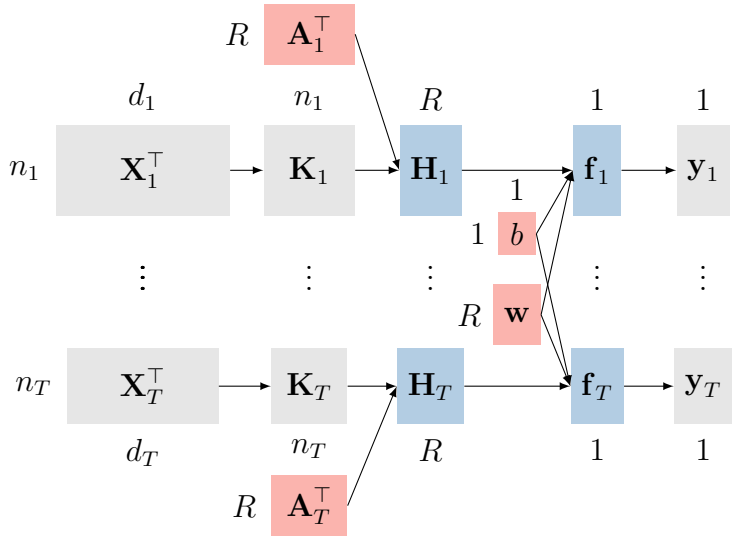


FIG. 16: Visualisation of KBTL – adapted from (Gönen and Margolin 2014).

such that $\mathbf{K}_t \in \mathbb{R}^{n_t \times n_t}$. Note: when subscripts/superscripts are cluttered, square bracket notation is used to index matrices and vectors.

Figure 16 is useful to visualise KBTL. The model can be split into two main parts: (i) the first projects data from different tasks into a shared subspace using kernel-based dimensionality reduction, (ii) the second performs *coupled* binary classification in the shared subspace, using common classification parameters. In terms of notation, the kernel embedding for each domain $\mathbf{K}_t$ is projected into a shared latent subspace by an optimal projection matrix $\mathbf{A}_t \in \mathbb{R}^{n_t \times R}$, where R is the dimensionality of the subspace. Following projection, there is a representation of each domain in the shared latent subspace, $\left\{\mathbf{H}_t = \mathbf{A}_t^\top \mathbf{K}_t\right\}_{t=1}^T$. In this shared space, a *coupled* discriminative classifier is inferred for the projected data from each domain $\left\{\mathbf{f}_t = \mathbf{H}_t^\top \mathbf{w} + \mathbf{1}b\right\}_{t=1}^T$. This implies the same set of parameters $\{\mathbf{w}, b\}$ are used across all tasks.

In a Bayesian manner, prior distributions are associated with the parameters of the model. For the $n_t \times R$ task-specific projection matrices, $\mathbf{A}_t$, there is an $n_t \times R$ matrix of priors, denoted $\mathbf{\Lambda}_t$. For the weights of the coupled classifier, the prior is $\boldsymbol{\eta}$, and for the bias b the prior is γ. These are standard priors given the parameter types in the model – for details refer to (Gönen and Margolin 2014). Collectively, the priors are $\mathbf{\Xi} = \left\{\{\boldsymbol{\lambda}_t\}_{t=1}^T, \boldsymbol{\eta}, \gamma\right\}$ and the latent variables are $\mathbf{\Theta} = \left\{\{\mathbf{H}_t, \mathbf{A}_t, \mathbf{f}_t\}_{t=1}^T, \mathbf{w}, b\right\}$; the observed variables (training data) are given by $\{\mathbf{K}_t, \mathbf{y}_t\}_{t=1}^T$.

