LETTER Communicated by Hien Nguyen

A Locally Optimal Algorithm for Estimating a Generating Partition from an Observed Time Series and Its Application to Anomaly Detection

Najah F. Ghalyan
nfg103@psu.edu
The Pennsylvania State University, Department of Mechanical Engineering, University Park, PA 16802, U.S.A.

David J. Miller
djmiller@engr.psu.edu
The Pennsylvania State University, Department of Electrical Engineering, University Park, PA 16802, U.S.A.

Asok Ray
axr2@psu.edu
The Pennsylvania State University, Department of Mechanical Engineering, University Park, PA 16802, U.S.A.

Estimation of a generating partition is critical for symbolization of measurements from discrete-time dynamical systems, where a sequence of symbols from a (finite-cardinality) alphabet may uniquely specify the underlying time series. Such symbolization is useful for computing measures (e.g., Kolmogorov-Sinai entropy) to identify or characterize the (possibly unknown) dynamical system. It is also useful for time series classification and anomaly detection. The seminal work of Hirata, Judd, and Kilminster (2004) derives a novel objective function, akin to a clustering objective, that measures the discrepancy between a set of reconstruction values and the points from the time series. They cast estimation of a generating partition via the minimization of their objective function. Unfortunately, their proposed algorithm is nonconvergent, with no guarantee of finding even locally optimal solutions with respect to their objective. The difficulty is a heuristic nearest neighbor symbol assignment step. Alternatively, we develop a novel, locally optimal algorithm for their objective. We apply iterative nearest-neighbor symbol assignments with guaranteed discrepancy descent, by which joint, locally optimal symbolization of the entire time series is achieved. While most previous approaches frame generating partition estimation as a state-space partitioning problem, we recognize that minimizing the Hirata

N.G. and D.M. contributed equally to this letter.

Neural Computation **30, 2500–2529** (2018)
doi:10.1162/neco_a_01101

et al. (2004) objective function does not induce an explicit partitioning of the state space, but rather the space consisting of the entire time series (effectively, clustering in a (countably) infinite-dimensional space). Our approach also amounts to a novel type of sliding block lossy source coding. Improvement, with respect to several measures, is demonstrated over popular methods for symbolizing chaotic maps. We also apply our approach to time-series anomaly detection, considering both chaotic maps and failure application in a polycrystalline alloy material.

1 Introduction

Classification, modeling, and abstraction of (possibly noisy) time series have attracted great attention in machine learning and pattern recognition (Petridis & Kehagias, 1996; Ahmad, Alexander, Purdy, & Agha, 2017; Baragona & Battaglia, 2007). Symbolic dynamics techniques, which involve discretizing a time series, have been widely used to help achieve these objectives in an efficient manner (Godelle & Letellier, 2000; Daw & Finney, 2003; Mukherjee & Ray, 2014). These techniques are also useful for accurately describing a nonlinear dynamical system, by estimating its generating partition (Hirata, Judd, & Kilminster, 2004; Kennel & Buhl, 2003). While other important time-series modeling objectives include time-series segmentation and the related change point detection problem (Chamroukhi, Same, Govaert, & Aknin, 2009) and time series prediction (Wong & Li, 2000), the focus in this work is on time-series abstraction (symbolization or discretization), applicable to both estimating a nonlinear dynamical system's generating partition and estimating a time series null model, useful for detecting anomalous time series.

As noted, an important problem in symbolic dynamics is the estimation of a generating partition, a symbolization process generating a finite cardinality symbol sequence which, for certain dynamical systems, uniquely specifies the time series. Such symbolization is the basis for the evaluation of measures that characterize the (in general unknown) dynamical system, such as Kolmogorov-Sinai entropy (Beck & Schlogl, 1993; Kennel, Shlens, Abarbanel, & Chichilnisky, 2005). Symbolization also provides a convenient representation useful for anomaly/fault detection and time-series classification (Daw, Kennel, Finney, & Connolly, 1998; Kennel & Mees, 2000; Ray, 2004; Rao, Ray, Sarkar, & Yasar, 2009). Some previous work has proposed methods for estimating a generating partition from an observed time series (Kennel & Buhl, 2003; Hirata et al., 2004). A popular method is a clustering algorithm developed in the seminal work by Hirata et al. (2004). They derived a discrepancy clustering objective from first principles and cast estimation of the generating partition as minimization of their objective. While they recognized the value in developing a locally optimal algorithm for minimizing their objective, the actual algorithm they proposed is

nonconvergent, exhibiting nonmonotonic behavior with respect to their discrepancy objective during the algorithm's steps, with no guarantee even of local optimality with respect to the discrepancy cost. The difficulty lies in their use of a heuristic nearest-neighbor symbol assignment step, which is not guaranteed to descend in the discrepancy cost. To overcome this problem, our letter develops and validates a novel, locally optimal generating partition estimation (LOGPE) algorithm with respect to the discrepancy cost, with guaranteed convergence, based on an iterative nearest-neighbor symbol assignment step that possesses guaranteed descent in the discrepancy cost.[1] In this way, locally optimal symbolization of the entire time series is achieved. Some approaches perform scalar or low-dimensional vector quantization and thus explicitly achieve a low-dimensional state-space partition (Beck & Schlogl, 1993). By contrast, our approach does not explicitly partition the state space, but rather the space consisting of the entire time series. Since the time series may be of any length, the proposed approach thus essentially performs partitioning in a (countably) infinite-dimensional space. The proposed method also amounts to a novel type of sliding block lossy source coding (Gray, 1975). We demonstrate improvement in the discrepancy objective over Hirata et al. (2004) starting from the same initialization, as well as using their final solution as our method's initialization, with consistent gains seen across different system parameter choices. We also demonstrate better dynamical systems characterization with respect to the widely applied Kolmogorov-Sinai entropy, for several well-known nonlinear maps. Finally, LOGPE is demonstrated to yield improved anomaly detection for both a well-known chaotic map and a metal fatigue application domain.

The rest of the letter is organized as follows. Section 2 reviews the method from Hirata et al. (2004) and develops LOGPE. Section 3 presents experimental results for LOGPE in symbolization of time series generated by chaotic maps and in anomaly detection, in comparison with Hirata et al. (2004), the maximum entropy partition, and K-means clustering. Section 4 discusses some future work extensions.

