Signal Processing 92 (2012) 625–635

Contents lists available at SciVerse ScienceDirect

Signal Processing

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

Optimization of symbolic feature extraction for pattern classification ☆

Soumik Sarkar, Kushal Mukherjee, Xin Jin, Dheeraj S. Singh, Asok Ray *

Mechanical Engineering Department, The Pennsylvania State University, University Park, PA 16802, USA

ARTICLE INFO

Article history:
Received 28 January 2011
Received in revised form
24 June 2011
Accepted 19 August 2011
Available online 12 September 2011

Keywords:
Time-series analysis
Feature extraction
Pattern classification
Symbolic dynamic filtering
Partitioning optimization

ABSTRACT

The concept of symbolic dynamics has been used in recent literature for feature extraction from time series data for pattern classification. The two primary steps of this technique are partitioning of time series to optimally generate symbol sequences and subsequently modeling of state machines from such symbol sequences. The latter step has been widely investigated and reported in the literature. However, for optimal feature extraction, the first step needs to be further explored. The paper addresses this issue and proposes a data partitioning procedure to extract low-dimensional features from time series while optimizing the class separability. The proposed procedure has been validated on two examples: (i) parameter identification in a Duffing system and (ii) classification of fatigue damage in mechanical structures, made of polycrystalline alloys. In each case, the classification performance of the proposed data partitioning method is compared with those of two other classical data partitioning methods, namely uniform partitioning (UP) and maximum entropy partitioning (MEP).

1. Introduction

Early detection of anomalies (i.e., deviations from the nominal behavior) in human-engineered systems is essential for prevention of catastrophic failures, performance enhancement, and survivability. Often, the success of data-driven anomaly detection depends on the quality of information extraction from sensor time-series; the problem of handling time series accrues from the data volume and the associated computational complexity. Unless the data sets are appropriately compressed into low-dimensional features, it is almost impractical to use such databases. In general, feature extraction is considered as the process of transforming high-dimensional data to be represented in a low-dimensional feature space with no significant loss of class separability. To this end, several tools of feature extraction, such as principal component analysis (PCA) [1], independent component analysis (ICA) [2], kernel PCA [3], and semi-definite embedding [4], have been reported in the literature. Among nonlinear methods for feature extraction and dimensionality reduction, commonly used ones are neighborhood-based graphical methods [5] and local embedding methods [6]. Recent literature has reported another nonlinear method, namely symbolic dynamic filtering (SDF) [7] for feature extraction and pattern classification [8] from time-series, which consists of the following four major steps:

1. Generation of symbol sequences via partitioning of the time series data sets.
2. Construction of probabilistic finite state automata (*PFSA*) from the respective symbol sequences.

☆ This work has been supported in part by NASA under Cooperative Agreement no. NNX07AK49A, and by the U.S. Army Research Office under Grant no. W911NF-07-1-0376. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

* Corresponding author. Tel.: +1 814 865 6377.
E-mail addresses: szs200@psu.edu (S. Sarkar), kum162@psu.edu (K. Mukherjee), xuj103@psu.edu (X. Jin), dss240@psu.edu (D.S. Singh), axr2@psu.edu (A. Ray).

0165-1684/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2011.08.013

3. Extraction of features as probability morph matrices or as state probability vectors from *PFSA*.
4. Pattern classification based on the extracted features.

In the above context, SDF serves as a tool for compressing and transferring information pertaining to a dynamical system from the space of time-series data to a low-dimensional feature space. Feature extraction and pattern classification algorithms, based on SDF, have been experimentally validated for real-time execution in different applications. Algorithms, constructed in the SDF setting, are shown to yield superior performance in terms of early detection of anomalies and robustness to measurement noise in comparison with other existing techniques such as principal component analysis (PCA), neural networks (NN) and Bayesian techniques [9].

While the properties and variations of transformation from a symbol space to a pattern space have been thoroughly studied in the disciplines of mathematics, computer science and especially data mining, similar efforts have not been expended to investigate partitioning of time series data to optimally generate symbol sequences for pattern classification. Steuer et al. [10] reported a comparison of maximum entropy partitioning and uniform partitioning; it was concluded that maximum entropy partitioning is a better tool for change detection in symbol sequences than uniform partitioning. Symbolic false nearest neighbor partitioning (SFNNP) [11] optimizes a generating partition by avoiding topological degeneracy. However, SFNNP may become extremely computation intensive if the dimension of the phase space of the underlying dynamical system is large. Furthermore, if the time series data become noise-corrupted, the symbolic false neighbors rapidly grow in number and may erroneously require a large number of symbols to capture pertinent information on the system dynamics. The wavelet transform largely alleviates the above shortcoming and is particularly effective with noisy data for large-dimensional dynamical systems [12]. Subbu and Ray [13] introduced a Hilbert-transform-based analytic signal space partitioning (ASSP) as an alternative to the wavelet space partitioning (WSP). Sarkar et al. [14] generalized ASSP for symbolic analysis of noisy signals. Nevertheless, these partitioning techniques primarily attempt to provide an accurate symbolic representation of the underlying dynamical system under a given quasi-stationary condition, rather than trying to capture the data-evolution characteristics due to a fault in the system. The goal of this paper is to overcome the difficulties of the above-mentioned partitioning methods with the objective of making SDF a robust time-series feature extraction tool for enhancement of pattern classification performance.

In the current SDF methodology [7], a time series at the nominal condition is partitioned to construct a framework for generating patterns from time series at (possibly) off-nominal conditions. Recently, Jin et al. [8] have reported the theory and validation of a wavelet-based feature extraction tool that used maximum entropy partitioning of the space of wavelet coefficients. Even if this partitioning is optimal (e.g., in terms of maximum entropy or some other criteria) under nominal conditions, it may not remain optimal at other conditions. The key idea of the work reported in this paper is to take advantage of non-stationary dynamics (in a slower scale) and optimize the partitioning process based on the statistical changes in the time series over a given set of training data belonging to different classes. This concept has been validated on two examples: (i) parameter identification in a Duffing system [15] and (ii) classification of fatigue damage in mechanical structures, made of polycrystalline alloys [16]. In each case, the classification performance of the proposed data partitioning method is compared with those of two other data partitioning methods, namely uniform partitioning (UP) and maximum entropy partitioning (MEP). Major contributions of the paper are delineated below:

1. Partitioning of time series for optimization of pattern classification.
2. Construction of a cost function to incorporate trade-offs among sensitivity to changes in data characteristics, robustness to spurious disturbances, and quantization error by using fuzzy partitioning cell boundaries.
3. Validation of the proposed concepts on a simulation test bed of a nonlinear Duffing system for multiple parameter identification [7] and on a computer-instrumented and computer-controlled fatigue test machine for fatigue damage classification.

The paper is organized into six sections including the present one. Section 2 presents a brief background of SDF in the context of feature extraction and classification. The partitioning optimization scheme is elaborated in Section 3 along with its key features. Section 4 validates the proposed concepts on a simulation test bed of a second order non-autonomous forced Duffing equation [15]; the second validation example dealing with fatigue damage classification is presented in Section 5. Section 6 summarizes the paper and makes major conclusions along with recommendations for future research.