The DGM associated with the model is shown in Figure 17; this highlights the variable dependences and the associated prior distributions. The distributional assumptions are *briefly* summarised, for details, refer to (Gönen and Margolin 2014). The prior for the elements $\mathbf{A}_t[i,s]$ of the projection matrix are (zero mean) normally distributed, with variance $\mathbf{\Lambda}_t[i,s]^{-1}$; in turn, the prior over $\mathbf{\Lambda}_t[i,s]$ is Gamma distributed. As a result, the observations are normally distributed in the latent space, i.e. $\mathbf{H}_t[s,i]$. For the coupled classifier, the prior for the bias b is assumed to be (zero mean) normally distributed, with variance γ^{-1}, such that γ is Gamma distributed. Similarly, the weights $\mathbf{w}[s]$ are (zero mean) normally distributed, with variance $\boldsymbol{\eta}[s]^{-1}$, such that $\boldsymbol{\eta}[s]$ is Gamma distributed. This leads to normal distributions over the functional classifier $\mathbf{f}_t[i]$. The label predictive equations are given by $p(y_*^{(t)} \mid f_*^{(t)})$, passing $f_*^{(t)}$ through a truncated Gaussian, parameterised by ν (Gardner et al. 2020c).

The hyperparameters associated with these assumptions are shown in the DGM, Figure 17. To infer the parameters of the model, approximate inference is required. Following (Gönen and Margolin 2014), a variational inference scheme is used; this utilises a lower bound on the marginal likelihood, to infer an *approximation*, denoted q, of the full joint distribution of the parameters $p(\mathbf{\Theta}, \mathbf{\Xi} \mid \{\mathbf{K}_t, \mathbf{y}_t\}_{t=1}^T)$ of the model. To achieve this, the posterior distribution is factorised as follows,

$$p\left(\mathbf{\Theta}, \mathbf{\Xi} \mid \{\mathbf{K}_t, \mathbf{y}_t\}_{t=1}^T\right) \approx q(\mathbf{\Theta}, \mathbf{\Xi}) = \prod_{t=1}^T \left(q(\mathbf{\Lambda}_t) q(\mathbf{A}_t) q(\mathbf{H}_t)\right) q(\gamma) q(\boldsymbol{\eta}) q(b, \mathbf{w}) \prod_{t=1}^T q(\mathbf{f}_t) \quad (26)$$

Each approximated factor is defined as in the full conditional distribution (Gönen and Margolin 2014). The lower bound can be optimised with respect to each factor separately, while fixing the remaining factors (iterating until convergence).

Numerical + experimental example: Shear-building structures

A numerical case study, supplemented with experimental data, is used for demonstration – an extension of the work in (Gardner et al. 2020a). A population of six different shear-building structures is considered, five are simulated, and one is experimental. A domain and task are associated with each structure (such that

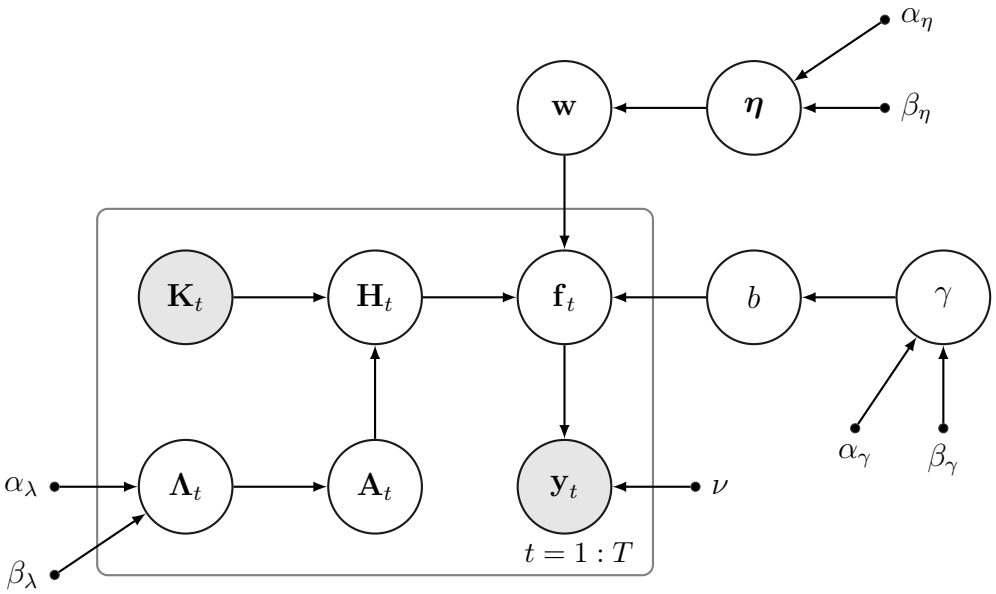


FIG. 17: Directed graphical model for binary classification KBTL.