2 Algorithm Formulation

2.1 Review of the Hirata et al. (2004) Method. Consider a discrete-time dynamical system

$$\underline{x}_{n+1} = f(\underline{x}_n), \tag{2.1}$$

[1] This paper is the journal version of Miller, Ghalyan, and Ray (2017), giving a fuller exposition of the method introduced there, more extensive experimentation, and applications to time-series anomaly detection.

where $n \in \mathcal{Z}$ and $f : \mathcal{R}^k \to U \subset \mathcal{R}^k$. Let $X = \{\ldots, \underline{x}_{-1}, \underline{x}_0, \ldots, \underline{x}_N, \ldots\}$, $\underline{x}_i \in \mathcal{R}^k$ be an infinite-duration (vector-valued) time series, generated according to this map. Further, define the infinite symbol sequence $\underline{s} = \{\ldots, s_{-1}, s_0, \ldots, s_N, \ldots\}$, $s_i \in \mathcal{A}$, $\mathcal{A}$ a finite alphabet, produced via some deterministic mapping operation that assigns a sequence $\underline{s}$ given a time series X. Hirata et al. (2004) focus on symbol sequences generated by applying state-space partitioning (i.e., of $\mathcal{R}^k$) to $\underline{x}_n, \forall n$. However, as we emphasize throughout this letter, this is not the only mapping of interest for generating $\underline{s}$. The purpose of the symbol sequence is to describe the time series. In particular, such a symbol sequence is called a generating partition if it uniquely specifies the initial state $\underline{x}_0$ (up to a set of measure zero). The initial state determines the subsequent time series and, further, if the map is invertible, it in fact determines the entire time series. Let us also define an infinite-duration reconstruction sequence $R = \{\ldots, \underline{r}_0(\underline{s}), \underline{r}_1(\underline{s}), \ldots, \underline{r}_N(\underline{s}), \ldots\}$, $\underline{r}_i(\underline{s}) \in \mathcal{R}^k$, where we can think of $\underline{r}_i(\underline{s})$ as an estimate of $\underline{x}_i$ that is informed by $\underline{s}$. Further, define the subsequence $\underline{s}[-m, m] = \{s_{-m}, s_{-m+1}, \ldots, s_0, s_1, \ldots, s_m\}$. Then $\underline{r}_0(\underline{s}[-m, m])$ is an estimate of $\underline{x}_0$ based on this finite subsequence of symbols. Hirata et al. (2004) prove that if $\underline{s}$ is such that $\underline{r}_0(\underline{s})$ can be chosen consistent with $\sup_{\underline{x}_0 \in U} ||\underline{x}_0 - \underline{r}_0(\underline{s}[-m, m])|| \to 0$ as $m \to \infty$, then the partitioning is generating. Further, note that the initial time is arbitrary. That is, if we consider $\underline{s}[-m + i, m + i]$, then if $\underline{r}_i()$ can be chosen so that $\sup_{\underline{x}_i \in U} ||\underline{x}_i - \underline{r}_i(\underline{s}[-m + i, m + i])|| \to 0$ as $m \to \infty$, the partitioning is also generating. That is, a (strict) generating partition uniquely specifies (or within a set of measure zero) the value of the time series at every time instant. Accordingly, a good estimate of a generating partition should be such that the sequence of reconstructions well approximates the time series (with small error at every time instant). The authors coin the phrase *symbolic shadowing*, referring to the reconstruction "alphabet" $\mathcal{R} = \{\underline{r}(\underline{s}) \in \mathcal{R}^k, \forall \underline{s}\}$ and a particular symbol sequence $\underline{s}$, jointly chosen so that the reconstruction sequence R (determined by $\underline{s}$) closely approximates the observed time series.

In order to develop a practical algorithm, one needs to consider a finite-duration time series $X = (\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_N\}$ and a practical (finite) reconsruction alphabet, $\mathcal{R}$. Accordingly, Hirata et al. (2004) define a finite temporal window, with the reconstruction at time n a function of the (possibly noncausal) length $m + l + 1$ symbol subsequence $\underline{s}[n] = \{s_{n-m}, s_{n-m+1}, \ldots, s_n, s_{n+1}, \ldots, s_{n+l}\}$. This limits the size of $\mathcal{R}$ to $|\mathcal{A}|^{m+l+1}$ k-dimensional vectors. Then, consistent with the symbolic shadowing idea, they define a goodness measure of reconstruction fidelity as the (mean-squared error-based) discrepancy:

$$D = \sum_{n=m}^{N-l} ||\underline{x}_n - \underline{r}(s_{n-m}, \ldots, s_{n+l})||^2.$$

Estimating a generating partition is thus boiled down in Hirata et al. (2004) to the optimization problem:

$$\min_{\mathcal{R},\underline{s}} \sum_{n=m}^{N-l} ||\underline{x}_n - \underline{r}(s_{n-m}, \ldots, s_{n+l})||^2.$$

This optimization problem bears close resemblance to that of K-means clustering (Duda, Hart, & Stork, 2012) and vector quantization (VQ) (Gersho & Gray, 1993). Accordingly, it is not surprising that Hirata et al. (2004) proposed an algorithm that seemingly resembles the Linde-Buzo-Gray (LBG) algorithm for designing vector quantizers (Linde, Buzo, & Gray, 1980). However, unlike the standard VQ problem, the reconstruction at time n is not a function of a single quantization index; it depends on the discrete symbol subsequence $\{s_{n-m}, s_{n-m+1}, \ldots, s_{n+l}\}$. It is this complication that causes nonmonotonicity and nonconvergence problems for the Hirata et al. (2004) algorithm. Thus, while the algorithm proposed in Hirata et al. (2004) (which is detailed next) seems to resemble LBG, unlike LBG, which possesses guaranteed convergence and local optimality properties, the algorithm from Hirata et al. (2004) does not possess these properties and suffers from nonmonotonicity and nonconvergence problems.

Suppose that m, l, and $|\mathcal{A}|$ are fixed.[2] Further, we introduce the indicator variable $\upsilon_{n,\underline{q}}$, $\underline{q} \in \mathcal{A}^{l+m+1}$, taking on 1 if $\underline{s}[n] = \underline{q}$, that is, if it is the symbol subsequence mapping (specifying the reconstruction) for sample $\underline{x}_n$ and 0 otherwise. The Hirata et al. (2004) algorithm, which aims to minimize D, is then summarized as follows:

1. *Initialization*. Choose an initial reconstruction table $\mathcal{R}$ with associated symbol sequence "codewords." While a variety of initializations are possible, Hirata et al. (2004) proposed one based on unstable periodic state-space points from the time series.
2. *"Nearest neighbor" symbolization Step*. For $n = m, \ldots, N - l$

$$\underline{q}^* = \operatorname*{argmin}_{\underline{q} \in \mathcal{A}^{l+m+1}} ||\underline{x}_n - \underline{r}(\underline{q})||^2$$

$$s_n = q_0^*, \text{ where } \underline{q}^* = (q_{-m}^*, \ldots, q_l^*)$$

End For

[2] In practice, these are all hyperparameters of the algorithm that must be chosen. One such approach, which will be considered in our future work, is to cut the time series in two, into a training "half" and a validation "half," with the best hyperparameters' configuration the one that minimizes the discrepancy measured on the validation half of the time series. Another is to apply K-fold cross-validation. However, this will entail $(K + 1)$ objective function minimizations for each hyperparameter configuration and is thus more costly computationally.

3. *Centroid rule.* $\underline{r}(\underline{q}) = \frac{\sum_{n=m}^{N-l} v_{n,\underline{q}} \underline{x}_n}{\sum_{n=m}^{N-l} v_{n,\underline{q}}}$, evaluated for all $\underline{q} \in \mathcal{A}^{l+m+1}$ for which the denominator is nonzero.
4. *Termination.* Go to step 2 unless there are no further changes or a stopping condition is met.

