OPTIMIZATION OF TIME-SERIES DATA PARTITIONING FOR PARAMETER IDENTIFICATION IN NONLINEAR DYNAMICAL SYSTEMS

Soumik Sarkar
Department of Mechanical Engineering
The Pennsylvania State University
University Park, PA 16802, USA
Email: szs200@psu.edu

Kushal Mukherjee
Department of Mechanical Engineering
The Pennsylvania State University
University Park, PA 16802, USA
Email: kum162@psu.edu

Xin Jin
Department of Mechanical Engineering
The Pennsylvania State University
University Park, PA 16802, USA
Email: xuj103@psu.edu

Asok Ray*
Department of Mechanical Engineering
The Pennsylvania State University
University Park, PA 16802, USA
Email: axr2@psu.edu

ABSTRACT

The concepts of symbolic dynamics and data set partitioning have been used for feature extraction and classification of time series data. Although modeling of state machines from symbol sequences has been widely reported, similar efforts have not been expended to investigate partitioning of time series data to optimally generate symbol sequences for classification.The paper proposes a partitioning procedure to optimally extract features from time series data and enhance classification accuracy. A multi-objective cost function is constructed for optimization to handle multi-class classification problems in general. Performance comparison of the optimal partitioning is done with the other traditional partitioning schemes, e.g. uniform and maximum entropy, etc. The multi-class classification is used here to identify ranges of multiple parameters of a well-known chaotic nonlinear dynamical system, namely the Duffing Equation.

INTRODUCTION

Accuracy in parameter identification for a dynamical system is crucial from various perspectives, such as, system identification, fault detection and performance monitoring. In human engineered complex systems, inherent nonlinearity and lack of model reliability often make the problem of parameter identification extremely challenging. Therefore, data-driven parameter identification in nonlinear dynamical systems is of paramount importance. In general, data-driven techniques either use snapshots of data from multiple sensors or sensor observation over a certain time window. While the use of snapshots reduces the data volume and computational expense, it fails to capture the statistical characteristics in the data (especially in noisy environment). The resulting misclassification of parameter range of the system can be often avoided by using a window of time series data. The problem with handling time series data is its volume and the associated computational complexity. Unless the data is compressed intelligently into low dimensional features, it is almost impractical to use any pattern matching algorithm. In general, feature extraction is considered as the process of transforming high dimensional data into a low dimensional feature space with minimal loss of class separability. To this end, several tools of feature extraction tools, such as principal component analysis (PCA) [1], independent component analysis (ICA) [2], kernel PCA [3], and semidefinite embedding [4], have been reported in literature. Recent literature [5] has developed a data-driven tool for nonlinear feature extraction, namely the Symbolic Dynamic Filtering (SDF) that is built upon the concepts of symbolic dynamics. The feature extraction procedure is shown to be particularly useful for time-series data that involves partitioning of the data space to generate symbol sequences. Single and

*Address all correspondence to this author.

multi-Parameter estimation algorithms, based on symbolic dynamic filtering (*SDF*), have been developed and experimentally validated for real-time execution in different applications, such as degradation monitoring in electronic circuits [6] [7], fatigue damage monitoring in polycrystalline alloys [8], etc.

Properties of various transformations from symbol space to feature space have been thoroughly studied in mathematics, computer science and especially data mining literature. However, similar efforts have not been expended to investigate partitioning of time series data to optimally generate symbol sequences for classification. Stauer et al. [9] reported a comparison of the maximum entropy partitioning (MEP) and the uniform partitioning (UP) schemes; 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) [10] optimizes a generating partition by avoiding topological degeneracy. However, a shortcoming of SFNNP is that it 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 [11]. Use of wavelet transform [12] and Hilbert transform [11, 13] before partitioning largely alleviates the above limitations. Nevertheless, these partitioning techniques primarily attempts to provide an accurate symbolic representation of the underlying dynamical system under a given quasistationary condition, rather than trying to capture the data-evolution characteristics. This paper tries to overcome this difficulty of the above mentioned traditional partitioning methods to make SDF, a robust data-driven feature extraction tool to enhance classification rate [14]. To this end, the problem of multiple parameter identification in nonlinear dynamical systems is formulated as a multi-class classification problem and a framework is presented toward optimization of the partitioning scheme to increase the accuracy of parameter identification. In the subsequent sections the resulting algorithms are developed and validated on a nonlinear Duffing system.

