Binary Tree SVM Based Framework for Mining Fatigue Induced Damage Attributes in Complex Lug Joints

Clyde K. Coelho*a, Santanu Dasb, Aditi Chattopadhyaya,

aDepartment of Mechanical and Aerospace Engineering, Arizona State University, Tempe, Arizona, USA 85287-6106
bNASA Ames Research Center, Building 269, Moffett Field, CA  94035

ABSTRACT

Research is being conducted in damage diagnosis and prognosis to develop state awareness models and residual useful life estimates of aerospace structures. This work describes a methodology using Support Vector Machines (SVMs), organized in a binary tree structure to classify the extent of a growing crack in lug joints. A lug joint is a common aerospace ‘hotspot’ where fatigue damage is highly probable. The test specimen was instrumented with surface mounted piezoelectric sensor/actuators and then subjected to fatigue load until failure. A Matching Pursuit Decomposition (MPD) algorithm was used to preprocess the sensor data and extract the input vectors used in classification. The results of this classification scheme show that this type of architecture works well for categorizing fatigue induced damage (crack) in a computationally efficient manner. However, due to the nature of the overlap of the collected data patterns, a classifier at each node in the binary tree is limited by the performance of the classifier that is higher up in the tree.

Keywords: Damage classification, support vector machines, matching pursuit decomposition, structural health monitoring, binary tree.

INTRODUCTION

Diagnosing faults that could lead to catastrophic failures in service structures is of utmost importance in structural health monitoring (SHM). An effective early warning system will allow the operators of these systems to do preventative maintenance and make decisions about the remaining useful life of the structure. Lug joints have been identified as structural hotspots in various aerospace systems due to their complex geometries that lead to multiple stress concentration points. Subjecting these samples to fatigue loading showed different failure points depending on the applied load and surface finish. This highlights the need for robust inspection schemes on such components. The health monitoring scheme used in this research is based on the excitation and reception of guided Lamb waves in a structure by piezoelectric transducers.

Support Vector Machines, an advanced machine learning based technique, has been utilized in this paper because it provides good generalization when few samples are available for training purposes. Also, SVMs have a strong mathematical foundation and has been widely used for classification in a number of fields. Although the original design of SVMs by Vapnik consisted of a two class structure, this approach has been expanded to do multiclass classification. Classification of multiclass problems has been conducted in the form of ‘one vs one’, ‘one vs rest’, hybrid and clustering algorithms. In ‘one vs one’, the amount of training time required is very large since classifiers need to be constructed for a k class problem. For a ‘one vs rest’ scheme, a problem involving k classes of data will require the construction of k classifiers. One problem with the latter case is that each classifier will require the use of the entire training set which becomes computationally intractable. For both methods, a voting scheme is used in which the classifier that scores the highest for a given data set assigns all the points in that set to a given class. Also, in such a case, it is very difficult to decide which class the test data belongs to if two classifiers have similar scores. Clustering schemes are able to learn signal characteristics well and can decide the uniqueness of different classes (or even classes within classes) based on the clustering of data points in hyperspace. While this approach is promising for damage detection scenarios where all possible damage types cannot be known, the computational expense involved with determining the cluster boundaries increases exponentially with the increase in training sets.

clyde.coelho@asu.edu; phone 1 480 965 2434; fax 1 480 965 1384
This paper applies a ‘one vs rest’ scheme\(^2\) organized into a binary tree structure that solves some of the aforementioned problems while accurately classifying crack damage. A lug joint was instrumented with surface mounted actuators/sensors and then subjected to fatigue loading. Data was collected at various damage states for analysis. The damage states were quantified using an optical telescope. A matching pursuit decomposition\(^7\) algorithm is used to extract the features of the collected signal before it could be classified using the binary tree SVMs framework.

This paper is organized as follows. Sections 2 and 3 presents a theoretical background on the method used for classification and its organization as a binary tree classifier. The theory behind the extraction of features that were used for classification is presented in section 4. The experimental setup and some details regarding data collection have been discussed in section 5. Section 6 demonstrates the effectiveness of the classification scheme through some selected results from a fatigue lug joint. Section 7 presents some observations made from the present study along with possibilities for further improvement of the current work.