The centroid step is a global minimization step, given the symbol sequence $\underline{s}$ fixed. Thus, this step descends in D. However, Hirata et al. (2004) were well aware that step 2 is not guaranteed to descend in D. In fact, they noted that the ideal step 2 would globally minimize D with respect to the symbol sequence, given fixed $\mathcal{R}$. However, they recognized that this is utterly intractable, requiring exhaustive evaluation of all $|\mathcal{A}|^{(N+1)}$ symbol sequences. They justified their chosen, heuristic step (which assigns as s_n the "middle" value q_0^* in $(q_{-m}^*, \ldots, q_0^*, \ldots, q_l^*)$) by noting that in the "majority of tested cases, a stationary state is achieved". Moreover, they also suggested the use of their nearest-neighbor step to estimate a state-space (generating) partition. That is, they divide $\mathcal{R}^k$ into $|\mathcal{A}|$ regions, where the region indexed by $a \in \mathcal{A}$ is defined by $\{\underline{x} : ||\underline{x} - \underline{r}(\underline{q})||^2 \leq ||\underline{x} - \underline{r}(\underline{q}')||^2$, where $\underline{q}$ satisfies $q_0 = a, \forall \underline{q}' \in \mathcal{A}^{l+m+1}\}$.

Experimentally, however, we have found that nonconvergence and oscillatory behavior of their algorithm are also often observed. This necessitates use of heuristic stopping criteria. Moreover, since the symbolization step is not guaranteed to descend in D, there is concern whether Hirata et al. (2004) makes good descent in D before the stopping condition is reached.

While there are problems with the algorithm proposed in Hirata et al. (2004), their work is seminal in that they derived their discrepancy objective function from first principles and explicitly cast generating partition estimation as the minimization of their objective function. In this letter, we build on their foundational work, proposing an algorithm that is locally optimal with respect to D, with guaranteed convergence. When initialized using a Hirata et al. (2004) solution, our method is guaranteed to find better solutions unless the Hirata solution is already at a local minimum of D. Moreover, when given the same initialization as Hirata et al.'s (2004) method, our algorithm often finds much better solutions. In fact, this initialization is generally preferred in practice.[3]

2.2 LOGPE Algorithm. The key observation we make is that since the reconstruction at time n' is a function of $(s_{n'-m}, \ldots, s_{n'+l})$, the choice of s_n affects (restricts) the reconstruction choices for time instants $n' = n - l, \ldots, n + m$.[4] For example, if $l = m = 1$, s_n restricts the possible

[3] Initializing from a poor (Hirata) solution may be an unfavorable starting point for our method, potentially close to a poor local minimum.

[4] This observation in fact makes it clear that the Hirata symbolization step is not a descent step in D, since their step first considers an unrestricted assignment for $\underline{x}_n$

reconstructions at time instants $n-1$, n, and $n+1$. Thus, s_n should be chosen to minimize the sum of the squared reconstruction errors incurred at these time instants. Specifically, if $\mathcal{R}$ is fixed and all other symbol assignments are fixed, choosing s_n in this way is optimal and is a guaranteed descent step in D. Accordingly, we specify the following locally optimal algorithm.

1. *Initialization*. Choose an initial symbol sequence $\underline{s}^{(0)}$ and reconstruction table $\mathcal{R}^{(0)}$. The Hirata et al. (2004) initialization is one good initialization approach that can be chosen. For all symbol subsequences that occur in $\underline{s}^{(0)}$, initialize their associated reconstructions using the centroid rule. For the remaining possible subsequences, choose a common initial reconstruction value (e.g., the mean value of the time series is a reasonable choice). In this way, $\mathcal{R}^{(0)}$ is determined; $t \leftarrow 0$.
2. *Symbolization pass*.

$$t++$$

$$\text{For } n = m, \ldots, N-l$$

$$s_n^{(t)} = \underset{s_n \in \mathcal{A}}{\operatorname{argmin}} \sum_{n'=n-l}^{n+m} ||\underline{x}_{n'} - \underline{r}^{(t-1)}(s_{n'-m}^{(t-1)}, \ldots, s_n, \ldots, s_{n'+l}^{(t-1)})||^2$$

   End For[5]
3. *Centroid rule*. Same as Hirata et al. (2004).
4. *Termination*. Go to step 2 unless there are no further changes.

A number of comments follow:

• *Symbolization that descends in D.* Note that step 2 is equivalent to $\operatorname{argmin}_{s_n} D$, given all other variables fixed. Thus, each symbol choice is a descent step in D.

• *Convergence.* The LOGPE algorithm is guaranteed to converge. The proof argument, which follows standard arguments for convergence of vector quantizer design (Gersho & Gray, 1993), is as follows. Although there are a huge number of possible symbol sequence assignments ($|\mathcal{A}|^{(N-l-m+1)}$), this number is finite. The two algorithm steps defining an iteration (centroid and symbolization pass) are both nonincreasing in D (the centroid step in fact globally minimizes D with respect to $\mathcal{R}$, given $\underline{s}$ fixed). Moreover, the centroid updates can be written as a function of the symbol assignments. Thus,

(considering the entire reconstruction table), which may lead to a reconstruction choice whose associated state subsequence is inconsistent with the $n+m$ (currently) fixed symbol values in the temporal window $[n-l, n+m]$ (i.e., all symbols except s_n).

[5] If multiple symbols attain the minimum distance given by the summation in this step, then the first of these symbols is chosen. This is the tie breaker rule we use in this letter, although other rules can also be used, such as randomly choosing from among this subset of symbols.

we can focus on the symbolization pass. The number of possible symbolization passes that may decrease D following the initialization is of course also finite (it is bounded below the total number of symbol sequence assignments). Thus, after a finite number of symbolization passes, D will no longer decrease. Thus, LOGPE converges in a finite number of passes (iterations). Furthermore, if there is always a unique state minimizing the partial discrepancy in step 2 (ties occur with measure zero for a continuous-valued map, $f(\cdot)$), LOGPE also converges to a (fixed point) symbol sequence.

• *Local optimality (following the proof argument given in Gray, Kieffer, & Linde, 1980 for vector quantization)*. Suppose that LOGPE has converged to a (fixed-point) symbol sequence $\underline{s}$. As noted above, the centroid rule determines the global minimum choice of $\mathcal{R}$ with respect to D, given fixed $\underline{s}$. Now suppose we consider a perturbation of $\mathcal{R}$ (i.e., a reconstruction table $\tilde{\mathcal{R}}$). Local optimality means that for every $\tilde{\mathcal{R}}$ such that $||\mathcal{R} - \tilde{\mathcal{R}}|| < \epsilon$ and with ϵ made as small as you like, the discrepancy cost will be larger when $\tilde{\mathcal{R}}$ is used compared with $\mathcal{R}$ (i.e., compared with the global centroid choice, given $\underline{s}$ fixed). This is shown as follows. If ϵ is made sufficiently small, applying the symbolization pass using any reconstruction table $\tilde{\mathcal{R}}$ will yield the same state sequence $\underline{s}$ as that determined when using $\mathcal{R}$. But $\mathcal{R}$ is the globally optimal reconstruction table with respect to D, given $\underline{s}$. Thus, the discrepancy incurred using $\tilde{\mathcal{R}}$ must be strictly greater than that using $\mathcal{R}$. We thus conclude that at convergence (assuming the state sequence $\underline{s}$ has converged (no ties)), the LOGPE reconstruction table $\mathcal{R}$ is locally optimal.