REVIEW AND PERFORMANCE EVALUATION OF SDF

This section presents a brief outline of the Symbolic Dynamic Filtering (SDF) technique for feature extraction from time series data and the performance of classical partitioning schemes for the current problem described below.

Problem Description

The externally excited Duffing system [15] which is a nonlinear system with chaotic properties, is considered here for validation. The system equation is as follows:

$$\frac{d^2y(t)}{dt^2} + \beta\frac{dy}{dt} + \alpha_1 y(t) + y^3(t) = A\cos(\Omega t) \qquad (1)$$

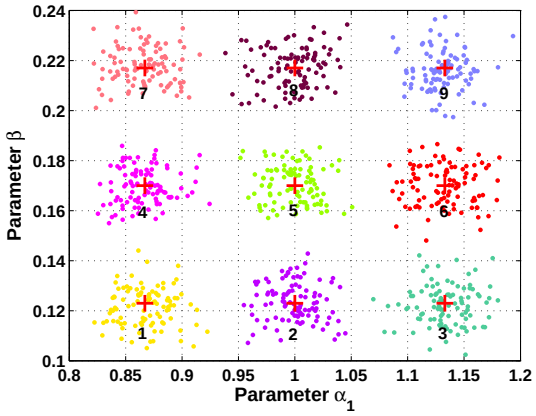

Figure 1. PARAMETER SPACE WITH CLASS LABELS

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 system goes through a bifurcation at different combinations of α_1 and β [7], which can be identified easily by standard feature extraction procedures. However, the challenge remains in accurately identifying α_1 and β parameter ranges when the system has not undergone any bifurcation. In this paper, multiple classes are defined based on the combination of approximate ranges of α_1 and β values as described below.

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

In this study, classes are defined as cartesian products of ranges of α_1 and β. Thus, there are 9 (3×3) classes of data that can be obtained when a class is uniquely defined by a range of α_1 and a range of β. Two hundred simulation runs of the Duffing system are conducted for each class to generate data set for analysis among which 100 samples are chosen as training set and the rest of the 100 samples are kept as testing set. α_1 and β parameters are chosen randomly from independent Gaussian distributions for both parameters, such that almost all the parameter values are within the prescribed ranges given in tables above. In other words, the mean of the Gaussian distribution used for a particular parameter range is taken as the central value of the range and the standard deviation is taken such that the boundary values of the parameter range are 3σ distance away from the central value. Figure 1 plots the training samples generated using the above logic in the two dimensional parametric space. Different classes of samples are shown in different colors and as well as marked with the class number in the figure. For each sam-

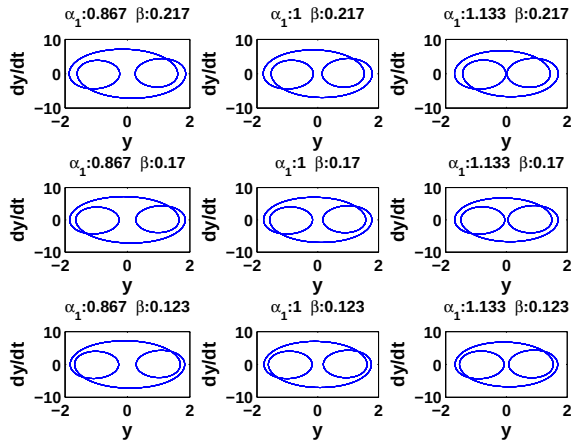

Figure 2. REPRESENTATIVE PHASE SPACE PLOTS FOR DIFFERENT CLASSES

ple point in the parameter space, time series has been collected for state y, the length of the simulation time window being 80 seconds (sampled at 100 Hz), that generates $8,000$ data points. Figure 2 shows representative phase plots of the system from each of the nine classes, where the data sets are generated using the mean parametric values for each class. Although details of SDF methodology can be found in literature, the following section succinctly describes it before evaluating the performances of the classical partitioning schemes.