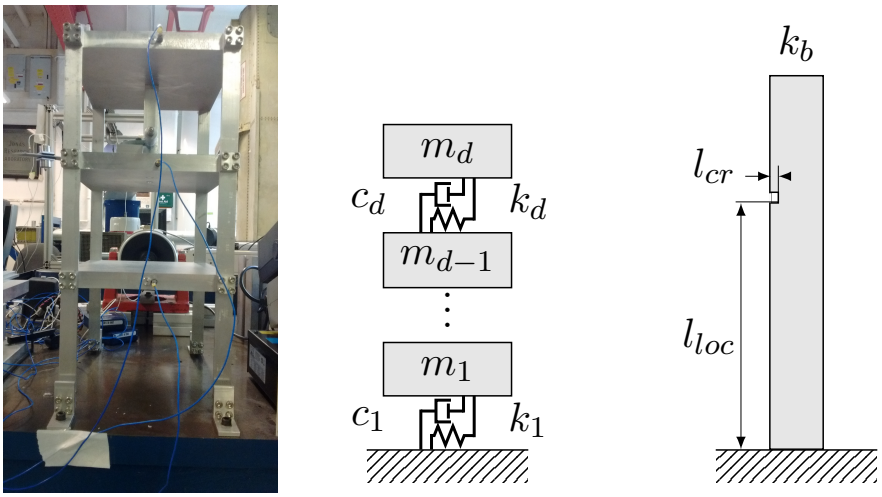


FIG. 18: Shear structures: (a) test rig; (b) a nominal representation of the five simulated systems; (c) depicts the cantilever beam component where $\{k_i\}_{i=1}^{d} = 4k_b$.

$T = 6$) – the experimental rig and (simulated) lumped-mass models are shown in Figure 18. For each structure (domain) there is a two-class classification problem (task), which is viewed as binary damage detection (normal or damaged).

Each *simulated* structure is represented by d mass, stiffness and damping coefficients, i.e. $\{m_i, k_i, c_i\}_{i=1}^{d}$. The masses have length l_m, width w_m, thickness

t_m, and density ρ. The stiffness elements are calculated from four cantilever beams in bending, $4k_b = 4(3EI/l_b^3)$, where E is the elastic modulus, I the second moment of area, and l_b the length of the beam. The damping coefficients are specified rather than derived from a physical model. Damage is simulated via an open crack, using a reduction in EI (Christides and Barr 1984). For each structure, each observation is a random draw from a base distribution for E, ρ and c. The properties of the five simulated structures are shown in Table 1.

TABLE 1: Properties of the five simulated structures. Degrees-of-freedom (DOF) are denoted d.

Domain	DOF	Beam dim.	Mass dim.	Elastic mod.	Density	Damping coeff.
(t)	(d_t)	$\{l_b,\, w_b,\, t_b\}$ mm	$\{l_m,\, w_m,\, t_m\}$ mm	E GPa	ρ kg/m^3	c Ns/m
1	4	$\{185, 25, 6.35\}$	$\{350, 254, 25\}$	$\mathcal{N}(71, 1.0 \times 10^{-9})$	$\mathcal{N}(2700, 10)$	$\mathcal{G}(50, 0.1)$
2	8	$\{200, 35, 6.25\}$	$\{450, 322, 35\}$	$\mathcal{N}(70, 1.2 \times 10^{-9})$	$\mathcal{N}(2800, 22)$	$\mathcal{G}(8, 0.8)$
3	10	$\{177, 45, 6.15\}$	$\{340, 274, 45\}$	$\mathcal{N}(72, 1.3 \times 10^{-9})$	$\mathcal{N}(2550, 25)$	$\mathcal{G}(25, 0.2)$
4	3	$\{193, 32, 5.55\}$	$\{260, 265, 32\}$	$\mathcal{N}(75, 1.5 \times 10^{-9})$	$\mathcal{N}(2600, 15)$	$\mathcal{G}(20, 0.1)$
5	5	$\{165, 46, 7.45\}$	$\{420, 333, 46\}$	$\mathcal{N}(73, 1.4 \times 10^{-9})$	$\mathcal{N}(2650, 20)$	$\mathcal{G}(50, 0.1)$