2. Symbolic dynamic filtering (SDF)

Although the methodology of symbolic dynamic filtering (SDF) has been reported in recent literature [7,12,13], a brief outline of the procedure is succinctly presented here for completeness of the paper.

2.1. Partitioning: a nonlinear feature extraction technique

Symbolic feature extraction from time series data is posed as a two-scale problem, as depicted in Fig. 1. The *fast scale* is related to the response time of the process dynamics. Over the span of data acquisition, dynamic behavior of the system is assumed to remain invariant, i.e., the process is statistically quasi-stationary on the fast scale. In contrast, the *slow scale* is related to the time span over which non-stationary evolution of the system dynamics may occur. It is expected that the features extracted from the fast-scale data will depict statistical changes between two different slow-scale epochs if the

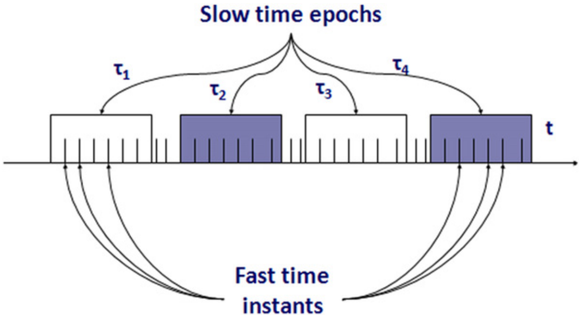

Fig. 1. Pictorial view of the two time scales: (i) *slow time scale* of anomaly evolution and (ii) *fast time instants* of data acquisition.

underlying system has undergone a change. The method of extracting features from quasi-stationary time series on the fast scale is comprised of the following steps:

- Sensor time series, denoted as $\mathbf{q}$, is generated at a slow-scale epoch from a physical system or its dynamical model. A compact (i.e., closed and bounded) region $\Omega \in \mathbb{R}^n$, where $n \in \mathbb{N}$, within which the (quasi-stationary) time series is circumscribed, is identified. Let the space of time series data sets be represented as $\mathcal{Q} \subseteq \mathbb{R}^{n \times N}$, where the $N \in \mathbb{N}$ is sufficiently large for convergence of statistical properties within a specified threshold. (Note: n represents the dimensionality of the time-series and N is the number of data points in the time series.)
- Encoding of Ω is accomplished by introducing a partition $\mathbb{B} \equiv \{B_0, \ldots, B_{(m-1)}\}$ consisting of m mutually exclusive (i.e., $B_j \cap B_k = \emptyset \; \forall j \neq k$), and exhaustive (i.e., $\bigcup_{j=0}^{m-1} B_j = \Omega$) cells. Let each cell be labeled by symbols $s_j \in \Sigma$, where $\Sigma = \{s_0, \ldots, s_{m-1}\}$ is called the alphabet. This process of coarse graining can be executed by uniform, maximum entropy, or any other scheme of partitioning. Then, the time series data points in $\{\mathbf{q}\}$ which visit the cell B_j are denoted as $s_j \; \forall j = 0, 1, \ldots, m-1$. This step enables transformation of the time series data $\{\mathbf{q}\}$ to a symbol sequence $\{\mathbf{s}\}$.
- A probabilistic finite state automaton (*PFSA*) is then constructed from the symbol sequence $\{\mathbf{s}\}$, where $j,k \in \{1, 2, \ldots, r\}$ are the states of the *PFSA* with the $(r \times r)$ state transition matrix $\Pi = [\pi_{jk}]$ that is obtained at slow-scale epochs (*Note*: Π is a stochastic matrix, i.e., the transition probability $\pi_{jk} \geq 0$ and $\sum_k \pi_{jk} = 1$). To compress the information further, the state probability vector $\mathbf{p} = [p_1 \ldots p_r]$ that is the left eigenvector corresponding to the (unique) unity eigenvalue of the irreducible stochastic matrix Π is calculated. The vector $\mathbf{p}$ is the extracted feature vector and is a low-dimensional compression of the long time series data representing the dynamical system at the slow-scale epoch.

2.2. *Classification using low-dimensional feature vectors*

For classification using SDF, the reference time series, belonging to a class denoted as Cl_1, is symbolized by one of the standard partitioning schemes (e.g., uniform partitioning (UP) or maximum entropy partitioning (MEP)) [7,12,13]. Then, using the steps described in Section 2.1, a low-dimensional feature vector $\mathbf{p}^{Cl_1}$ is constructed for the reference slow-scale epoch. Similarly, from a time series belonging to a different class denoted as Cl_2, a feature vector $\mathbf{p}^{Cl_2}$ is constructed using the same partitioning as in Cl_1.

The next step is to classify the data in the constructed low-dimensional feature space. In this respect, there are many options for selecting classifiers that could either be parametric or non-parametric. Among the parametric classifiers, one of the commonly used techniques relies on the second-order statistics in the feature space, where the mean feature is calculated for every class along with the variance of the feature space distribution in each class of the training set. Then, a test feature vector is classified by using the Mahalnobis distance [17] or the Bhattacharya distance [18] of the test vector from the mean feature vector of each class. However, these methods are not efficient for non-Gaussian distributions, where the feature space distributions may not be adequately described by the second order statistics. Consequently, a non-parametric classifier (e.g., k-NN classifier [19]) is potentially a better choice. In this study, Gaussian probability distribution may not be assured in the feature space due to the nonlinear partitioning process and therefore, k-NN classifier has been chosen. However, in general, any other suitable classifier such as the support vector machines (SVM) or the Gaussian Mixture Models (GMM) may also be used [19]. To classify the test data set, the time series sets are converted into feature vectors using the same partitioning that has been used to generate the training features. Then, using the labeled training features, the test features are classified by a k-NN classifier with suitable specifications (e.g., neighborhood size and distance metric).

3. Optimization of partitioning

In the literature of multi-class classification, many optimization criteria can be found for optimal feature extraction. However, the primary objective across all the criteria is minimization of classification error. In this context, an ideal objective function may be described in terms of the classification confusion matrix [20]. In pattern recognition literature, a confusion matrix is used to visualize the performance of a classification process, where each column represents the instances in a class predicted by the classifier, while each row represents the instances in an actual class. Formally, in a classification problem with n classes, $Cl_1, \ldots, Cl_n$, the ijth element c_{ij} of confusion matrix $\mathbf{C}$ denotes the frequency of data from class Cl_i being classified as data from Cl_j. Therefore, ideally one should jointly minimize every off-diagonal element and maximize every diagonal element of the confusion matrix. However, in that case, the dimension of the objective space may rapidly increase with an increase in the number of classes. To circumvent this situation in the present work, two costs are defined on the confusion matrix by using another weighting matrix, elements of which denote the relative penalty values for different confusions in the classification process.

Let there be $Cl_1, \ldots, Cl_n$ classes of labeled time-series data in the training set. A partitioning $\mathbb{B}$ is employed to extract features from each sample and a k-NN classifier $\mathbb{K}$ is used as a classifier. The confusion matrix $\mathbf{C}$ is obtained upon completion of the classification process. Let $\mathbf{W}$ be the weighting matrix, where the *ij*th element w_{ij} of $\mathbf{W}$ denotes the penalty incurred for classifying data from Cl_i as data from class Cl_j. (Note: since there is no penalty for correct classification, the diagonal elements of $\mathbf{W}$ are identically equal to 0, i.e., $w_{ii}=0\ \forall i$.) With these specifications, two costs, *CostE* and *CostW*, that are to be minimized are defined as follows.