Partitioning: A Nonlinear Feature Extraction Technique

Symbolic feature extraction from time series data is posed as a two-scale problem. 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 quasi-stationary at the fast scale. On the other hand, 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 underlying system has undergone a change. The method of extracting features from stationary time series data is comprised of the following steps.

1. Let $\Omega \in \mathbb{R}^n$, where $n \in \mathbb{N}$, be a compact (i.e., closed and bounded) region within which the stationary time series (obtained from the dynamical System) is circumscribed. The space of time series data set is represented as $Q \subseteq \mathbb{R}^{n\times N}$, where $N \in \mathbb{N}$ is sufficiently large for convergence of statistical properties within a specified threshold. Then, $\{\mathbf{q}\} \in Q$ denotes a time series at the slow-scale epoch of data collection.
2. Encoding of Ω is accomplished by introducing a partition $\mathbb{B} \equiv \{B_0, ..., B_{(m-1)}\}$ consisting of m mutually exclusive (i.e., $B_j \cap B_k = \emptyset \; \forall j \neq k$), and exhaustive (i.e., $\cup_{j=0}^{m-1} B_j = \Omega$) cells. Let, each cell be labeled by symbols $s_j \in \Sigma$ where $\Sigma = \{s_0, ..., s_{m-1}\}$ is called the alphabet. This process of coarse graining can be executed by uniform, maximum entropy, or any other scheme of partitioning. In fact, subsequent parts of the paper deals with finding an optimal partitioning scheme. The time series data points $\{\mathbf{q}\}$ that visit the cell B_j are denoted as $s_j \; \forall j = 0, 1, ..., m-1$. This step enables transformation of the time series data $\{\mathbf{q}\}$ to a symbol sequence $\{\mathbf{s}\}$.
3. A probabilistic finite state machine (*PFSA*) is then constructed with a chosen depth, and the symbol sequence $\{\mathbf{s}\}$ is run through it. Thus a state transition matrix $\Pi = [\pi_{jk}]$, where $j,k \in \{1,2,...,r\}$ are the states of the *PFSA* with an $(r \times r)$ state transition matrix, is obtained at the slow-scale epoch. Since $\pi_{jk} \geq 0$ is the transition probability from state j to state k, Π is a stochastic matrix, i.e., $\sum_k \pi_{jk} = 1$. To compress the information further, the state probability vector $\mathbf{p} = [p_1 \; \cdots \; p_r]$ 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 relatively low-dimensional representation of the long time series data (dynamical system) at the slow-scale epoch.

The following subsection applies the classical partitioning schemes, e.g., the Uniform and the Maximum Entropy Partitioning to extract such low dimensional features for classification for the problem formulated earlier.

Performance of Classical Partitioning Schemes

In the usual setting of anomaly detection using SDF, the reference time series data space is partitioned using either Uniform Partitioning (UP) or Maximum Entropy partitioning (MEP) scheme In brief, Uniform Partitioning refers to dividing the range of the signal into partition of equal size, while Maximum Entropy partitioning refers to creating a partition that transforms the signal into a symbol sequence with maximum entropy (please see [5, 16] for details of these partitioning schemes). In the present problem of multi-class classification, the partitioning is constructed based on a reference time series chosen from data set for Class 5 (the reference class comprised of ranges α_1=0.934 to 1.067 and β=0.147 to 0.194) using an alphabet size (number of partition cells) m=4. The obtained symbol sequence is then compressed to a *PFSA* with depth equal to 1 (please see [5] for the general logic behind choice of alphabet size and *PFSA* depth). The stationary distribution of the *PSFA*, $\mathbf{p}$ is chosen as the low-dimensional feature vector. Being a probability mass function, the sum of the elements of $\mathbf{p}$ always be equal to 1. Hence, among its 4 elements only 3 will be linearly independent. The partitioning obtained for the reference class 5 is kept the same while analyzing other classes too. Using the same partitioning and structure of the *PFSA*, feature vectors are generated for the training data sets for all classes. Figure 3 and 4 show the location of each