• *Variable memory*. Note that the LOGPE algorithm need not require dedicated memory storage for all $|\mathcal{A}|^{(l+m+1)}$ vector reconstructions; we need not (initially) store the reconstruction values for symbol sequences that do not occur, based on $\underline{s}^{(0)}$. Essentially, in step 2, whenever a symbol subsequence that has not been seen before is being evaluated, we do not need to do a table lookup to find its reconstruction value; we can simply use the rule of evaluating that reconstruction as the (chosen) common value (e.g. the mean of the time series). The implication here is that in practice, variable memory allocation can be used, with a new reconstruction added to the table $\mathcal{R}$ each time a given symbol subsequence is found to have occurred for the first time in the current $\underline{s}$, following step 2.

• *Alternative implementation*. Step 2 performs one pass over the time series. A valid alternative is to perform multiple passes until there are no further changes, before performing the centroid rule. Both approaches are locally optimal, with guaranteed convergence.

• *Sequence space partition*. Since the symbol assignments for the entire time series are jointly determined, our approach explicitly partitions in the space consisting of the entire time series, that is, a sequence space partition (it does not explicitly induce a partitioning of the k-dimensional state space). Specifically, for $\underline{x}_n = \underline{x}_{n'} = \underline{\tilde{x}}$, our symbolization could be such that $s_n \neq s_{n'}$. To put this in perspective, Kennel and Buhl (2003) suggest that symbolization necessitates forming a partition of the state space: "The obligatory

discretization requires a partition of the state space." Our approach demonstrates this is not necessary. In fact, discretizing consistent with minimizing D only performs explicit partitioning in a (countably) infinite-dimensional space (considering time series of arbitrary integer length, N). Note that the above does not imply that our approach does not yield smooth state-space partition estimates in practice (we will give an example shortly).

• *Lossy source coding interpretation*. The above clustering algorithm is also a special type of sliding block lossy source coding design algorithm (Gray, 1975), which, to our knowledge, is itself novel. The associated source coder operates on k-dimensional input blocks, has a source coding rate of $(\log_2 |\mathcal{A}|)/k$ bits per sample, requires infinite delay at the encoder side (it jointly encodes the entire time series), and uses a decoder codebook of size $|\mathcal{A}|^{l+m+1}$ k-dimensional reconstructions.

• *Distinction from time series clustering*. In LOGPE (and in the Hirata et al., 2004, algorithm) we are essentially discretizing or symbolizing a single (possibly vector-valued) time series, with the measurement $\underline{x}_n$ at each time n assigned to one of $|\mathcal{A}|$ possible values. For a time series of length N, there are $|\mathcal{A}|^N$ possible such sequence symbolizations. By contrast, in time series clustering (Aghabozorgi, Shirkhorshidi, & Wah, 2015; Nguyen, McLachlan, Orban, Bellec, & Janke, 2017) one has a collection of, for example, M time series, with each of these time series assigned to one of $|\mathcal{A}|$ clusters. This is wholly different from our time-series symbolization problem. The similarities are that both problems involve time series and both problems involve assigning data to an element from a finite set ($|\mathcal{A}|$ possible values). The fundamental differences lie in the facts that in time-series clustering, there is a finite collection of time series, whereas in our problem, there is only a single time series. Moreover, in time-series clustering, the entire time series is a data object getting assigned (in its entirety) to a cluster, whereas in our problem, the measurement at each time is individually assigned to one of the possible symbol values (which are not really "clusters" per se).

• *Distinction from time-series segmentation and prediction*. LOGPE performs a time-series symbolization especially useful for estimating the generating partition for an unknown nonlinear dynamical system. We will demonstrate that this approach is also very effective for learning a time-series null model, used to interrogate new time series to detect an anomalous time series or anomalous time-series segment (e.g., for fault detection). There are time-series modeling approaches that also involve symbolization, albeit focused on other objectives, for which they are better suited and for which our model is not really appropriate. However, these other models are not necessarily appropriate for symbolizing nonlinear dynamical systems. In particular, Chamroukhi et al. (2009) focus on time-series segmentation, seeking to partition the time series into K segments, each represented by a different polynomial model. In such an approach, every measurement in the same segment is essentially assigned the same symbol value (k). That is, the symbol sequence contains long intervals where the same symbol value is used

(indicating the same polynomial model is being used). But LOGPE's symbol sequence is not likely to be smooth in this sense at all, and the LOGPE symbols do not carry the same meaning as those in Chamroukhi et al. (2009). LOGPE's sequence discretization is a joint encoding of the entire time series, yielding a symbol sequence that mirrors (tracks) the original time series produced by the nonlinear map. Rather than partitioning the time axis as in Chamroukhi et al. (2009), LOGPE, as will be discussed next, induces partitions of the state space (Hirata et al., 2004; Kennel & Buhl, 2003) and the sequence space. Likewise, Wong and Li (2000) proposed a mixture of autoregressive models. Again, this model does involve (latent) symbolization of the time series, with a (latent) autoregressive mixture component used to generate the observation at each time instant. However, the focus in Wong and Li (2000) is on time-series prediction (determining the predictive distribution), not on "describing" the time series by a discrete (mixture component) symbol sequence. Also, Wong and Li (2000) is well suited to modeling data generated according to mixtures of autoregressive models, but not, obviously, to data generated according to a nonlinear dynamical system. Other differences between LOGPE and Wong and Li (2000) are that LOGPE does not directly learn a (stochastic) generative model; moreover, LOGPE has nonlinear (quantized) memory, with the reconstruction at the current time a function of a finite memory symbol sequence, whereas in Wong and Li (2000), conditioned on each mixture component, there is a linear (autoregressive) memory model.

2.3 Partitions of the State Space and the Sequence Space. Consider a discrete-time dynamical system described by the map

$$\underline{x}_{n+1} = f(\underline{x}_n), \tag{2.2}$$

where $f : \mathcal{R}^k \to \mathcal{R}^k$. Let $\mathcal{M} = \{X\}$ denote the set of all possible time series generated by equation 2.2. Let $\mathcal{P}$ be a partition, mapping each time-series $X = \{\underline{x}_0, \underline{x}_1, \underline{x}_2, \ldots\}$ generated by equation 2.2 to a symbol sequence $\underline{s} = \{s_0, s_1, s_2, \ldots\}$, $s_i \in \mathcal{A}$, $\mathcal{A}$ a finite alphabet. The shift space Σ is the set of all possible such symbol sequences (Lind & Marcus, 1995).