The cost *CostE* due to expected classification error is defined as

$$CostE = \frac{1}{N_s}\left(\sum_i \sum_j w_{ij} c_{ij}\right) \tag{1}$$

where N_s is the total number of training samples including all classes. The above equation represents the total penalty for misclassification across all classes. Thus *CostE* is related to the expected classification error. The weights w_{ij} are selected based on the domain knowledge and user requirements (e.g., trade-off between false alarm and missed detection [21]). In many fault detection problems, missed detections are more risky compared to false alarms. Accordingly, the weights for missed detection (false negative) should be chosen to be larger compared to those for false alarms (false positive).

It is implicitly assumed in many supervised learning algorithms [22] that the training data set is a statistically similar representation of the whole data set. However, this assumption may not be accurate in practice. A solution to this problem is to choose a feature extractor that minimizes the worst-case classification error. In this setting, that cost *CostW* due to worst-case classification error is defined as

$$CostW = \max_i \left(\frac{1}{N_i}\sum_j w_{ij} c_{ij}\right) \tag{2}$$

where N_i is the number of training samples in the class Cl_i. (*Note*: in the present formulation, the objective space is two-dimensional for a multi-class classification problem and the dimension is not a function of the number of classes.)

3.1. Sensitivity and robustness

Cost minimization requires a choice of partitioning that could be sensitive to class separation in the data space. However, the enhanced sensitivity of optimal partitioning may cause large classification errors due to noise and spurious disturbances in the data and statistical variations among training and testing data sets. Hence, it is important to invoke robustness properties into the optimal partitioning. In this paper, robustness has been incorporated by modifying the costs *CostE* and *CostW*, introduced in the previous subsection.

For one-dimensional time series data, a partitioning consisting of $|\Sigma|$ cells is represented by $(|\Sigma|-1)$ points that serve as cell boundaries. In the sequel, a Σ-cell partitioning $\mathbb{B}$ is expressed as $\Lambda_{|\Sigma|} \triangleq \{\lambda_1, \lambda_2, \ldots, \lambda_{|\Sigma|-1}\}$, where λ_i denotes a partitioning boundary. Thus, a *Cost* is dependent on the specific partitioning $\Lambda_{|\Sigma|}$ and is denoted by $Cost(\Lambda_{|\Sigma|})$. The key idea of a robust cost for a partitioning $\Lambda_{|\Sigma|}$ is that it is expected to remain invariant under small perturbations of the partitioning points, $\lambda_1, \lambda_2, \ldots, \lambda_{|\Sigma|-1}$ (i.e., fuzzy cell boundaries). To define a robust cost, a distribution of partitioning $f_{\Lambda_{|\Sigma|}}(\cdot)$ is considered, where a sample of the distribution is denoted as $\tilde{\Lambda}_{|\Sigma|} = \{\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_{|\Sigma|-1}\}$ and $\tilde{\lambda}_i$'s are chosen from independent Gaussian distributions with mean λ_i and a uniform standard deviation σ_λ; the choice of σ_λ is discussed later in Section 3.2.

The resulting cost distribution is denoted as $f_{Cost(\Lambda_{|\Sigma|})}(\cdot)$, and $Cost_{robust}$ is defined as the cost value below which 95% of the population of distribution $f_{Cost(\Lambda_{|\Sigma|})}(\cdot)$ remains and is denoted as

$$Cost_{robust} = P_{95}[f_{Cost(\Lambda_{|\Sigma|})}(\cdot)] \tag{3}$$

For the analysis to be presented in Section 3.2, it is assumed that sufficient samples are generated to obtain a good estimate of $Cost_{robust}$ as explained in Fig. 2, where the firm lines denote the boundaries that divide the range space into four cells, namely *a*,*b*,*c* and *d*. However, in general, the cell boundaries are fuzzy in nature, which leads to a probabilistic $Cost_{robust}$ instead of a deterministic cost based on the firm partitioning boundaries. Thus, a robust cost due to expected classification error, $CostE_{robust}$, and a robust cost due to worst-case classification error, $CostW_{robust}$, are defined for a given partitioning $\Lambda_{|\Sigma|}$.

Finally, a multi-objective overall cost *CostO* is defined as a linear convex combination of $CostE_{robust}$ and $CostW_{robust}$ in terms of a scalar parameter $\alpha \in [0,1]$:

$$CostO \triangleq \alpha CostE_{robust} + (1-\alpha) CostW_{robust} \tag{4}$$

Ideally, the optimization procedure involves construction of the Pareto front [23] by minimizing *CostO* for different values of α that can be freely chosen as the operating point. Thus, the optimal Σ-cell partitioning $\mathbb{B}^*$ is the solution to the following optimization problem:

$$\mathbb{B}^* = \arg\min_{\mathbb{B}} CostO(\mathbb{B}) \tag{5}$$

Fig. 3 depicts a general outline of the classification process. Labeled time series data from the training set are partitioned. The low-dimensional feature vectors that are

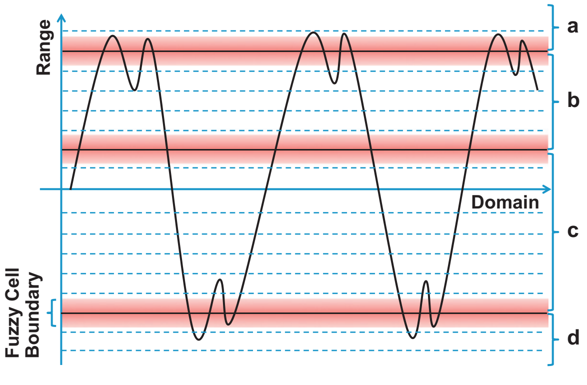

Fig. 2. Fuzzy cell boundaries to obtain $CostE_{robust}$ and $CostW_{robust}$.

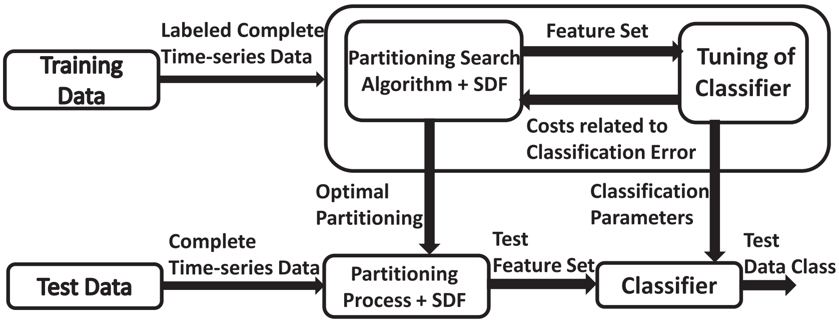
Training Data
Labeled Complete Time-series Data
Partitioning Search Algorithm + SDF
Feature Set
Tuning of Classifier
Costs related to Classification Error
Optimal Partitioning
Classification Parameters
Test Data
Complete Time-series Data
Partitioning Process + SDF
Test Feature Set
Classifier
Test Data Class