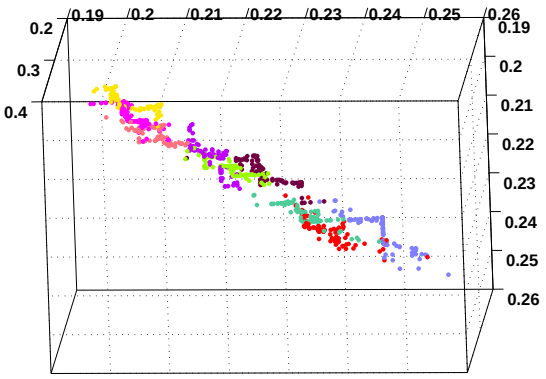

Figure 3. FEATURE SPACE OF TRAINING SET USING UNIFORM PARTITIONING

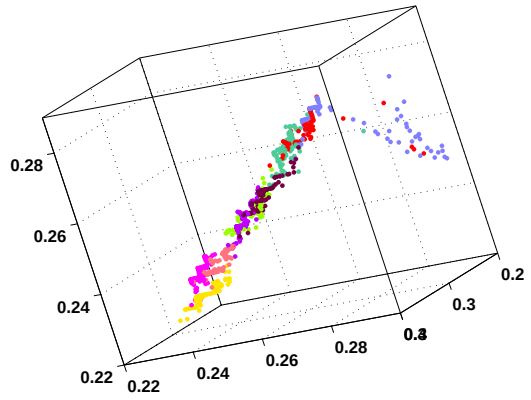

Figure 4. FEATURE SPACE OF TRAINING SET USING MAXIMUM ENTROPY PARTITIONING

training time series in the three dimensional (using first three linearly independent elements of the feature vectors) feature space plot using the Uniform and Maximum Entropy partitioning, respectively.

The next step is to classify the data in the low dimensional feature space. In literature, plenty of choices for parametric and non-parametric classifiers exist. Among the parametric type of classifiers, one of the most common techniques is to consider up to two orders of statistics in the feature space. In other words, mean feature can be calculated for every class along with the variance of the feature space distribution in each of the classes in the training set. Then a test feature vector can be 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 extremely inefficient when the feature space distributions are too complex to be described by second order statistics (i.e., non-Gaussian in nature), which seems to be the case here (see Fig. 3 and Fig. 4). Therefore,

Table 1. CLASSIFICATION RESULTS USING CLASSICAL PARTITIONING SCHEMES

Partitioning Scheme	Classification Error % (Testing Set)
UP	11.33
MEP	14.67

a non-parametric classifier, such as the k-NN classifier may a better candidate for this study. However, in general, any other suitable classifier, such as the Support Vector Machines (SVM) or the Gaussian Mixture Models (GMM) may also be used. To classify the test data set, the time series from the set are converted to feature vectors using the same partitioning and $PFSA$ that have been used to generate the training features. Then using the labeled training features, the test features are classified by a k-NN classifier with $k = 5$ and the Euclidean distance metric (after several trials, the neighborhood size and the distance metric are chosen to obtain good classification rate). The classification results are given in Table 1 for both classical partitioning schemes. The classification error % is defined as the percentage of total number of misclassifications made by the classification process for the test data set. At this point, it should be noted that in the current SDF methodology, partitioning is done based on nominal data. In such cases, even if the partitioning is optimal (e.g., in terms of maximum entropy or some other criteria) under reference conditions, it may not be an optimal feature extraction tool for classification. Hence, it may be advantageous to take non-stationary dynamics into consideration and optimize the partitioning process based on changes of time series data over some the training data sets of different classes. This is the key idea of the current study. The following section outlines the optimization of partitioning methodology and relevant results for the current application.