As indicated in section 2.2, LOGPE explicitly partitions in the sequence space $\mathcal{M}$ and does not necessarily partition the state-space $\mathcal{R}^k$. An interesting question is, In the symbolic dynamics literature, which partition definition is used—a partition of the state space (which of course also determines a partition of the sequence space) or just a partition of the sequence space ? Multiple references imply that the relevant definition is a partition of the state space (Beck & Schlogl, 1993; Buhl & Kennel, 2005).

However, Cornfeld, Fomin, and Sinai (1982) use a sequence space partition definition. Likewise, Kantz and Schreiber (2004) describe partitioning as the process of labeling specific patterns in the time series (not

necessarily of fixed length) by symbols. Our LOGPE method forms such a sequence partition $\mathcal{P} : \mathcal{M} \to \Sigma$. Such a partition is generating if this map is one-to-one up to a set of measure zero.[6]

Like Beck and Schlogl (1993) and Buhl and Kennel (2005), Hirata et al. (2004) use a state-space partition definition. The derivation of their clustering objective in fact begins from the assumption that one is seeking a partition of the state space, and their theoretical results all assume a state-space partition. However, this is somewhat ironic, because their resulting algorithm does not, in fact, optimize over a state-space partition in attempting to minimize their discrepancy measure, D. In fact, their symbol assignment step is not consistent with a state-space partitioning rule (the symbol assignment at time n in their method depends on the symbol choices at surrounding times). At the same time, their algorithm does not exploit this "freedom" (i.e., not being restricted to assign consistent with a state-space partition) to define an algorithm that descends in D.

We also relax the assumption of a state-space partition, but unlike Hirata et al. (2004), we do exploit this relaxation to define an algorithm that descends in and is locally optimal with respect to D. In fact, to achieve a locally optimal solution with respect to D, one must define a sequence partition (obviously, the global minimizer of D is a sequence space partition, not a state-space partition, as is even pointed out in Hirata et al. (2004)). As we will show next, our approach not only yields solutions with lower D than those of Hirata et al. (2004), as one would expect. It also yields lower maximum absolute error; better estimation of Kolmogorov-Sinai entropy, a unique characterizer of a chaotic map; and better performance when applied to time-series anomaly detection.

3 Experiments

3.1 Symbolization of Time Series Generated by Chaotic Maps. In this section we validate LOGPE on several chaotic maps, including the Ikeda map (Ikeda, 1979), described by the following pair of equations:

$$x_{1,n+1} = a + b(x_{1,n} \cos \phi_n - x_{2,n} \sin \phi_n),$$
$$x_{2,n+1} = b(x_{1,n} \sin \phi_n + x_{2,n} \cos \phi_n),$$

[6] It is worth mentioning that for many time-varying dynamical systems, the same point in state space may recur multiple times. However, for each such occurrence, the point may be part of a quite distinct (short- or long-term) temporal pattern. For a state-space partition, the time series must be assigned the same symbol at these "recurrence" times. However, it should be clear that such a restricted symbol assignment may be suboptimal, with the symbol at time n, s_n, giving information only about $\underline{x}_n$. By contrast, a sequence partition is less restrictive, with s_n, each n, providing some information about the entire time series.

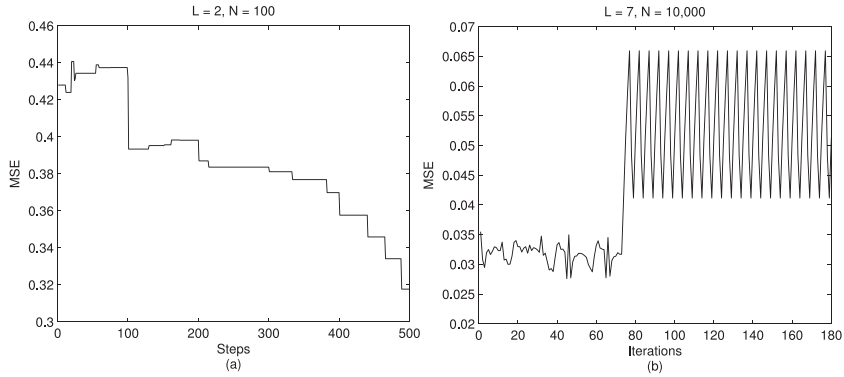

Figure 1: (a) Nonmonotonicity in D of Hirata et al. (2004). (b) Limit cycles of Hirata et al. (2004).

where $\phi_n = \kappa - \eta/(1 + x_{1,n}^2 + x_{2,n}^2)$. We choose $a = 1$, $b = 0.9$, $\kappa = 0.4$, $\eta = 6$, and an initial condition $(x_{1,0}, x_{2,0}) = (0.5328, 0.2469)$ as in (Davidchack, Lai, Bolt, & Dhamala, 2000). N denotes the time-series length and $L = m + l + 1$ the window size. Also, we chose $A = \{0, 1\}$. In this section, whenever a figure is for a fixed window length, we used $l = 0$. When the figure is for an increasing window length, we used the procedure in (Hirata et al., 2004) to choose l and m, as follows:

1. Choose an initial window length L.
2. $l = \lfloor L/2 \rfloor$ and $m = \lfloor (L-1)/2 \rfloor$.
3. $L \leftarrow L + 1$, and goto 2.

First, we illustrate nonmonotonicity in D of the algorithm from Hirata et al. (2004). Figure 1a shows MSE versus algorithm steps for a length $N = 100$ Ikeda sequence, where there are 99 symbol assignment steps ($N - l - m$), followed by a centroid step. It is clear from the figure that Hirata et al. (2004) is not strictly decreasing in D. As a consequence, there is no guarantee the algorithm will converge. Figure 1b demonstrates an example where limit cycle behavior occurs. Here, in this example, the centroid update and a symbol assignment sweep over the time series are the two steps consisting of one iteration.

Figure 2a shows a comparison between Hirata et al. (2004) and LOGPE for 12 sweeps over the time series (with each symbol update and the centroid update considered to be steps). Both algorithms used precisely the same initial conditions, based on the procedure from Hirata et al. (2004). In this case, Hirata converges. However, LOGPE converges to a much smaller value of D, with significant descent occurring long after Hirata et al. (2004) converged to a (shallower) fixed point. In Figure 2b LOGPE is initialized

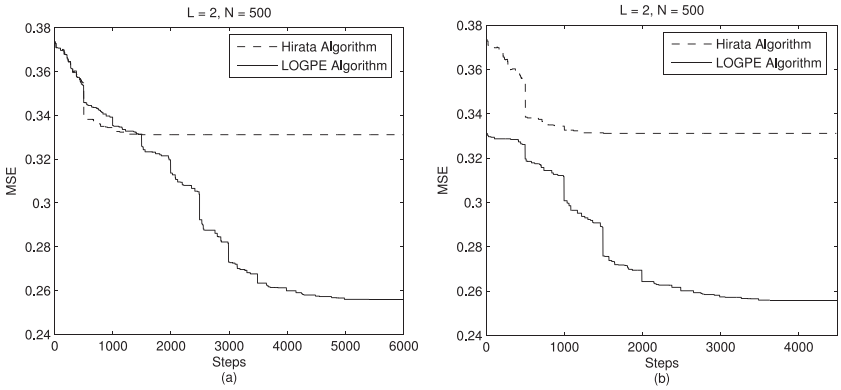

Figure 2: Comparison of Hirata et al. (2004) and LOGPE algorithms. (a) Same initialization. (b) LOGPE initialized from Hirata et al. (2004) converged solution.