The experimental structure is constructed from aluminium 6082, with dimensions nominally similar to those in Table 1. Observational data (the first three natural frequencies) were collected via model testing, where an electrodynamic shaker applied up to 6553.6 Hz broadband white-noise excitation containing 16384 spectral lines (0.2 Hz resolution). Forcing was applied to the first storey, and three uni-axial accelerometers measured the response at all storeys. Damage was artificially introduced as a 50% saw-cut to the-mid point of the front-right beam in Figure 18a.

In each domain, the damped natural frequencies act as features, such that $\mathbf{X}_t[i,:] = \{\omega_i\}_{i=1}^d$. Therefore, as each domain has different DOFs/dimensions, heterogeneous transfer is required. The label set is consistent across all domains, corresponding to normal or damaged, i.e $y_i \in \{-1, 1\}$ respectively. The training and test data for each domain are summarised in Table 2. The training data have various degrees of class imbalance, to reflect scenarios where certain structures in SHM provide more information about a particular state.

Figure 19 shows the coupled binary classifier in the (expected) shared latent subspace for all the data $\{\mathbf{H}_t\}_{t=1}^T$. The observations associated with each of the

TABLE 2: Number of data for all domains (numerical and experimental*).

Domain	**Training**		**Testing**	
(t)	$y=-1$	$y=+1$	$y=-1$	$y=+1$
1	250	100	500	500
2	100	25	500	500
3	120	20	500	500
4	200	150	500	500
5	500	10	500	500
6*	3	3	2	2

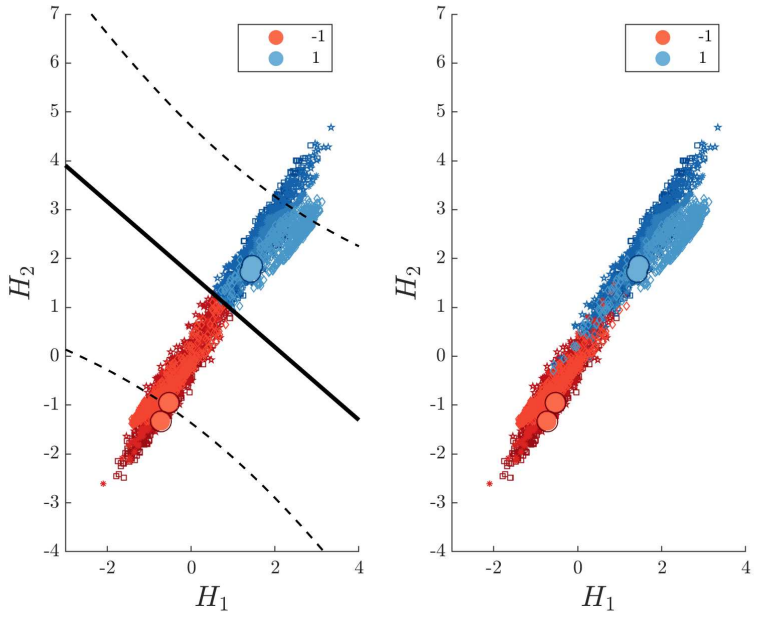


FIG. 19: The KBTL probabilistic decision boundary for the coupled classification model in the shared subspace. Markers $\{\times, \square, \star, *, \diamond, \triangle, \bullet\}$ correspond to tasks and domains $\{1, 2, 3, 4, 5, 6\}$ respectively.

six domains are distinguished via different markers. The left plot shows the test data and their predicted labels given $\mathbf{f}_t$, while the right plot shows the ground truth labels. KBTL has successfully embedded and projected data from different domains into a shared latent space ($R = 2$), where the data can be categorised by a coupled discriminative classifier. It can also be seen that, due to class imbalance (weighted towards the undamaged class -1 for each structure), there is greater uncertainty in the damaged class ($+1$), leading to more significant scatter in the latent space.