OPTIMIZATION OF PARTITIONING

In literature, many optimization criteria can be found for feature extraction in a multi-class classification problem. However, none can be more fundamental than minimization of classification error. Although, there have been attempts to approximate the classification error as the Bayes error using pairwise weighting functions for multi-class problems [19], the Bayes error itself cannot be expressed analytically except few special cases. On the other hand, the Fisher criteria is very useful for binary classification problems, especially when the samples are distributed in a Gaussian manner in the feature space. Therefore, for multi-class problems, in [20] a criteria has been proposed based on the minimum classification error (MCE), where the classification error is calculated by the misclassification rate over the training samples. Formally, these two fundamentally different optimization criteria

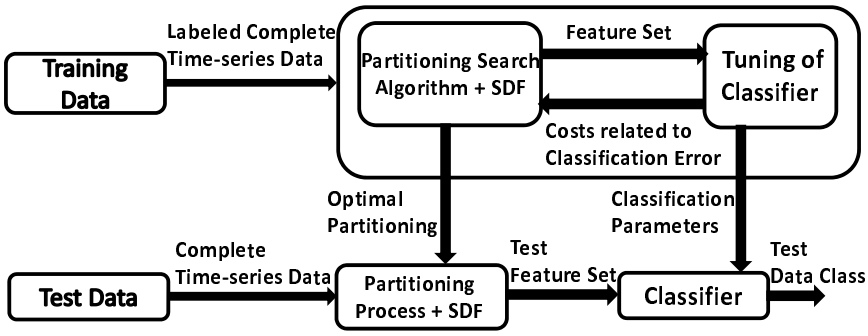

Figure 5. GENERAL FRAMEWORK FOR OPTIMIZATION OF FEATURE EXTRACTION

are known as the i) Filter method and the ii) Wrapper method. The filter methods use information content feedback, e.g. Fisher criteria, Statistical Dependence, Information Theoretic measures etc. as optimization criteria for feature extraction whereas, the wrapper methods include the classifier inside the optimization loop, and try to maximize the predictive accuracy, e.g. classification rate using statistical re-sampling or cross-validation [21]. In the present study, the wrapper method is adopted, i.e. minimization of the classification error on the training set is used for optimization, primarily because of the non-binary nature of the problem at hand and the possible non-Gaussian distribution of training samples in the feature space.

In a multi-class problem, ideally one should jointly minimize every off-diagonal element of the confusion matrix (i.e. misclassified samples). However, in that case, the dimension of the objective space blows up with increase in the number of classes which is obviously impractical. Therefore, in the present work, two costs have been 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. Formally, let there be $Cl_1, \cdots, Cl_n$ classes of labeled time-series data given as the training set. A partitioning $\mathbb{B}$ is employed to extract features from each sample and a k-NN classifier $\mathbb{K}$ is used to classify them. After the classification process, the confusion matrix $\mathbf{C}$ is obtained, where the value of its element c_{ij} denotes the frequency of data from class Cl_i being classified as Cl_j. Let, $\mathbf{W}$ be the weighting matrix, where the value of its element w_{ij} denotes the penalty incurred by the classification process for classifying a data set from Cl_i as a data set from class Cl_j (usually $w_{ii} = 0$, so as to not penalize correct classifications). With these definitions, two costs that are to be minimized can be defined as follows. The cost due to expected classification error, $Cost_E$ can be defined as:

$$Cost_E = \frac{1}{N_s}\left(\sum_i \sum_j w_{ij} c_{ij}\right) \tag{2}$$

where, N_s is the total number training samples including all classes. The outer sum in the above equation sums the total penalty values for misclassifying each class Cl_i. Thus $Cost_E$ is related to the expected classification error. Although, in the current formulation, the total penalty values are equally weighted for all classes, that can be changed based on prior knowledge about the data and the user requirements.

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

$$Cost_W = \max_i \left(\frac{1}{N_i}\sum_j w_{ij} c_{ij}\right) \tag{3}$$

where, N_i is the number of training samples in class Cl_i. Note, that in the present construction of the objective space is two dimensional for a multi-class classification problem and the dimension is not a function of the number of classes, which makes it useful for classification with large number classes. As described earlier, classification needs to be performed on the training data set to calculate the costs during optimization of the feature extractor, i.e., the partitioning. Figure 5 depicts the general outline of the classification process. Labeled time series data from the training set are partitioned and the generated (by symbolization and PFSA construction) low-dimensional feature vectors are fed to the classifier. After classification, the two training error costs defined as above are computed and fed back to the feature extraction block. During classification, the classifier may be tuned to the obtain better classification rates. For example, for k-NN classifiers, choice of neighborhood size or the distance metric can be tuned. For support vector machines, identification of the best hyperplane can be made for better classification rate. Upon getting the feedback of the costs, the partitioning is updated to reduce the costs. The iteration goes on until the set of optimal partitionings (as it is a multi-objective scenario) and the correspondingly tuned classifier are obtained. Choice of the optimal partitioning is done using the Neyman-Pearson criterion as described later. After the