2. SUPPORT VECTOR MACHINES

Support Vector Machines have been used for classification in a number of different fields because of their good generalization ability which means they are able to learn the behavior of the system even with relatively few example behaviors of the system. The ability of SVMs to separate nonlinearly separable data is based on Cover’s Theorem\(^8\) which states that non-separable or non-linearly separable patterns in input space (low dimensional space) is more likely to be linearly separable in a new high-dimensional feature space, provided that the transformation is non linear and the dimensionality of the feature space is high enough. The patterns in this high dimensional (say \(N\) dimensional) space are then separated by constructing an \(N-1\) dimension hyperplane which takes into account, the nonlinear relationship of the data. The mapping kernel used in this research is the Radial Basis Function (RBF), which is popular in machine learning applications\(^9,10\) involving data sets that are not linearly separable. The RBF kernel takes the form of

\[
K(x, x_i) = e^{-\frac{||x-x_i||^2}{2\sigma^2}}
\]

where \(x\) is the input vector and \(x_i\) is the \(i\)th input pattern, and \(\sigma\) is the width of the kernel that has been optimized during the training phase. Once the training data has been mapped into high dimensional space, an optimal hyperplane is constructed. In the feature space, there exists multiple hyperplanes that can separate the mapped features from two different classes but only one will maximize\(^8\) the margin of separation between feature vectors with positive and negative labels (Fig.1). The classifier has better generalization when the margin between the classes is larger, allowing the classifier to ‘learn’ the characteristic behavior of the system.

![Figure 1. (a) Possible hyperplane with small margin between classes. (b) Possible hyperplane with larger margin of separation.](image)

The optimal hyperplane for patterns that are linearly separable is defined as,
\[ w \cdot z_i + b = 0, \]  
\[ y_i(w \cdot z_i + b) \geq 1. \]

In most practical applications, the data is nonlinearly separable and it is not possible to construct a hyperplane without admissible training errors. In such a case, a soft margin\(^1\) is implemented (Fig. 2) and the above equation can be modified as,

\[ y_i(w \cdot z_i + b) \geq 1 - \xi_i, \]

where \( \xi_i \geq 0 \ \forall \ i \) is a slack variable. In order to find this optimal hyperplane which minimizes the classification error, the following optimization problem needs to be solved

\[ \min \frac{1}{2} ||w||^2 + D \sum_i \xi_i \]

subject to the constraint shown in Eqn. (4). The variable \( D \) refers to a regularization parameter that can be modified to control the complexity of the model. A large value of \( D \) means that the classifier will only classify separable data. In this research, the value of \( D \) was chosen to be 1 since the data used for this paper is not completely separable. Defining \( w(\alpha) = \sum_i \alpha_i y_i z_i \)\(^{12}\), the dual problem can be constructed as,

\[ \max W(\alpha) = \sum_i \alpha_i - \frac{1}{2}w(\alpha) \cdot w(\alpha), \]

subject to \( 0 \leq \alpha_i \leq C \ \forall \ i \) and \( \sum_i \alpha_i y_i = 0 \). Solving Eqn. (6) for the Lagrange Multipliers \( \alpha \), it is possible to recover the solution to the primal problem. The decision function for the classifier becomes,

\[ y(x) = \text{sign} \{ \sum_i \alpha_i y_i K(x, x_i) + b \}. \]

\[ \text{Figure 2: Representation of parameters needed for hyperplane construction in two dimensions.} \]
3. SVMS AS A BINARY TREE CLASSIFIER

The multi-class classifier used in this paper combines a modified ‘one vs rest’ algorithm with a binary tree structure to minimize the number of comparisons that are necessary to identify a data class while still taking into account all possible classes. For a four class problem as shown in Fig 3, the binary tree classifier is setup as follows.

1. A two class SVM is trained using pattern 1 as Class A and patterns 2, 3 and 4 as Class B and a hyperplane is constructed.
2. Next, data points corresponding to pattern 1 are removed from the training set and pattern 2 is taken as Class A and patterns 3 and 4 are taken as Class B for hyperplane construction.
3. This process is repeated until the last classifier compares only two patterns.

![Figure 3: Construction of multiple hyperplanes without overlapping regions for multi-class problems.](image)

The advantage of this approach is that for a $k$ class problem, only $(k-1)$ hyperplanes need to be constructed. Also, removing patterns after each classifier is constructed reduces the computational expense. In this way, it is also possible to prioritize damage classes and terminate the classification algorithm during testing before checking all possible cases. In this paper, since the damage considered is only of one type, the classifiers are arranged in order of increasing crack length. In a more complex structure like a bolted joint for example, it would be possible to prioritize torque loss over crack damage and the algorithm can be terminated midway if a loose bolt is detected. In order to ensure there is no region of overlap one point could be classified as belonging to multiple classes, a point in the first comparison that is classified as belonging to Class A is removed from the test set and the points in Class B move on to the next classifier.