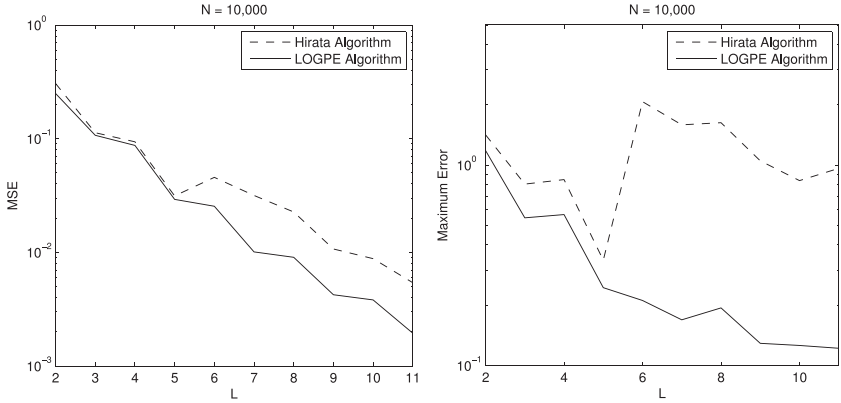

Figure 3: Comparison of Hirata et al. (2004) and LOGPE algorithms.

from the converged Hirata solution. In this case, LOGPE is mathematically guaranteed to improve on (or do no worse than) the Hirata et al. (2004) solution. As seen from the figure, LOGPE converges to a solution with essentially the same MSE LOGPE achieved in Figure 2a, but in 9 iterations (4491 steps) instead of 12 (5988 steps).

We next consider longer time series – $N = 10,000$, with both algorithms again using precisely the same initialization. This time we plot performance at stopping, as a function of L. The results are shown in Figures 3 and 4. As shown in Figure 3, Hirata et al. (2004) does not strictly improve in D with

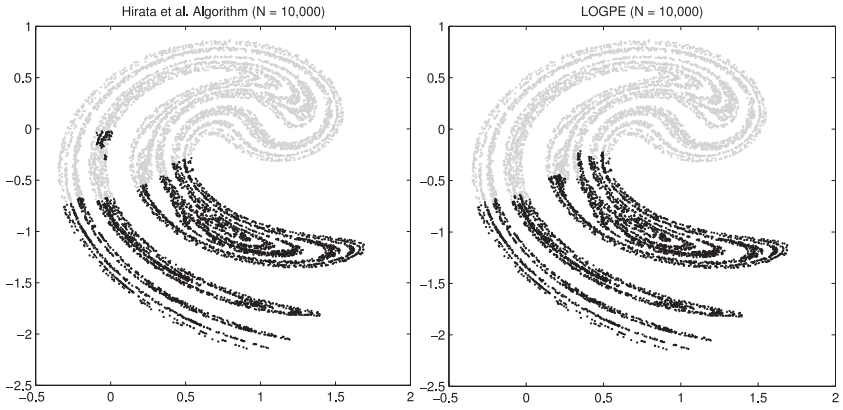

Figure 4: Estimated partition using the Hirata et al. (2004) and LOGPE algorithms.

increasing L.[7] This is not totally surprising due to the heuristic nature of the algorithm, which may result in finding a poor solution at a given L, and nonconvergence at a given L, which necessitates use of a heuristic termination policy, possibly again resulting in a poor (final) solution.[8] However, this result is still disappointing. Since $|\mathcal{R}|$ grows exponentially with L, one might still expect D to strictly decrease with increasing L. This monotonic behavior is in fact observed for LOGPE. Note that LOGPE also gives significant MSE performance gain over Hirata as L increases. Finally, as shown in Figure 3, LOGPE also gives much better maximum absolute error.[9]

Suboptimality of Hirata et al. (2004) is not only reflected in MSE and maximum error performance. It is also reflected with respect to the main objective of the algorithm: estimating a generating partition. The estimated partitions for $L = 11$ and $|\mathcal{A}| = 2$ are shown for both algorithms in Figure 4.[10]

[7] While we faithfully implemented the Hirata algorithm in its entirety, our results in Figures 3 and 4 are not perfectly consistent with those reported in Hirata et al. (2004) for the Ikeda map. Specifically, Hirata et al. (2004) does not show nonmonotonicity for D as a function of L and they also do not show nonsmoothness in the estimated state-space partition.

[8] The stopping policy we used when implementing the Hirata algorithm is that if the algorithm does not reach a fixed point, it is terminated after a maximum number of iterations is reached, with the best solution (lowest D) found up to that point retained. The exception is that whenever limit cycles are detected, the algorithm is terminated when the cost function (D) reaches a minimum value of the limit cycle.

[9] Although LOGPE (designed to minimize D) does not exhibit strictly monotonically decreasing maximum error for increasing L, there is a clear monotonic decreasing tendency, in contrast to what is seen for Hirata et al. (2004).

[10] A state-space partition is estimated for both algorithms by labeling each sample $\underline{x}_n$ by the symbol s_n.

Here, we see that LOGPE achieves a smooth manifold partition of the state space (expected and desirable), whereas the partition of Hirata et al. (2004) possesses two regions of discontinuity.

There are two further important points to emphasize about Figure 4. First, it is not possible for a state-space partitioning algorithm such as K-means/LBG (applied to the vector measurements that form the time series) to have produced these state-space partitions. For Figure 4, a symbol alphabet of cardinality two was used. In general, vector quantization (assuming squared Euclidean distance) induces a Voronoi partition on the given feature space, with each cluster's region a convex polytope. However, the situation here is actually far simpler. The number of clusters is only two: the Voronoi partition's decision boundary (in the plane) in this case is mathematically guaranteed to be a straight line. Note that neither the LOGPE nor the Hirata estimated partitions are consistent with a straight-line decision boundary. Thus, neither of them could have been produced by a K-means/LBG algorithm (with symbol alphabet/number of clusters equal to two). Second, we note that the LOGPE partition is in very good (visual) agreement with the published estimated generating partition (obtained using a different method) in Davidchack et al. (2000). This is a further validation of our obtained partition and of our approach.