choice is made, the optimal partitioning and the tuned classifier are used to classify the test data set. Although this is the general framework that is being proposed for the optimization methodology, tuning of the classifier has not been performed in this paper as the main focus here is to choose the optimal partitioning to minimize the classification error related costs.

Similar to the classical partitioning cases, the value of k is chosen to be 5 and the distance metric is chosen as the Euclidean distance. For partitioning optimization, at first, the number of cells m of the partitioning $\mathbb{B}$ should be chosen ($m = 4$ in this case, same as it was for the classical partitioning schemes). Then the region $\Omega \in \mathbb{R}^1$ that circumscribes the one dimensional times series data space, is identified. For computation purpose, a suitably fine grid size depending on the data characteristics is then assumed. It should be clear that each of the grid boundaries denote a possible position of a partitioning cell boundary. In this paper, the data space region Ω is divided into 32 grid cells, i.e., 31 grid boundaries excluding the boundaries of Ω and there are 4 partitioning cells, i.e., 3 partitioning boundaries to choose. Hence, the number of elements in the space of all possible partitionings $\mathcal{P}$ is: ${}^{31}C_3 = 4495$

As the cardinality of $\mathcal{P}$ is computationally tractable, a straight forward search based Pareto optimization procedure is followed in this paper. By searching the space $\mathcal{P}$, the positions of its elements (the partitionings) are located in the two dimensional objective space. The Pareto front is generated by identifying the non-dominated points [23] in the objective space. In the present case, a non-dominated point (or partitioning) is such that no other partitioning has lower values of both $Cost_E$ and $Cost_W$ compared to that. Finally, the Neyman-Pearson criterion [23] is applied to choose the optimal partitioning $\mathbb{B}^*$ to have minimum $Cost_E$, while not allowing $Cost_W$ to exceed a certain value, say ε. In other words, the optimal partitioning $\mathbb{B}^*$ according to the Neyman-Pearson criterion is the solution to the following constrained optimization problem:

$$\mathbb{B}^* = \arg\min_{\mathbb{B}} Cost_E(\mathbb{B}),\ such\ that,\ Cost_W(\mathbb{B}) \leq \varepsilon \qquad (4)$$

The optimization results for the current problem and the performance of the optimal partitioning is discussed in the following subsection.

Results and Discussion

Given the confusion matrix obtained by using a partitioning and a classifier on the training set, a weighting matrix $\mathbf{W}$ needs to be defined to calculate the costs $Cost_E$ and $Cost_W$. In the present case, $\mathbf{W}$ is defined according to the adjacency properties of classes in the parameter space, i.e. $w_{ii} = 0,\ \forall i \in \{1,2,3,4\}$, i.e. there is no penalty when Cl_i is classified as Cl_i. However, in general $w_{ij} = |R_{\alpha_1}(i) - R_{\alpha_1}(j)| + |R_{\beta}(i) - R_{\beta}(j)|,\ \forall i \in \{1,2,3,4\}$, where $R_{\gamma}(k)$ denotes the range number (see Problem Description) for parameter γ in class k. Thus, $\mathbf{W}$ can be written as,