4. FEATURE EXTRACTION ALGORITHM

As with all machine learning based approaches, the outcomes obtained are heavily dependent on the type of features used to classify the data. In this research, matching pursuit decomposition has been used as a feature extraction tool. MPD has been used for various applications such as feature extraction, signal characterization and classification, and signal encoding and reconstruction. The working principle of MPD relies on decomposing a given signal into linear expansions of elementary functions (or atoms). The resulting decomposition reveals the waveform’s time-frequency structure. The algorithm looks at the signal and tries to match them to a comprehensive set of dictionary elements to guarantee an accurate decomposition of the waveform. During this comparison, if a perfect match is not obtained, the closest atom in the dictionary is selected and then subtracted from the signal to obtain the residue. This process is repeated for successive iterations until most of the information from the signal is decomposed or certain stopping criteria is satisfied. In this research, the dictionary elements were composed of Gabor atoms, normalized in both the time and frequency domain. These atoms were selected since they are concentrated in the time-frequency domain and there exist closed-form analytical time-frequency representations for such atoms. Also, the algorithm is guaranteed to converge if the dictionary used is complete and the atoms have unitary energy. The decomposition of the signal is based on four variables that define each dictionary element: expansion coefficient ($C$), time shift ($\tau$), frequency shift ($f$) and atom width ($k$). The expression for the atoms used is given by

$$g_i(t) = e^{-k^2(t-\tau)^2} \cos(2\pi f t).$$ (8)

Using these atoms, the decomposition after $M$ iterations can be written as,
\[ x(t) = \sum_{i=0}^{M-1} C_i g_i + R^M x(t), \tag{9} \]

where \( R^M x(t) \) is the residue of the signal after decomposing the signal \( M \) times and \( R^0 x(t) \) is the original signal. As \( M \to \infty \), the signal residue will go to zero and the entire signal will be decomposed, i.e.

\[ \lim_{M \to \infty} \|R^M x(t)\|_2 = \lim_{M \to \infty} \|x(t) - \sum_{i=0}^{M-1} C_i g_i\|_2 = 0. \tag{10} \]

This procedure is adopted because it reduces a given signal into fewer representative components that are more easily classified. Also, for physical systems the number of iterations can be limited so that the part of the signal that contains information is decomposed while the noise is contained in the residue. Figure 4 shows the steps involved in the matching pursuit decomposition algorithm. First, the weighted contribution of the dictionary element that best matches the signal (or the residue) is calculated. The dictionary element that has the highest time correlation with the signal is then selected and the weighted element is then extracted from the signal. The signal residue that is left is then put back into the algorithm until the stopping criteria is reached. The stopping criteria can be defined in terms of the minimum energy that is extracted from the signal or the total number of iterations of the algorithm. For this research, each signal was decomposed using 15 iterations which corresponded with 84% of the total signal energy. It was found that adding more iterations did not add much information to the result that would help in classification.

| Step-1: Initialize Stopping criteria And Assign test signal \( x^0[n] = x[n] \) |
| Step-2: Weighted contribution of best dictionary element: \( a^k = \langle x^k, d^k \rangle \) |
| Step-3: Select the dictionary element whose time correlation with the test signal is maximum |
| Step-4: Compute Residual Signal: \( x^k[n] = x^k[n] - a^k d^k[n] \) |
| Step-5: If(Stopping criteria satisfied) Stop Else Increment iteration number Go to Step-2 End |

Figure 4: Matching pursuit decomposition algorithm.

The flowchart in Fig. 5 shows the steps used to analyze the data collected from the sensors. First the raw data is passed through the matching pursuit decomposition algorithm where features that better represent the characteristics of the system which change with damage are extracted. Next, these meaningful parameters are mapped into a high dimensional space where they are separated into different classes.

![Figure 5: Main steps of the nested binary tree classification algorithm](image)

5. EXPERIMENTAL SETUP

The specimen tested was a lug joint that was subjected to tensile fatigue loading as shown in Fig. 4(a). The sample was machined out of Al 2024 T351. One surface of the lug joint was polished using 1200 grit silicon carbide paper so that more accurate measurements of crack length could be made using an optical telescope. The sample was tested at a load ratio of 0.1 with a maximum load of 1100lbs at 20Hz using an Instron 1331 servohydraulic test frame. Figure 4(b) shows the different modes of failure that were observed when the maximum load was changed. Failure mode 1 occurred at the load mentioned and the data analyzed for this case is presented. Failure mode 2 occurred when the maximum load was reduced to 800lbs under the same load ratio and frequency. Images of the crack length were taken every time the test was halted for data collection from the piezoelectric transducers using a CCD camera.
For the active detection scheme used in this research, a 130 kHz, Gaussian windowed sine wave was used as the excitation signal. The duration of the excitation was 500µs. The data collected from the sensors was sampled at 2MHz. Before preprocessing, each observation was downsampled to 500 kHz with a signal length of 512 points. Downsampling was feasible since the excitation was narrow band and most of the components of the sensor signal were between 100 kHz and 150 kHz and the Nyquist frequency was still well above the maximum frequency component of the signal. It also made the matching pursuit algorithm more computationally efficient as the required dictionary size is reduced. A total of 300 observations were taken every time the damage state was measured.