Fig. 3. General framework for optimization of feature extraction.

generated by symbolization and *PFSA* construction are fed to the classifier. After classification, the training error costs, defined above, are computed and fed back to the feature extraction block. In the classification aspect, the classifier may be tuned to the obtain better classification rates. For example, for k-NN classifiers [19], the choice of neighborhood size or the distance metric can be tuned. Similarly, for support vector machines [19], an appropriate hyperplane should be selected to achieve good classification. The key idea is to update the partitioning to reduce the cost based on the feedback. The iteration is continued until the set of optimal partitioning in a multi-objective scenario and the correspondingly tuned classifier are obtained. In particular, the iteration can be stopped as the rate of decrease of overall cost fall below a certain threshold. This stopping rule is specified in the following subsection. Generally, the choice of the optimal partitioning is made based on the choice of operating point α by the user. After the choice is made, the optimal partitioning and the tuned classifier are used to classify the test data set. Although a general framework is proposed for optimization, the issue of tuning the classifier is not the main focus of this paper. However, the reported work specifically uses the k-NN classifier. The neighborhood size and distance metric are tuned appropriately depending on the data sets used for validation.

3.2. Optimization procedure

A sequential search-based technique has been adopted in this paper for optimization of the partitioning. As the continuity of the partitioning function with respect to the range space of classification error-related costs may not exist or at least are not adequately analyzed, gradient-based optimization methods are not explored here. To construct the search space, a suitably fine grid size depending on the data characteristics needs to be assumed. Each of the grid boundaries denotes a possible position of a partitioning cell boundary, as illustrated in Fig. 2. Here, the dotted lines denote the possible positions of a partitioning cell boundary and as discussed before, for a chosen partitioning (denoted by firm lines), the partitioning boundaries are perturbed to obtain a $Cost_{robust}$.

Let the data space region Ω be divided into G grid cells for search, i.e., there are $(G-1)$ grid boundaries excluding the boundaries. Thus, there are $|\Sigma|-1$ partitioning boundaries to choose among $(G-1)$ possibilities, i.e., the number of elements (i.e., $(|\Sigma|-1)$-dimensional partitioning vectors) in the space $\mathcal{P}$ of all possible partitioning is ${}^{(G-1)}C_{(|\Sigma|-1)}$. It is clear from this analysis that the partitioning space $\mathcal{P}$ may become significantly large with an increase in values of G and $|\Sigma|$ (e.g., for $G \gg |\Sigma|$, computational complexity increases approximately by a factor of $G/|\Sigma|$ with increase in the value of $|\Sigma|$ by one). Furthermore, for each element of $\mathcal{P}$, a sufficiently large number of perturbed samples need to be collected in order to obtain the $Cost_{robust}$. Therefore, usage of a direct search approach becomes infeasible for evaluation of all possible partitionings. Hence, a sub-optimal solution is developed in this paper to reduce the computational complexity of the optimization problem.

The objective space consists of the scalar-valued cost $CostO$, while decisions are made in the space $\mathcal{P}$ of all possible partitionings. The overall cost is dependent on a specific partitioning Λ and is denoted by $CostO(\Lambda)$. This sub-optimal partitioning scheme involves sequential estimation of the elements of the partitioning Λ.

The partitioning process is initiated by searching the optimal cell boundary to divide the data set into two cells, i.e., $\Lambda_2 = \{\lambda_1\}$, where λ_1 is evaluated as

$$\lambda_1^* = \arg\min_{\lambda_1} CostO(\Lambda_2) \tag{6}$$

Now, the two-cell optimal partitioning is given by $\Lambda_2^* = \{\lambda_1^*\}$.

Note that to obtain $CostO$ (i.e., both $CostE_{robust}$ and $CostW_{robust}$) for a given partitioning, a suitable σ_λ needs to be chosen. Let the gap between two search grid boundaries (i.e., two consecutive dotted lines in Fig. 2) be l_λ, and σ_λ is chosen as $l_\lambda/3$ in this paper. The choice of such a standard deviation of the Gaussian perturbation is made for approximately complete coverage of the search space. Note that there could be an overlap of perturbation regions between two consecutive search grid boundaries, which leads to a smoother (that is essentially robust) cost variation across the domain space. This issue will be further discussed in Section 4.2. In addition, depending on the gap l_λ and data characteristics, a suitable sample size is chosen to approximate the cost distribution under the fuzzy cell boundary condition.

The next step is to partition the data into three cells as Λ_3 by dividing either of the two existing cells of Λ_2^* with the placement of a new partition boundary at λ_2, where λ_2

is evaluated as

$$\lambda_2^* = \arg\min_{\lambda_2} CostO(\Lambda_3) \tag{7}$$

where $\Lambda_3 = \{\lambda_1^*, \lambda_2\}$. The optimal 3-cell partitioning is obtained as $\Lambda_3^* = \{\lambda_1^*, \lambda_2^*\}$. In this (local) optimization procedure, the cell that provides the largest decrement in *CostO* upon further segmentation ends up being partitioned. Iteratively, this procedure is extended to obtain the $|\Sigma|$ cell partitioning as follows:

$$\lambda_{|\Sigma|-1}^* = \arg\min_{\lambda_{|\Sigma|-1}} CostO(\Lambda_{|\Sigma|}) \tag{8}$$

where $\Lambda_{|\Sigma|} = \Lambda_{|\Sigma|-1}^* \cup \{\lambda_{|\Sigma|-1}\}$ and the optimal $|\Sigma|$ cell partitioning is given by $\Lambda_{|\Sigma|}^* = \Lambda_{|\Sigma|-1}^* \cup \{\lambda_{|\Sigma|-1}^*\}$.

In this optimization procedure, the cost function decreases monotonically with every additional sequential operation, under the assumption of correct estimation of $CostE_{robust}$, $CostW_{robust}$ and hence, *CostO* under the fuzzy cell boundary condition. Formally, $CostO(\Lambda_{|\Sigma|-1}^*) \geq CostO(\Lambda_{|\Sigma|}^*)$ as explained below.

Let $\Lambda_{|\Sigma|-1}^*$ be the $(|\Sigma|-1)$-cell partitioning that minimizes *CostO*, based on the algorithm, $\Lambda_{|\Sigma|} = \Lambda_{|\Sigma|-1}^* \cup \{\lambda_{|\Sigma|-1}\}$. If $\lambda_{|\Sigma|-1}$ is chosen such that it already belongs to $\Lambda_{|\Sigma|-1}^*$, then there would be no change in the partitioning structure, i.e.,

$$CostO(\Lambda_{|\Sigma|}) = CostO(\Lambda_{|\Sigma|-1}^*) \quad \text{for } \lambda_{|\Sigma|-1} \in \Lambda_{|\Sigma|-1}^* \tag{9}$$

If $\lambda_{|\Sigma|-1} \in \Lambda_{|\Sigma|-1}^*$ then the partitioning $\Lambda_{|\Sigma|}$ is essentially treated as a $(|\Sigma|-1)$-cell partitioning for the purpose of cost calculation. By definition,

$$CostO(\Lambda_{|\Sigma|}^*) \leq CostO(\Lambda_{|\Sigma|}) \quad \forall \Lambda_{|\Sigma|} \tag{10}$$

Then it follows that

$$\min\,(CostO(\Lambda_{|\Sigma|-1})) \geq \min(CostO(\Lambda_{|\Sigma|})) \tag{11}$$

or

$$CostO(\Lambda_{|\Sigma|-1}^*) \geq CostO(\Lambda_{|\Sigma|}^*) \tag{12}$$

The monotonicity in the cost function allows formulation of a rule for termination of the sequential optimization algorithm. The process of creating additional partitioning cells is stopped if the cost decrease falls below a specified positive scalar threshold η_{stop} and the stopping rule is as follows.