We also provide an estimate of Kolmogorov-Sinai (K-S) entropy, h, for the Ikeda map using both algorithms. By definition, K-S entropy is the supremum of the joint entropy of all symbol sequences taken over all alphabets (see, e.g., Beck & Schlogl (1993), and Kantz & Schreiber (2004)). Theoretically, this supremum can be achieved by partitioning the state space with an infinite-cardinality alphabet. Interestingly, a generating partition achieves the same supremum with finite, and it is hoped, a very small, alphabet. Furthermore, K-S entropy is unique to each map (Beck & Schlogl, 1993). In particular, for the Ikeda map, the true K-S entropy has been approximated in the literature by the value 0.726 (Hirata et al., 2004). Given a symbol sequence $\{s_1, s_2, \ldots, s_N\}$, we construct a finite set of states $Q = \{q_i\}$, where each state q_i is a concatenation of symbols occurring in the symbol sequence. One way to create such a set of states is to use the so-called context-tree model (Willems, Shtarkov, & Tjalkens, 1995). Then the set Q is the collection of all terminal nodes in the tree. It is clear that given a tree depth and an alphabet $\mathcal{A}$, there are many possible tree models that could be constructed. However, tree models that satisfy the well-known minimum description length (MDL) model-selection criterion (Barron, Rissanen, & Yu, 1998) provide an efficient and reliable means to model a time series (Kennel et al., 2005). To compute the K-S entropy, we use the estimate $h = L_{MDL}/N$, where L_{MDL} is the data code length component of the MDL given by $L_{MDL} = \sum_{j=0}^{N-1} -\log_2 p(s_{j+1}|q^{(j)})$, where $q^{(j)} \in Q$ is the most recent state (including symbols up to time j), and $p(s_{j+1}|q^{(j)})$ is estimated from frequency counts (Kennel & Buhl, 2003). The resulting K-S entropy estimate is

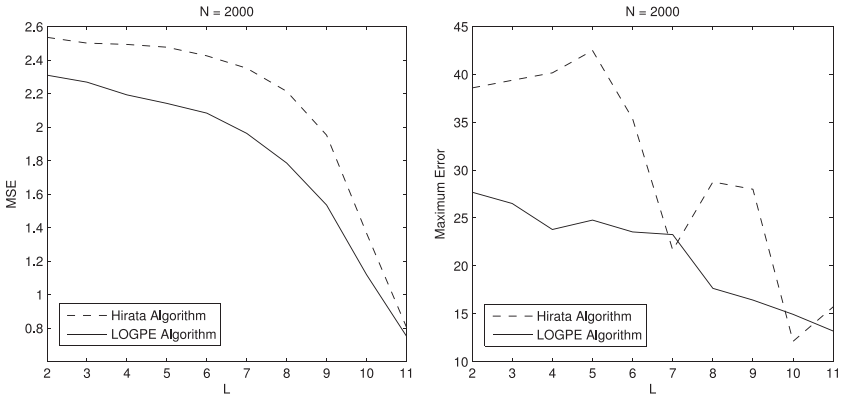

Figure 5: Comparison of Hirata et al. (2004) and LOGPE algorithms with additive noise.

$h = 0.6915$ for Hirata et al. (2004) and $h = 0.7239$ for LOGPE. The K-S entropy was computed for both algorithms using $L = 11$ and an alphabet size of two. Clearly, LOGPE's estimate is very close to the true K-S entropy and closer than Hirata's estimate. As a second example, we also compute the K-S entropy for the Hénon map (Kantz & Schreiber, 2004), given by

$$x_{1,n+1} = 1 - ax_{1,n}^2 + bx_{2,n},$$
$$x_{2,n+1} = x_{1,n},$$

where $(a, b) = (1.4, 0.3)$. For this map, the true entropy is approximated by the value 0.6048 (Hirata et al., 2004). We follow the same procedure we did to compute the K-S entropy for the Ikeda map. The resulting K-S entropy is $h = 0.6204$ for Hirata et al. (2004) and $h = 0.6084$ for LOGPE. Again, LOGPE's estimate is very close to the true K-S entropy and closer than Hirata's estimate. These K-S entropies computed by the LOGPE algorithm for both the Ikeda map and the Hénon map are suggestive of the accuracy of the algorithm in estimating the generating partition for these two maps using an alphabet size of just two.

Figure 5 considers the Ikeda map corrupted by additive noise. We used zero-mean, gaussian white noise with a variance of two in both dimensions. The figures show significant improvement in both MSE and the maximum error curves for LOGPE compared with Hirata et al. (2004).

Although Figure 3 shows LOGPE achieves monotonic decrease in D for increasing L, this is in fact not guaranteed because LOGPE produces only a locally optimal solution at a given L. To ensure such descent, we propose a prefix-based initialization of the solution for window length $L + 1$, given the solution at length L. After updating L to $L + 1$, keep the symbol sequence

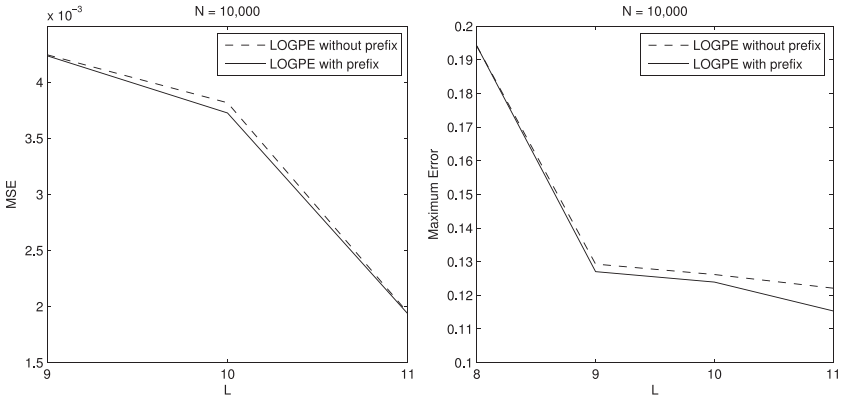

Figure 6: LOGPE performance with and without prefix-based initialization.

fixed; then consider the smaller window size symbol subsequence as a prefix, and for all the larger window size symbol subsequences with the same prefix, assign them to the same centroid reconstruction. This makes the solution at window size $L+1$ initially equivalent to the solution at window size L (with the same value of D). Now, if we take one centroid step for the larger window size, this will descend in D (unless already at a local minimum) and will improve the solution. Hence, a guaranteed decrease in D for increasing L is achieved. Figure 6 shows that beyond theoretically guaranteed decrease in D for increasing L, there is modest performance gain for this prefix-based approach, compared with LOGPE initialized using the method from Hirata et al. (2004) at each L.