6. RESULTS AND DISCUSSION

The fatigue experiment carried out on the lug joint resulted in five different damage states being measured. The damage states differed in the length of the crack that was present. The different damage states corresponded to:

- a) C1 → Healthy
- b) C2 → 6mm crack
- c) C3 → 8mm crack
- d) C4 → 10mm crack
- e) C5 → 12mm crack

The data collected from the experiments were first passed through the MPD algorithm to extract features that are more easily classifiable. Figure 5(a), shows a principal component analysis (PCA) of the raw signals. It can be seen from this figure that there is a tremendous amount of overlap and that accurate separation of these points in this form will be extremely difficult. The true dimension of the data being analyzed is 512. Figure 5(b) shows a PCA plot of the same signals after feature extraction. The feature extraction procedure reduced the dimension of the data from 512 to 60. It can be seen that even though there is still some overlap in the points, separation of these points in high dimensional space is made easier.
The training of each classifier in the binary tree was completed using 200 examples that belonged to either side of the hyperplane. In the case of the first classifier for example, 200 randomly selected data points from C1 and 50 randomly selected points from each of the C2-C5 classes were used to construct the first decision plane. The testing of the classifier was completed using 100 data points from each damage class. The results of the classification algorithm are presented in the nested binary confusion matrix (Table 1). From Table 1, it can be seen that the classifier performs extremely well in identifying data points that belong to C1, especially considering that a 15% error was permitted when training the classifier. This percentage was selected because of the nature of the overlap of the data patterns. It also prevented the classifier from ‘over-fitting’ the hyperplane to the data resulting in a loss of generalization.

For data points belonging to the other classes, a small but significant portion of the data was misclassified as belonging to C1. Since points that are positively classified are removed from the test set before further classification, the elements in every column are classification results from a smaller set of data. As an example, the number of points in C2 that were correctly classified was 70 out of 80 test points. A drawback of this classification scheme is that the results of a classifier are dependent on the performance of classifiers that are evaluated before it.

In order to better visualize the data overlap, a histogram of the normalized distance of all the points from the hyperplane is constructed (Fig. 6). The histogram clearly shows that when testing the decision hyperplane, there are relatively few points from C1 that are mistakenly classified but there is a much larger number of points belonging to C2-C5 that fall into the C1 side of the decision plane. This causes the relatively large misclassification of points belonging to C2-C5.
Figure 6: Histogram of distance from the optimal hyperplane.

Figure 7 shows the receiver operating characteristic (ROC) curve for each of the constructed classifiers. An ROC curve is a robust way to test the ability of a classifier to discriminate between classes. It allows a user to weigh the cost savings from maintaining or replacing a part in before failure (true positive) against the added cost of replacing a part when it is still undamaged (false positive). An ideal classifier would have a point at (0,1) which means that the classifier was able to correctly identify all the damage states and there was no overlap in the data patterns. The five classes being studied result in the construction of four classifiers that are constructed at different levels of the binary tree. The curves for each classifier represent the performance of each individual classifier and should not be used to judge the performance of the entire classification scheme. It can be seen from this plot that all of the classifiers have a performance that is well above the line of no-discrimination.
7. CONCLUSIONS

A SVMs based classifier organized into a binary tree structure has been used to classify the damage caused by fatigue in a lug joint. The MPD algorithm was used to decompose the signal into a set of significant features that were more easily classifiable. It was shown that this system was able to classify the damage state of the system with good accuracy. The computational efficiency of this binary tree framework lies in the reduced number of comparisons required and the smaller amount of required for training subsequent classifiers. The data set used for classification in this paper had significant overlap because the fatigue damage in the structure caused only small changes in the sensor response. Due to this, the performance of classifiers in lower nodes of the binary tree is heavily dependent of the performance of classifiers higher up in the tree. The next step in the development of this classification architecture would be to develop a scheme to reclassify points that are initially misclassified due to overlapping data patterns.

ACKNOWLEDGEMENT

This research was supported by the MURI Program, Air Force Office of Scientific Research, grant number: FA9550-06-1-0309; Technical Monitor, Dr. Victor Giurgiutiu.

REFERENCES