$\Lambda_{|\Sigma|-1}^*$ is the optimal partitioning (and $|\Sigma|-1$ is the optimal alphabet size) if

$$CostO(\Lambda_{|\Sigma|-1}^*) - CostO(\Lambda_{|\Sigma|}^*) \leq \eta_{stop} \tag{13}$$

In contrast to the direct search of the entire space of partitioning, the computational complexity of this approach increases linearly with $|\Sigma|$. This approach also allows the user to have finer grid size for the partitioning search.

4. Validation example 1: parameter identification

This section describes the first example and the associated results to validate the merits of the proposed technique. The problem of parameter identification in the nonlinear Duffing system that is posed as a multi-class classification problem in Section 4.1 and Section 4.2 presents the classification results along with relevant discussions.

4.1. Problem description

The exogenously excited Duffing system [15] is nonlinear with chaotic properties and its governing equation is

$$\frac{d^2y(t)}{dt^2} + \beta\frac{dy}{dt} + \alpha_1 y(t) + y^3(t) = A\cos(\Omega t) \tag{14}$$

where the amplitude $A=22.0$, excitation frequency $\Omega = 5.0$, and reference values of the remaining parameters, to be identified, are $\alpha_1 = 1.0$ and $\beta = 0.1$. It is known that this system goes through a bifurcation at different combinations of α_1 and β, which can be identified by standard feature extraction procedures [19]. The problem at hand is to accurately identify the ranges of the parameters α_1 and β when the system has not undergone any bifurcation. In this paper, multiple classes are defined based on the combination of approximate ranges of the parameters α_1 and β as described below.

Parameter	**Values of α_1**
Range 1	0.800–0.934
Range 2	0.934–1.067
Range 3	1.067–1.200
Parameter	**Values of β**
Range 1	0.100–0.147
Range 2	0.147–0.194
Range 3	0.194–0.240

In this study, classes are defined as Cartesian products of the ranges of α_1 and β. Thus, there are nine (i.e., 3 × 3) classes of data, where a class is uniquely defined by a range of α_1 and a range of β. Two hundred simulation runs of the Duffing system have been conducted for each class to generate data set for analysis among which 100 samples are chosen as the training set and the remaining 100 samples are kept as testing set. Parameters α_1 and β are chosen randomly from independent uniform distributions for both parameters within the prescribed ranges given in above table. Fig. 4 plots the samples generated using the above logic in the two-dimensional parameter space. Different classes of samples are shown in different colors and as well as marked with the class numbers in the figure. For each sample point in the parameter space, time series has been collected for *State y*, the length of the simulation time window being 80 s sampled at 100 Hz, which generates 8000 data points. Fig. 5 exhibits typical phase plots of the Duffing system from each of the nine classes. The following section presents the classification performance of the optimal partitioning along with a comparison with that of the classical partitioning schemes.

4.2. Results and discussion

A weighting matrix **W** needs to be defined to calculate the classification error related costs as discussed in Section 3. In the present case, **W** is defined according to the adjacency properties of classes in the parameter space. That means $w_{ii} = 0$, $\forall i \in \{1,2,\ldots,9\}$, i.e., there is

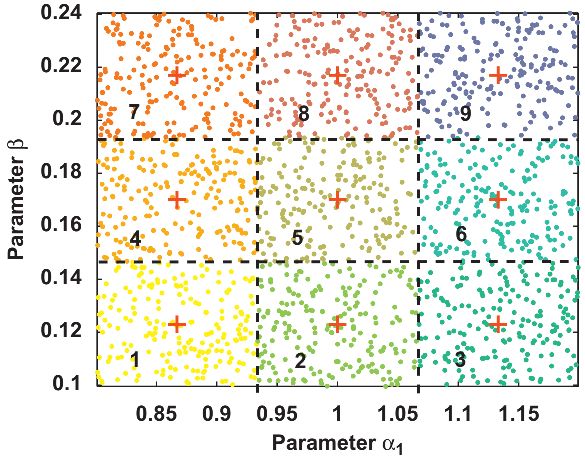

Fig. 4. Parameter space with class labels. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

no penalty when Cl_i is classified as Cl_i and in general $w_{ij} = |R_{\alpha_1}(i) - R_{\alpha_1}(j)| + |R_\beta(i) - R_\beta(j)|$, $\forall i \in \{1, 2, \ldots, 9\}$, where $R_\gamma(k)$ denotes the range number (see Section 4.1) for parameter γ (in this case, α_1 or β) in class k. In this context, **W** is written as

$$\mathbf{W} = \begin{pmatrix} 0 & 1 & 2 & 1 & 2 & 3 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 3 \\ 2 & 1 & 0 & 3 & 2 & 1 & 4 & 3 & 2 \\ 1 & 2 & 3 & 0 & 1 & 2 & 1 & 2 & 3 \\ 2 & 1 & 2 & 1 & 0 & 1 & 2 & 1 & 2 \\ 3 & 2 & 1 & 2 & 1 & 0 & 3 & 2 & 1 \\ 2 & 3 & 4 & 1 & 2 & 3 & 0 & 1 & 2 \\ 3 & 2 & 3 & 2 & 1 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 3 & 2 & 1 & 2 & 1 & 0 \end{pmatrix}$$

The data space region Ω is divided into 40 grid cells, i.e., 39 grid boundaries excluding the boundaries of Ω. The sequential partitioning optimization procedure described in Section 3.2 is then employed to identify the optimal partitioning. The parameter α is taken to be 0.5 in this example, i.e., equal weights for the costs $CostE_{robust}$ and $CostW_{robust}$. Fig. 6 depicts the optimization process for obtaining the optimal partitioning, where λ_1^* is evaluated by minimizing $CostO(\Lambda_2)$. Both the cost curve and its corresponding optimal value λ_1^* are shown in Fig. 6. Similarly, the second optimal partitioning boundary λ_2^* is obtained by minimizing the cost function $CostO(\Lambda_3) \triangleq CostO(\{\lambda_1^*, \lambda_2\})$. As described in Section 3.2, λ_1^* is kept fixed while λ_2 is optimized. This suboptimal process is recursively continued until the threshold $\eta_{stop} = 0.01$ is reached, which leads to the creation of six cells (i.e., five partitions) denoted by $\Lambda_6^* = \{\lambda_1^*, \ldots, \lambda_5^*\}$ as shown in Fig. 6. For SDF analysis, the depth for constructing *PFSA* states is taken to be $D = 1$ and features are classified by a k-NN classifier (with $k = 5$) using the Euclidean distance metric. Also, for estimation of $CostE_{robust}$ and $CostW_{robust}$, σ_λ is taken to be $l_\lambda = 0.0333$ and 50 perturbed samples are taken for each partitioning elements in the search space. Choice of such σ_λ leads to smooth cost curves across the *State y* values (domain space) as seen in Fig. 6.