The classification results for each domain are presented in Figure 20. An observations is considered to belong to class $+1$ if $p(\mathbf{y}_t[*] = +1 \mid \mathbf{f}_t[*]) \geq 0.5$. KBTL is compared to a relevance vector machine (RVM) (Tipping 2000) as a benchmark – learnt for each domain independently. It is acknowledged that the RVM differs in implementation; however, similarities make it useful for comparison as a standard (non multi-task) alternative to KBTL.

Multi-task learning has accurately inferred a general model. For domains $\{1, 2, 3, 5, 6\}$, the SHM task is improved by considering the data from all structures in a shared latent space. In particular, extending the (effective) training data has improved the classification for domain 5. This is because there are few training data associated with the damage class for domain 5 (see Table 2); therefore, considering damage data from similar structures (in the latent space) has proved beneficial. Interestingly, for domain four ($t = 4$) there is a marginal *decrease* in the classification performance. Like domain one, domain four has *less* severe class imbalance, thus, it appears that the remaining domains (with severe class imbalance) have negatively impacted the score for this specific domain/task.

These results highlight that the data from a group (or population) of *similar* structures can be considered together, to increase the (effective) amount of training data (Bull et al. 2020a; Gosliga et al. 2020; Gardner et al. 2020b). This can lead to significant improvements in the predictive performance of SHM tools – particularly those learnt from small sets of supervised data.

CONCLUSIONS

Three new techniques for statistical inference with SHM signals have been collected and summarised (originally introduced in previous work), including partially-supervised learning (semi-supervised/active learning), Dirichlet process clustering, and multi-task learning. Primarily, each approach looks to address, from a different perspective, the issues of incomplete datasets and missing information, which lead to incomplete training-data. The algorithms consider that: a) label information (to describe what measurements represent) is likely to be incomplete; b) the available data *a priori* will usually correspond to a *subset* of the expected *in situ* conditions only. Considering the importance of uncertainty quantification in SHM, probabilistic methods are suggested, which can be (intuitively) updated to account for missing information.

The case study applications for each mode of inference highlight the potential advantages for SHM. Partially-supervised methods for active and semi-supervised

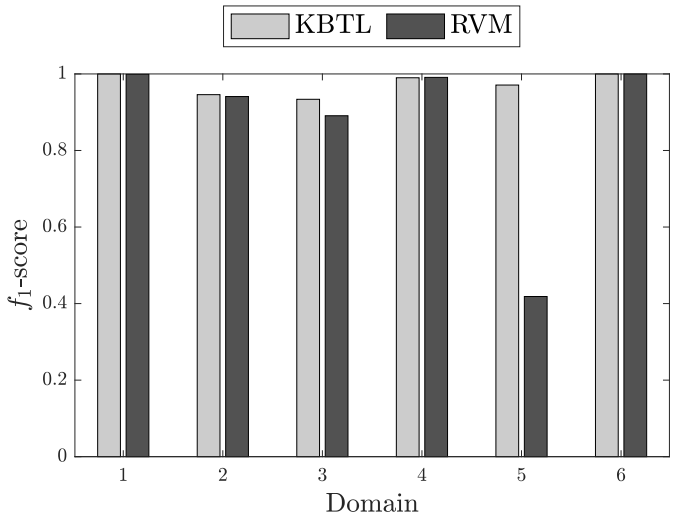


FIG. 20: KBTL classification performance, given an independent test set: f_1-scores across each domain compared to an RVM benchmark.

learning were utilised to manage the cost system inspections (to label data), while considering the unlabelled instances, both offline and online. Dirichlet process clustering has been applied to streaming data, as an unsupervised method for automatic damage detection and classification. Finally multi-task learning was applied to model shared information between systems – to extend the data available for training, this approach considers multiple (potentially incomplete) datasets associated with different tasks (structures).

DATA AVAILABILITY

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) through grant references EP/R003645/1, EP/R004900/1, EP/S001565/1 and EP/R006768/1.