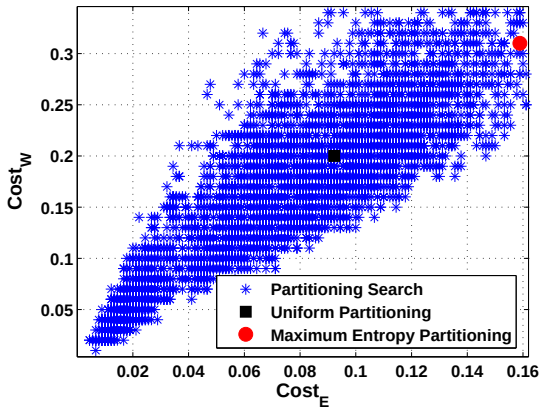

Figure 6. TWO DIMENSIONAL OBJECTIVE SPACE FOR PARTITIONING OPTIMIZATION

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

All the partitionings in space $\mathcal{P}$ are evaluated by calculating the costs $Cost_E$ and $Cost_W$. Figure 6 shows a portion of the two-dimensional objective space where the elements of space $\mathcal{P}$ are located. The location of the classical partitionings (i.e., UP and MEP) are also plotted along with the elements of $\mathcal{P}$ in the figure for comparative evaluation. The Pareto front is also generated by identifying the non dominated points in the objective space. The threshold ε, i.e., the maximum $Cost_W$ allowable is taken to be 0.025 in this case and the optimal partitioning (OptP) is chosen by the Neyman-Pearson criterion as described in the previous subsection. Figure 7 shows the location of the training features in the three dimensional plot using the first three linearly independent elements of the feature vectors obtained by using the chosen optimal partitioning OptP. Note that the class separability is retained by the feature extraction (partitioning) process even after compressing a time series data (with 8,000 data points) into 3 numbers.

Finally, the confusion matrices for Uniform, Maximum Entropy and the chosen optimal partitioning on the testing data set

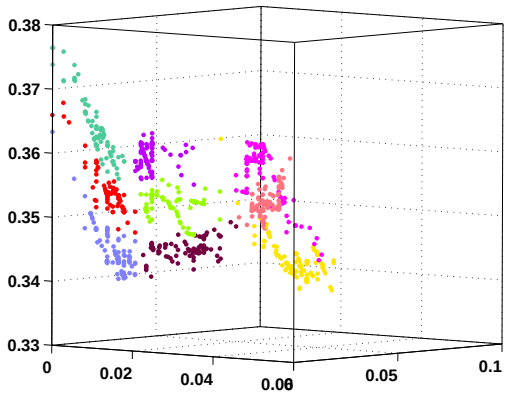

Figure 7. FEATURE SPACE OF TRAINING SET USING OPTIMAL PARTITIONING

are given by $\mathbf{C}_{test}^{UP}$, $\mathbf{C}_{test}^{MEP}$ and $\mathbf{C}_{test}^{OptP}$ respectively.

$$\mathbf{C}_{test}^{UP} = \begin{pmatrix} 94 & 5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 84 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 12 & 83 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 80 & 20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 95 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 95 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 87 & 9 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 19 & 80 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{pmatrix}$$

$$\mathbf{C}_{test}^{MEP} = \begin{pmatrix} 94 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 91 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 4 & 85 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 85 & 5 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 95 & 1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 4 & 6 & 86 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & 81 & 8 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 15 & 70 & 15 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 19 & 81 \end{pmatrix}$$

$$\mathbf{C}_{test}^{OptP} = \begin{pmatrix} 99 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 95 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 98 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 98 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 96 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 99 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 99 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 99 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{pmatrix}$$

Table 2 presents the comparison of all the classification performance related quantities, that are $Cost_E$, $Cost_W$ and the Classification error % for UP, MEP and OptP on the test set.

Table 2. COMPARISON OF CLASSIFICATION PERFORMANCES OF DIFFERENT PARTITIONING SCHEMES ON TEST DATA SET (100×9 samples)

Partitioning	$Cost_E$	$Cost_W$	Classification Error %
UP	0.1322	0.2300	11.33
MEP	0.2200	0.3700	14.67
OptP	0.0189	0.0500	1.89

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. 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 (or the number of partitions) in the *PSFA*. In our approach, during the process of optimization of the partitioning scheme, the number of partitions is kept constant. This, to a degree, alleviates the issue of over-training that could arise as a result of optimization performed on training set.