Fig. 7 plots the optimal partitioning Λ_6^* on a representative time-series from the reference class 5. Finally, the decrease in $CostE_{robust}$ and $CostW_{robust}$ with the increase in alphabet size is shown in Fig. 8. The optimal alphabet size and corresponding cost values are marked on each plate. The confusion matrix obtained by using the optimal partitioning (OptP) on the test data set is given below:

$$\mathbf{C}_{test}^{OptP} = \begin{pmatrix} 98 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 92 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 96 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 88 & 5 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 84 & 12 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 94 & 0 & 0 & 3 \\ 0 & 0 & 0 & 4 & 0 & 0 & 92 & 4 & 0 \\ 0 & 0 & 0 & 1 & 4 & 0 & 8 & 83 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13 & 87 \end{pmatrix}$$

It is observed that the class separability is retained in an efficient way by the nonlinear feature extraction (partitioning) process even after compressing a time series data (with 8000 data points) into a six-dimensional feature (state probability) vector. The confusion matrices for uniform and maximum entropy partitioning on the test data set are also provided below for comparison:

$$\mathbf{C}_{test}^{UP} = \begin{pmatrix} 84 & 10 & 5 & 1 & 0 & 0 & 0 & 0 & 0 \\ 7 & 87 & 4 & 1 & 1 & 0 & 0 & 0 & 0 \\ 14 & 3 & 77 & 1 & 5 & 0 & 0 & 0 & 0 \\ 10 & 1 & 2 & 76 & 11 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 16 & 79 & 3 & 0 & 1 & 0 \\ 0 & 0 & 3 & 3 & 3 & 84 & 0 & 2 & 5 \\ 0 & 0 & 0 & 2 & 2 & 0 & 88 & 8 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 13 & 76 & 8 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 11 & 85 \end{pmatrix}$$

$$\mathbf{C}_{test}^{MEP} = \begin{pmatrix} 83 & 12 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 13 & 82 & 3 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 5 & 87 & 1 & 5 & 0 & 0 & 0 & 0 \\ 1 & 1 & 4 & 85 & 3 & 6 & 0 & 0 & 0 \\ 0 & 2 & 0 & 9 & 84 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 81 & 0 & 4 & 5 \\ 0 & 0 & 0 & 2 & 0 & 1 & 86 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8 & 11 & 74 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 10 & 87 \end{pmatrix}$$

Table 1 presents the comparison of the classification error related costs for OptP, UP and MEP on the test data set.

The observations made from these results indicate that the classification performance may be improved compared to that of the classical partitioning schemes by optimizing the partitioning process over a representative training set for the particular problem at hand. However, it should be noted that for some problems, the classical partitioning schemes may perform as well as the optimal one. Therefore, the optimization procedure may also be used to evaluate the capability of any partitioning scheme toward achieving a better classification rate. The evaluation can be performed by

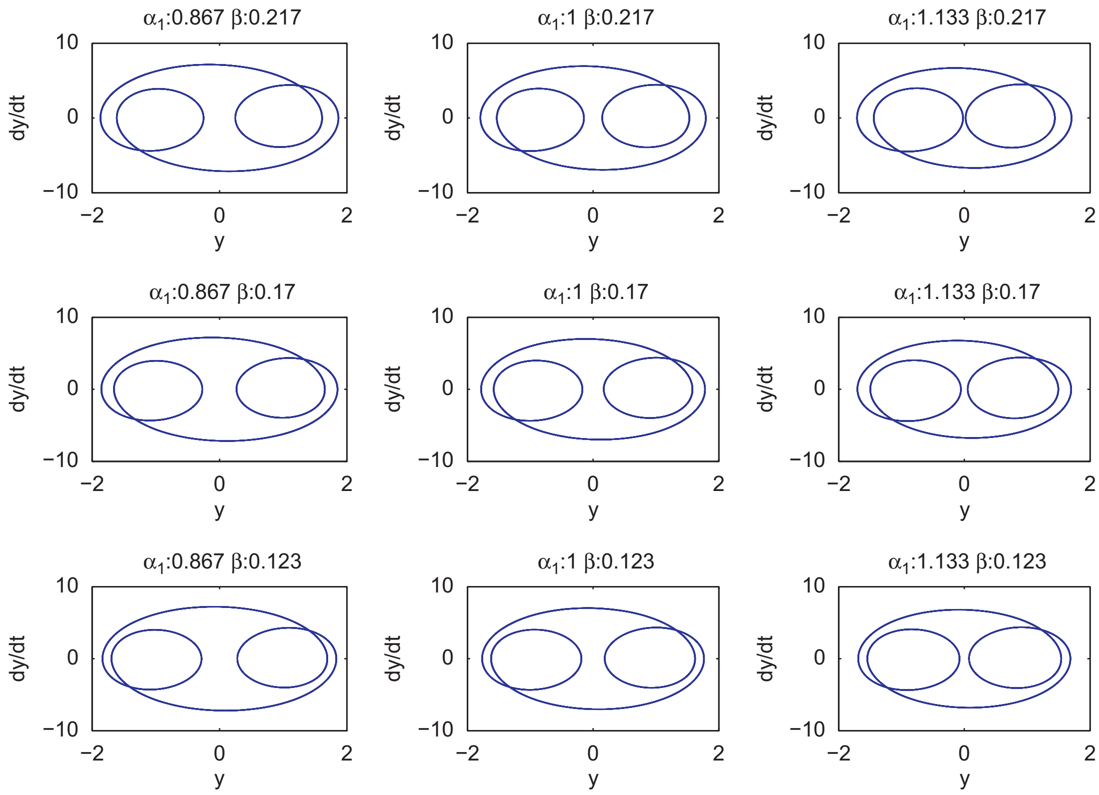

Fig. 5. Representative phase space plots for different classes.

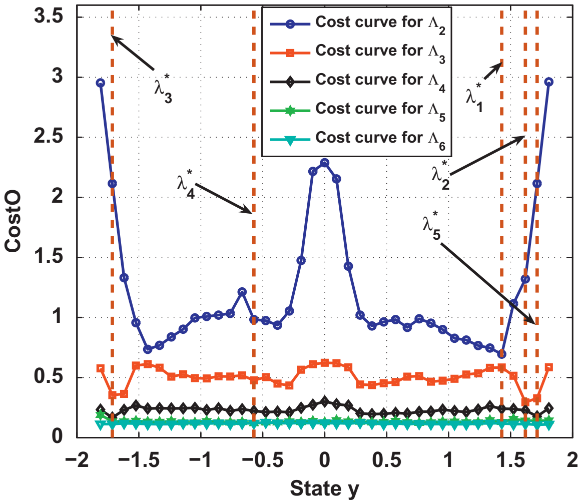

Fig. 6. Cost curves and optimal partitioning boundaries for different alphabet sizes obtained during the sequential optimization procedure.