REFERENCES

Aldous, D. J. (1985). "Exchangeability and related topics." *École d'Été de Probabilités de Saint-Flour XIII—1983*, Springer, 1–198.

Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

Blei, D. M., Jordan, M. I., et al. (2006). "Variational inference for Dirichlet process mixtures." *Bayesian Analysis*, 1(1), 121–143.

Bornn, L., Farrar, C. R., Park, G., and Farinholt, K. (2009). "Structural health monitoring with autoregressive support vector machines." *Journal of Vibration and Acoustics*, 131(2).

Bull, L., Gardner, P., Gosliga, J., Dervilis, N., Papatheou, E., Maguire, A., Campos, C., Rogers, T., Cross, E., and Worden, K. (2020a). "Foundations of population-based structural health monitoring, Part I: Homogeneous populations and forms." *Preprint submitted to Mechanical Systems and Signal Processing*.

Bull, L. A. (2019). "Towards probabilistic and partially-supervised structural health monitoring." Ph.D. thesis, University of Sheffield, University of Sheffield.

Bull, L. A., Manson, G., Worden, K., and Dervilis (2019a). "Active learning approaches to structural health monitoring." *Special Topics in Structural Dynamics, Volume 5*, N. Dervilis, ed., Springer International Publishing, 157–159.

Bull, L. A., Rogers, T. J., Wickramarachchi, C., Cross, E. J., Worden, K., and Dervilis, N. (2019b). "Probabilistic active learning: An online framework for structural health monitoring." *Mechanical Systems and Signal Processing*, 134, 106294.

Bull, L. A., Worden, K., and Dervilis, N. (2019c). "Damage classification using labelled and unlabelled measurements." *Structural Health Monitoring 2019*.

Bull, L. A., Worden, K., and Dervilis, N. (2020b). "Towards semi-supervised and probabilistic classification in structural health monitoring." *Mechanical Systems and Signal Processing*, 140, 106653.

Bull, L. A., Worden, K., Manson, G., and Dervilis, N. (2018). "Active learning for semi-supervised structural health monitoring." *Journal of Sound and Vibration*, 437, 373–388.

Bull, L. A., Worden, K., Rogers, T. J., Cross, E. J., and Dervilis, N. (2020c). "Investigating engineering data by probabilistic measures." *Special Topics in Structural Dynamics & Experimental Techniques, Volume 5*, Springer, 77–81.

Bull, L. A., Worden, K., Rogers, T. J., Wickramarachchi, C., Cross, E. J., McLeay, T., Leahy, W., and Dervilis, N. (2019d). "A probabilistic framework for online structural health monitoring: Active learning from machining data streams." *Journal of Physics: Conference Series*, Vol. 1264, IOP Publishing, 012028.

Cappello, C., Bolognani, D., and Zonta, D. (2015). "Mechanical equivalent of logical inference from correlated uncertain information." *Proc. of 7th International Conference on Structural Health Monitoring of Intelligent Infrastructure*.

Chakraborty, D., Kovvali, N., Chakraborty, B., Papandreou-Suppappola, A., and Chattopadhyay, A. (2011). "Structural damage detection with insufficient data using transfer learning techniques." *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, 798147.

Chapelle, O., Scholkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. MIT press.

Chatzi, E. N. and Smyth, A. W. (2009). "The unscented Kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing." *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures*, 16(1), 99–123.

Chen, S., Cerda, F., Guo, J., Harley, J. B., Shi, Q., Rizzo, P., Bielak, J., Garrett, J. H., and Kovacevic, J. (2013). "Multiresolution classification with semi-supervised learning for indirect bridge structural health monitoring." *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 3412–3416 (May).

Chen, S., Cerda, F., Rizzo, P., Bielak, J., Garrett, J. H., and Kovacevic, J. (2014). "Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring." *IEEE Transactions on Signal Processing*, 62(11), 2879–2893.

Christides, S. and Barr, A. (1984). "One-dimensional theory of cracked bernoulli-euler beams." *International Journal of Mechanical Sciences*, 26(11-12), 639–648.

Cozman, F. G., Cohen, I., and Cirelo, M. C. (2003). "Semi-supervised learning of mixture models." *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 99–106.