Summary, Conclusions and Future work

This article presents a data driven parameter identification technique in nonlinear dynamical systems using the concepts of nonlinear symbolic feature extraction from time series data. The problem has been formulated as a multi-class classification problem and optimization of feature extraction (i.e. partitioning) has been performed to enhance the classification rate. A multi-objective cost function has been constructed based on the classification error and it has been shown that using partitioning optimization, the classification rate can be improved beyond the performance of classical partitioning techniques. Moreover, it should be noted, that the success of the partitioning optimization process relies on the very nature of the slowly varying non-stationary evolution characteristics of the time series data over different classes. Suppose the evolution characteristics have very minimal signature in the time series space, then that will result in indistinguishability in the low dimensional feature space, inherently making the detection tool less efficient. However, an ill-posed problem like that may become a well-posed one when the data is transformed into another domain, such as wavelet

space or analytic signal space. Identification of suitable data pre-processing methods from the training data set is an important aspect, that will be investigated in future. Apart from this, the following research topics are currently being pursued as well.

1. Optimization of partitioning increases the sensitivity towards change in data characteristics, therefore, a comprehensive robustness analysis is required to make the classification process stable;
2. Use of other classifiers (e.g., Support Vector Machines) and comparison of performances among different classifiers;
3. Inclusion of the step of tuning the classifier inside the optimization loop as described in the general framework;
4. Use of other suitable cost functions and comparison of performances among them;
5. Extension of the optimization of partitioning methodology from classification to estimation perspective;
6. Application of the methodology to real life problems, e.g., fault level detection in complex human engineered systems, robotic type and gait classification from suitable sensor observation etc.

ACKNOWLEDGMENT

This work has been supported in part by NASA under Cooperative Agreement No. NNX07AK49A. 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.

REFERENCES

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

[2] Lee, T., 1998. *Independent component analysis: Theory and applications*. Kluwer Academic Publishers, Boston, USA.

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

[4] Weinberger, K., and Saul, L., 2004. "Unsupervised learning of image manifolds by semidefinite programming". *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04), Washington D.C.*

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

[6] Rao, C., Ray, A., Sarkar, S., and Yasar, M., 2008. "Review and comparative evaluation of symbolic dynamic filtering for detection of anomaly patterns". *Signal Processing*, p. in press.

[7] Rao, C., Mukherjee, K., Sarkar, S., and Ray, A., 2009. "Statistical estimation of multiple parameters via symbolic dynamic filtering". *Signal Processing,* **89**, June, pp. 981–988.

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

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

[10] Kennel, M., and Buhl, M., 2003. "Estimating good discrete partitions from observed data: symbolic false nearest neighbors". *Phys. Rev. E,* **91**(8), pp. 84–102.

[11] Subbu, A., and Ray, A., 2008. "Space partitioning via Hilbert transform for symbolic time series analysis". *Applied Physics Letters,* **92**(8), pp. 084107–1 to 084107–3.

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

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

[14] Jin, X., Sarkar, S., Mukherjee, K., and Ray, A., 2009. "Suboptimal partitioning of time-series data for anomaly detection". In Proceedings of 48th IEEE Conference on Decision and Control, Shanghai, China, pp. 1020–1025.

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

[16] Rajagopalan, V., Chakraborty, S., and Ray, A., 2008. "Estimation of slowly-varying parameters in nonlinear systems via symbolic dynamic filtering". *Signal Processing,* **89**(2), pp. 339–348.

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

[18] Choi, E., and Lee, C., 2003. "Feature extraction based on the bhattacharyya distance". *Pattern Recognition,* **36**, August, pp. 1703 – 1709.

[19] Loog, M., Duin, R., and Haeb-Umbach, R., 2001. "Multiclass linear dimension reduction by weighted pairwise fisher criteria". *IEEE Trans. Pattern Anal. Mach. Intell.,* **23**(7), p. 762766.

[20] Biem, A., Katagiri, S., and Juang, B., 1997. "Pattern recognition using discriminative feature extraction". *IEEE Trans. Signal Process.,* **45**(2), p. 500504.

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

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

[23] Steuer, R., 1986. *Multiple Criteria Optimization: Theory, Computations, and Application*. John Wiley & Sons, Inc., New York, USA.