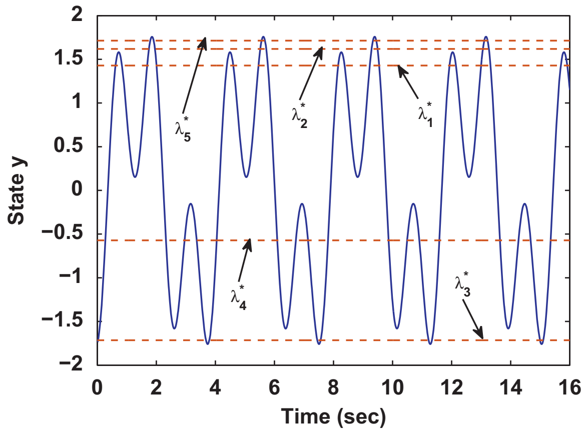

Fig. 7. Optimal partitioning marked on the data space with a representative time-series from Class 5.

using a part of the labeled training data set as the validation set. Finally, although the construction of the cost functions theoretically allow problems with large number of classes, in practice it should be understood that its upper limit will be constrained by the alphabet size used for partitioning which is also the dimension of the feature space. Also note that the model complexity of a probabilistic finite state automaton (*PSFA*), as obtained from time series data, is related to the number of states (hence, to the number of partitions) in the *PSFA*. Therefore, the choice of η_{stop} is critical in our approach during the process of partitioning optimization to alleviates the issue of over-training [12].

5. Validation example 2: damage classification

Fatigue damage is one of the most commonly encountered sources of structural degradation of mechanical

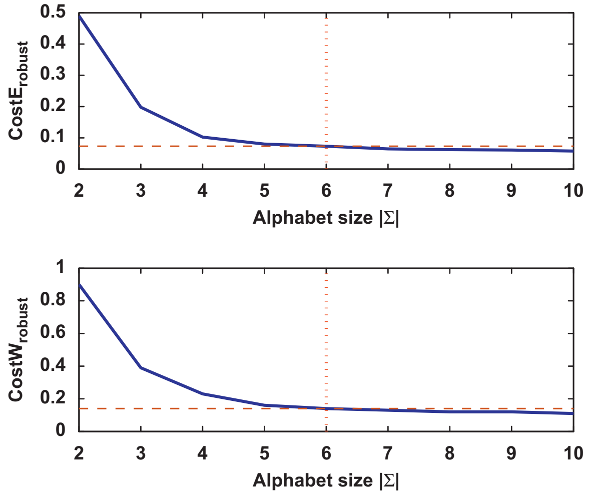

Fig. 8. Decrease in *CostE* and *CostW* with increase in alphabet size, for optimal partitioning $|\Sigma|$ is chosen to be 6.

Table 1
Comparison of classification performances of partitioning schemes on test-data set (100×9 samples).

Partitioning	*CostE*	*CostW*
OptP	0.0978	0.1800
UP	0.2289	0.4400
MEP	0.2200	0.3400

structures, made of polycrystalline alloys. Therefore, analytical tools for online fatigue damage detection are critical for a wide range of engineering applications. Recently, symbolic time series analysis (STSA) [16] has been proposed and successfully demonstrated on ultrasonic time series data for early detection of evolving fatigue damage. This section uses the concepts of optimal partitioning for feature extraction from the ultrasonic signals collected from an experimental apparatus, described in Section 5.1, for quantification of fatigue damage.

The process of fatigue damage is broadly classified into two phases: crack initiation and crack propagation. The damage mechanism of these two phases are significantly different and similar feature extraction policies may not work effectively to classify different damage levels in these two phases. For example, damage evolution in the crack initiation phase is much slower, resulting in smaller change in ultrasonic signals as compared to that in the crack propagation phase. The phase transition from crack initiation to crack propagation occurs when several small micro-cracks coalesce together to develop a single large crack that propagates under the oscillating load.

This section focuses on damage level identification in the crack propagation phase. Several crack propagation models have been developed based on the inherent stochastic nature of fatigue damage evolution for prediction of the remaining useful life. Due to stochastic nature of material microstructures and operating conditions, a physics-based model would require the knowledge of the parameters associated with the material and geometry of the component. These parameters are often randomly varying and may not be accurately predicted a priori. Crack propagation rate is a function of crack length and, after a certain crack length, it becomes unstable. The crack propagation stage is divided into four classes as presented below.

Class	Crack length
1	0.5–1.75 mm
2	1.75–3.5 mm
3	3.5–5.5 mm
4	More than 5.5 mm

As described earlier, the weighing matrix **W** is chosen based on the user's requirement. In the present case, **W** is defined from the perspective of risk involved in misclassification. For example, penalty is low when data in Class 1 (i.e., small crack length) is classified as in classes for larger crack length; however, penalty is high for the converse. For the results presented in this paper, **W** matrix is defined as follows:

$$\mathbf{W}=\begin{pmatrix}0&1&2&3\\3&0&1&2\\6&3&0&1\\9&6&3&0\end{pmatrix}$$

In the present work, a large volume of ultrasonic data has been collected for different crack lengths on different specimens to demonstrate the efficiency of the present classification scheme to account for the stochastic nature of material properties. The following subsection presents the details of the experimental apparatus used for damage detection using ultrasonic sensors. It also describes the test specimens and the test procedure.

5.1. Experimental apparatus and test procedure

The experimental apparatus is designed to study the fatigue damage growth in mechanical structures. The apparatus consists of an MTS 831.10 Elastomer Test System that is integrated with Olympus BX Series microscope with a long working-distance objective. A camera, mounted on the microscope, takes images with a resolution of 2 μ per pixel at a distance of 20 mm. The ultrasonic sensing device is triggered at a frequency of 5 MHz at each peak of the fluctuating load. Various components of the apparatus communicate over a TCP/IP network and post sensor, microscope and fatigue test data on the network in real time. This data can used by analysis algorithms for anomaly detection and health monitoring of the specimens in real-time.

A side-notched 7075-T6 aluminum alloy specimen has been used in the fatigue damage test apparatus. Each specimen is 3 mm thick and 50 mm wide, and has a slot of 1.58 mm × 4.57 mm on one edge. The notch is made to increase the stress concentration factor that localizes crack initiation and propagation under the fluctuating load. Fatigue tests were conducted at a constant amplitude sinusoidal load for low-cycle fatigue, where the maximum and minimum loads were kept constant at 87 MPa and 4.85 MPa, respectively. For low cycle fatigue studied in this paper, the stress amplitude at the crack tip is sufficiently high to observe the elasto-plastic behavior in the specimens under cyclic loading. A significant amount of internal damage caused by multiple small cracks, dislocations and microstructural defects alters the ultrasonic impedance, which results in signal distortion and attenuation at the receiver end.

The optical images were collected automatically at every 200 cycles by the optical microscope which is always focussed in the crack tip. As soon as crack is visible by the microscope, crack length is noted down after every 200 cycles. Ultrasonic waves with a frequency of 5 MHz were triggered at each peak of the sinusoidal load to generate data points in each cycle. Since the ultrasonic frequency is much higher than the load frequency, data acquisition was done for a very short interval in the time scale of load cycling. Therefore, it can be implied that ultrasonic data were collected at the peak of each sinusoidal load cycle, where the stress is maximum and the crack is open causing maximum attenuation of the ultrasonic waves. The slow time epochs for data analysis were chosen to be 1000 load cycles (i.e., ~ 80 s) apart. To generate training and test data sample multiple experiments are conducted on different specimens. For each specimen all ultrasonic signals are labeled with crack length.