Dasgupta, S. (2011). "Two faces of active learning." *Theoretical Computer Science*, 412(19), 1767–1781.

de Roeck, G. (2003). "The state-of-the-art of damage detection by vibration monitoring: the SIMCES experience." *Structural Control and Health Monitoring*, 10(2), 127–134.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). "Maximum likelihood

from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1–22.

Dervilis, N., Cross, E., Barthorpe, R., and Worden, K. (2014). "Robust methods of inclusive outlier analysis for structural health monitoring." *Journal of Sound and Vibration*, 333(20), 5181–5195.

Dorafshan, S., Thomas, R. J., and Maguire, M. (2018). "Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete." *Construction and Building Materials*, 186, 1031–1045.

Farrar, C. R. and Worden, K. (2012). *Structural Health Monitoring: A Machine Learning Perspective*. John Wiley & Sons.

Flynn, E. B. and Todd, M. D. (2010). "A Bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing." *Mechanical Systems and Signal Processing*, 24(4), 891–903.

Gao, Y. and Mosalam, K. M. (2018). "Deep transfer learning for image-based structural damage recognition." *Computer-Aided Civil and Infrastructure Engineering*, 33(9), 748–768.

Gardner, P., Bull, L., Dervilis, N., and Worden, K. (2020a). "Kernelised Bayesian transfer learning for population-based structural health monitoring." *Proceedings of IMAC XXXVIII, the 38th International Modal Analysis Conference*, Springer.

Gardner, P., Bull, L., Gosliga, J., Dervilis, N., and Worden, K. (2020b). "Foundations of population-based structural health monitoring, part III: Heterogeneous populations – mapping and transfer." *Preprint submitted to Mechanical Systems and Signal Processing*.

Gardner, P., Bull, L. A., Dervilis, N., and Worden, K. (2020c). "A sparse Bayesian approach to heterogeneous transfer learning for population-based structural health monitoring." *Submitted to Mechanical Systems and Signal Processing*.

Gardner, P., Liu, X., and Worden, K. (2020d). "On the application of domain adaptation in structural health monitoring." *Mechanical Systems and Signal Processing*, 138, 106550.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC.

Gönen, M. and Margolin, A. (2014). "Kernelized Bayesian transfer learning." *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Gosliga, J., Gardner, P., Bull, L., Dervilis, N., and Worden, K. (2020). "Founda-

tions of population-based structural health monitoring, part II: Heterogeneous populations – graphs, networks and communities." *Preprint submitted to Mechanical Systems and Signal Processing*.

Huang, Y., Beck, J. L., and Li, H. (2019). "Multitask sparse bayesian learning with applications in structural health monitoring." *Computer-Aided Civil and Infrastructure Engineering*, 34(9), 732–754.

Jang, K., Kim, N., and An, Y. (2019). "Deep learning-based autonomous concrete crack evaluation through hybrid image scanning." *Structural Health Monitoring*, 147592171882171.

Janssens, O., Van de Walle, R., Loccufier, M., and Van Hoecke, S. (2017). "Deep learning for infrared thermal image based machine health monitoring." *IEEE/ASME Transactions on Mechatronics*, 23(1), 151–159.

Kremer, J., Steenstrup, K. P., and Igel, C. (2014). "Active learning with support vector machines." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(4), 313–326.

MacKay, D. J. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

Manson, G., Worden, K., and Allman, D. (2003). "Experimental validation of a structural health monitoring methodology: Part III. damage location on an aircraft wing." *Journal of Sound and Vibration*, 259(2), 365–385.

McCallumzy, A. K. and Nigamy, K. (1998). "Employing EM and pool-based active learning for text classification." *Proc. International Conference on Machine Learning (ICML)*, Citeseer, 359–367.

Murphy, K. P. (2012). *Machine Learning: a Probabilistic Perspective*. MIT press.

Neal, R. M. (2000). "Markov chain sampling methods for Dirichlet process mixture models." *Journal of Computational and Graphical Statistics*, 9(2), 249–265.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. (1998). "Learning to classify text from labeled and unlabeled documents." *AAAI/IAAI*, 792, 6.

Ou, Y., Chatzi, E. N., Dertimanis, V. K., and Spiridonakos, M. D. (2017). "Vibration-based experimental damage detection of a small-scale wind turbine blade." *Structural Health Monitoring*, 16(1), 79–96.

Pan, S. J. and Yang, Q. (2009). "A survey on transfer learning." *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359.

Papoulis, A. (1965). *Probabilities, Random Variables, and Stochastic Processes*.

McGraw-Hill.

Peeters, B. and de Roeck, G. (2001). "One-year monitoring of the Z24-bridge: environmental effects versus damage events." *Earthquake Engineering & Structural Dynamics*, 30(2), 149–171.

Rasmussen, C. E. (2000). "The infinite Gaussian mixture model." *Advances in Neural Information Processing Systems*, 554–560.

Rasmussen, C. E. and Ghahramani, Z. (2001). "Occam's razor." *Advances in neural information processing systems*, 294–300.

Rippengill, S., Worden, K., Holford, K. M., and Pullin, R. (2003). "Automatic classification of acoustic emission patterns." *Strain*, 39, 31–41.

Rogers, T. J., Worden, K., Fuentes, R., Dervilis, N., Tygesen, U. T., and Cross, E. J. (2019). "A Bayesian non-parametric clustering approach for semi-supervised structural health monitoring." *Mechanical Systems and Signal Processing*, 119, 100 – 119.

Rousseeuw, P. J. and Driessen, K. V. (1999). "A fast algorithm for the minimum covariance determinant estimator." *Technometrics*, 41(3), 212–223.

Schwenker, F. and Trentin, E. (2014). "Pattern classification and clustering: a review of partially supervised learning approaches." *Pattern Recognition Letters*, 37(1), 4–14.

Settles, B. (2012). "Active learning." *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1), 1–114.

Sohn, H., Farrar, C. R., Hemez, F. M., Shunk, D. D., Stinemates, D. W., Nadler, B. R., and Czarnecki, J. J. (2003). "A review of structural health monitoring literature: 1996–2001." *Los Alamos National Laboratory, USA*.

Tipping, M. E. (2000). "The relevance vector machine." *Advances in Neural Information Processing Systems*, 652–658.

Vanik, M. W., Beck, J. L., and Au, S. (2000). "Bayesian probabilistic approach to structural health monitoring." *Journal of Engineering Mechanics*, 126(7), 738–745.

Vlachos, A., Korhonen, A., and Ghahramani, Z. (2009). "Unsupervised and constrained Dirichlet process mixture models for verb clustering." *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, Association for Computational Linguistics, 74–82.

Wan, H. and Ni, Y. (2019). "Bayesian multi-task learning methodology for reconstruction of structural health monitoring data." *Structural Health Monitoring*,

18, 1282–1309.

Wang, M., Min, F., Zhang, Z.-H., and Wu, Y.-X. (2017). "Active learning through density clustering." *Expert Systems with Applications*, 85, 305–317.

Worden, K. and Manson, G. (2006). "The application of machine learning to structural health monitoring." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851), 515–537.

Worden, K., Manson, G., Hilson, G., and Pierce, S. (2008). "Genetic optimisation of a neural damage locator." *Journal of Sound and Vibration*, 309(3), 529–544.

Ye, J., Kobayashi, T., Tsuda, H., and Murakawa, M. (2017). "Robust hammering echo analysis for concrete assessment with transfer learning." *Proceedings of the The 11th International Workshop on Structural Health Monitoring*, 943–949.

Zhang, Y. and Yang, Q. (2018). "An overview of multi-task learning." *National Science Review*, 5(1), 30–43.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., and Gao, R. X. (2019). "Deep learning and its applications to machine health monitoring." *Mechanical Systems and Signal Processing*, 115, 213–237.

Zhu, X. J. (2005). "Semi-supervised learning literature survey." *Report no.*, University of Wisconsin-Madison Department of Computer Sciences.

Zonta, D., Glisic, B., and Adriaenssens, S. (2014). "Value of information: impact of monitoring on decision-making." *Structural Control and Health Monitoring*, 21(7), 1043–1056.