5.2. Results and discussion

This section presents the damage level classification results for the crack propagation phase using the optimal partitioning along with results obtained by using maximum entropy and uniform partitioning [12]. The classification process is started by wavelet transformation of the time series data with suitable scales and time shifts for a given basis function. Each transformed signal is normalized with the maximum amplitude of transformed signal obtained at the beginning of experiment, when there is no damage in the specimen. The data are normalized to mitigate the effects of variability in the placement of ultrasonic sensors during different experiments.

As described earlier, both training and test data sets are divided into four classes based on crack length as all the ultrasonic signals are labeled with the corresponding crack lengths. The sequential optimization of partitioning has been carried out on the training data set to find optimal partitioning. The parameter α is taken to be 0.5 in this example, i.e., equal weights for the costs $CostE_{robust}$ and $CostW_{robust}$. The specifications of SDF and the classifier remain same as in the first example. The optimal alphabet size is 6 with stopping threshold value $\eta_{stop} = 0.005$. The confusion matrices for optimal, uniform and maximum entropy partitioning on the test data set are given by $\mathbf{C}_{test}^{OptP}$ $\mathbf{C}_{test}^{UP}$ and $\mathbf{C}_{test}^{MEP}$, respectively. Table 2 shows the comparison of classification performances using different partitioning processes:

$$\mathbf{C}_{test}^{OptP} = \begin{pmatrix} 93 & 7 & 0 & 0 \\ 5 & 89 & 6 & 0 \\ 0 & 4 & 92 & 4 \\ 0 & 3 & 7 & 90 \end{pmatrix}$$

Table 2
Performance comparison of partitioning schemes.

Partitioning	$Cost_E$	$Cost_W$
OptP	0.255	0.5
UP	0.40333	0.68
MEP	0.31333	0.52

$$\mathbf{C}_{test}^{UP} = \begin{pmatrix} 94 & 5 & 1 & 0 \\ 15 & 74 & 11 & 0 \\ 0 & 9 & 85 & 6 \\ 1 & 5 & 11 & 83 \end{pmatrix}$$

$$\mathbf{C}_{test}^{MEP} = \begin{pmatrix} 93 & 7 & 0 & 0 \\ 12 & 82 & 6 & 0 \\ 0 & 7 & 89 & 4 \\ 1 & 3 & 7 & 89 \end{pmatrix}$$

It is observed that both costs $Cost_E$ and $Cost_W$ are reduced for optimal partitioning as compared to maximum entropy and uniform partitioning. It is also evident from the confusion matrix that optimal partitioning has improved the classification results. A close observation of confusion matrix indicates that chances of higher damage level data samples being classified as a lower level damage is reduced.

6. Summary, conclusions and future work

This article presents symbolic feature extraction from time-series of observed sensor data. The feature extraction algorithm maximizes the classification performance with a trade-off between sensitivity and robustness, where the time series is optimally partitioned in the symbolic dynamic filtering (SDF) framework. It is demonstrated that the classification performance is improved beyond what is achieved using the conventional partitioning techniques (e.g., maximum entropy partitioning and uniform partitioning). The optimization procedure can also be used to evaluate the capability of other partitioning schemes toward achieving a particular objective. Nevertheless, efficacy of the proposed partitioning optimization process depends on the very nature of the time series.

Often class separability among time series data sets is more conducive in the frequency or time-frequency domain than in the time domain. Thus, identifying suitable data pre-processing methods from the training data set is an important aspect, which is a topic of future investigation. Apart from this issue, the following research topics are currently being pursued as well.

- Use of other classifiers (e.g., support vector machines [19]) and performance comparison among different classifiers.
- Tuning of the classifier within the optimization loop as described in the general framework in Fig. 3.
- Development of an adaptive feature extraction framework for optimal partitioning under different environmental and signal conditions.
- Validation of the proposed algorithm in other applications of pattern classification.

References

[1] K. Fukunaga, Statistical Pattern Recognition, 2nd ed., Academic Press, Boston, USA, 1990.

[2] T. Lee, Independent Component Analysis: Theory and Applications, Kluwer Academic Publishers, Boston, USA, 1998.

[3] R. Rosipal, M. Girolami, L. Trejo, Kernel PCA feature extraction of event-related potentials for human signal detection performance, Proc. Int. Conf. Artificial Neural Networks Medicine Biol. (2000) 321–326.

[4] K. Weinberger, L. Saul, Unsupervised learning of image manifolds by semidefinite programming, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04), Washington, DC.

[5] J. Tenenbaum, V.d. Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.

[6] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.

[7] A. Ray, Symbolic dynamic analysis of complex systems for anomaly detection, Signal Processing 84 (7) (2004) 1115–1130.

[8] X. Jin, S. Gupta, K. Mukherjee, A. Ray, Wavelet-based feature extraction using probabilistic finite state automata for pattern classification, Pattern Recognition 44 (7) (2011) 1343–1356.

[9] C. Rao, A. Ray, S. Sarkar, M. Yasar, Review and comparative evaluation of symbolic dynamic filtering for detection of anomaly patterns, Signal, Image and Video Processing 3 (2) (2009) 101–114.

[10] R. Steuer, L. Molgedey, W. Ebeling, M. Jimenez-Montano, Entropy and optimal partition for data analysis, The European Physical Journal B 19 (2001) 265–269.

[11] M. Buhl, M. Kennel, Statistically relaxing to generating partitions for observed time-series data, Physical Review E (2005) 046213.

[12] V. Rajagopalan, A. Ray, Symbolic time series analysis via wavelet-based partitioning, Signal Processing 86 (11) (2006) 3309–3320.

[13] A. Subbu, A. Ray, Space partitioning via Hilbert transform for symbolic time series analysis, Applied Physics Letters 92 (8) (2008) 084107-1–084107-3.

[14] S. Sarkar, K. Mukherjee, A. Ray, Generalization of Hilbert transform for symbolic analysis of noisy signals, Signal Processing 89 (6) (2009) 1245–1251.

[15] J. Thompson, H. Stewart, Nonlinear Dynamics and Chaos, Wiley, Chichester, UK, 1986.

[16] S. Gupta, A. Ray, E. Keller, Symbolic time series analysis of ultrasonic data for early detection of fatigue damage, Mechanical Systems and Signal Processing 21 (2) (2007) 866–884.

[17] G.J. Mclachlan, Discriminant Analysis and Statistical Pattern Recognition (Wiley Series in Probability and Statistics), Wiley-Interscience, 2004.

[18] E. Choi, C. Lee, Feature extraction based on the Bhattacharyya distance, Pattern Recognition 36 (2003) 1703–1709.

[19] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[20] A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms, Springer-Verlag, Berlin-Heidelberg, Germany, 2002.

[21] H.V. Poor, An Introduction to Signal Detection and Estimation, 2nd ed., Springer-Verlag, New York, NY, USA, 1994.

[22] R. Alaiz-Rodríguez, A. Guerrero-Curieses, J. Cid-Sueiro, Minimax classifiers based on neural networks, Pattern Recognition 38 (1) (2005) 29–39.

[23] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic, Boston, MA, USA, 1998.