3.2 Computational Complexity. We now explicate the computational complexity of LOGPE in comparison to Hirata et al. (2004). The two algorithms use the same centroid update rule, so excepting differences in the size of the reconstruction tables for the two algorithms, the main difference lies in the symbolization step. Inspecting the given pseudocodes, one can see that for Hirata et al. (2004), to determine each symbol (in one sweep or pass over the time series), one must perform a nearest-neighbor rule, exhaustively evaluating all possible vectors in the reconstruction table to find the one that is nearest to the time-series vector at the current time, $\underline{x}_n$. This requires at most $|A^{l+m+1}|$ vector distance computations (the actual number required is the number of reconstructions that occur for the training set (the size of the reconstruction table), which may be much smaller). On the other hand, for LOGPE, one only needs to consider $|A|$ reconstructions (since one is only considering changing s_n). However, for each such choice, $l+m+1$ vector distance computations (and $l+m$ additions) are required to implement the summation in a LOGPE symbolization step at time n. Thus, for

determining s_n, LOGPE requires $|A|(l+m+1)$ vector distance computations and $|A|(l+m)$ additions. While this analysis characterizes the complexity of the two algorithms at each symbol update step, both the actual size of the reconstruction table that is used and the number of symbolization sweeps performed until convergence or termination will determine the relative required execution times of the two algorithms in practice.

Thus, to assess execution times, we conducted 10 trials, in each generating an Ikeda time series of length 10,000 from slightly different initial conditions, and then we report average execution times (and standard deviation), as well as the average number (and standard deviation) of iterations (passes over the data) until convergence, as a function of L. The 10 initial conditions were $\underline{x}_0 =$ (0.53280, 0.24690), (0.53281, 0.24690), (0.53280, 0.24691), (0.53281, 0.24691), (0.53282, 0.24690), (0.53280, 0.24692), (0.53282, 0.24691), (0.53281, 0.24692), (0.53282, 0.24692), and (0.53283, 0.24690). The experiments were performed on a Dell Precision T3400, with an Intel Core2 Quad CPU Q9550 at 2.83 GHz, with 8 GB RAM, and running under Windows 7. The results are shown in Table 1. Inspecting the average number of sweeps until convergence (or until termination for Hirata et al., 2004), we observe that this number tends to decrease as L increases for both algorithms. At the same time, the mean execution time for Hirata et al. (2004) tends to increase with L. This is due to the increasing size of the reconstruction table with L. Such an execution time tendency does not clearly exist for LOGPE. The results show that LOGPE, run until convergence, is more computationally expensive than Hirata et al. (2004) for $L = 2$ to 9, but less expensive for $L = 10$ and 11. Based on our complexity analysis, these results, for varying L, are a function of both the average number of sweeps taken and the reconstruction table sizes of the two algorithms.

Table 2 shows the results for LOGPE initialized from the Hirata et al. (2004) solution at each L and run until convergence. The results show that in contrast to Table 1, the average number of sweeps until convergence for LOGPE does not decrease as L increases. Thus, since the reconstruction table size grows with L, the average execution time tends to increase with L, as seen in the table.

Comparing Tables 1 and 2 indicates that the choice of the initial symbol sequence affects the amount of LOGPE optimization needed to reach convergence. However, as noted earlier, it is not required to run LOGPE until convergence; one can instead use a stopping condition or a fixed computational allowance. Either way, the algorithm is guaranteed to achieve a lower discrepancy than Hirata et al. (2004), initialized from their solution.

3.3 Anomaly Detection for a Noisy Duffing System. In this section, we further demonstrate the efficacy of LOGPE applied to anomaly detection for a nonlinear system. Specifically, we consider the forced Duffing system described by the following differential equation (Thompson & Stewart, 2002):

Table 1: Number of Sweeps and Execution Time for the Hirata et al. (2004) and LOGPE Algorithms for the Ikeda Map, versus L, Starting from the Same Initial Symbol Sequence.

	Hirata et al. (2004) Algorithm				LOGPE Algorithm			
	Number of Sweeps		Elapsed Time (seconds)		Number of Sweeps		Elapsed time (seconds)	
L	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
2	12.2	3.3267	23.069	6.2690	21.7	2.8694	92.04	12.1326
3	10.8	2.2509	21.544	4.4833	20.2	3.8816	113.13	21.7555
4	7.6	1.9550	16.621	4.2426	10.9	5.2799	76.177	36.8633
5	9.1	1.3703	23.23	3.4928	11.4	1.9550	95.419	16.4225
6	8.3	3.0203	27.31	9.9146	10.2	4.4422	99.372	43.2100
7	10.5	2.7588	48.398	12.7043	8.3	2.4518	92.473	27.2323
8	5.4	2.1187	39.261	15.5563	8.3	2.4060	104.34	30.2010
9	4.8	1.3165	54.235	14.8155	8	1.9437	111.91	27.2085
10	4.4	0.8433	80.443	15.4208	4.4	1.1738	67.968	18.0970
11	4.8	1.5492	144.14	45.8944	5.6	2.0656	94.923	34.9614

Table 2: Number of Sweeps and Execution Time for LOGPE Applied to the Ikeda Map and Initialized, at Each L, from the Hirata et al. (2004) Solution.

LOGPE Algorithm				
	Number of Sweeps		Elapsed Time (seconds)	
L	Mean	Standard Deviation	Mean	Standard Deviation
2	15.1	2.1318	64.082	9.1104
3	16.4	6.9634	91.825	39.0256
4	14.7	9.2382	102.62	64.5755
5	15.7	4.5228	131.54	38.1445
6	18.1	4.7947	176.65	46.8828
7	22.1	7.0309	246.94	78.1921
8	19.4	4.4771	244.37	55.9821
9	21.1	7.2793	296.68	102.4597
10	17.1	6.1001	264.63	94.4140
11	16	3.9721	270.58	67.0298

$$\frac{d^2y}{dt^2} + \beta\frac{dy}{dt} + y(t) + y^3(t) = A\ \cos(\Omega t). \tag{3.1}$$

The steady state solution of equation 3.1 is especially affected by the value of β and the initial conditions $y(0)$ and $\dot{y}(0)$. Figure 7 shows the steady-state behavior of equation 3.1 for a given initial condition and different values of β. Each panel represents the phase plot at the given β shown at the top of the plot. We used $\Omega = 5$ and $A = 22$. As shown in this figure, a phase transition occurs between $\beta = 0.2$ and $\beta = 0.25$. For different initial conditions, this phase transition may occur at different values of β. We generated 90 realizations of equation 3.1 using 10 different initial conditions and 9 values of β. Therefore, for each initial condition, we have nine time series. Time series before phase transition are labeled class 0 (the nominal class), and those after phase transition are labeled class 1 (the anomalous class). Then all these time series are corrupted by additive noise. Figure 8 shows the phase plots of the time series from Figure 7 after being corrupted by additive noise. From this figure, the process of identifying whether a time series, after noise corruption, belongs to class 0 or class 1 appears to be a difficult challenge.

We applied LOGPE (Hirata et al., 2004), K-means (Duda et al., 2012), and maximum entropy partitioning (MEP; Rajagopalan & Ray, 2006) to symbolize each of the 90 time series. Then, for each symbol sequence a D-Markov machine of length $D = 2$ is constructed.[11] To explain further, suppose that

[11] A *probabilistic finite state automata* (PFSA) is a quadruple $K = (\mathcal{A}, Q, \delta, \pi)$, where $\mathcal{A}$ is a finite-cardinality alphabet, Q is a finite set of states, $\delta : Q \times \mathcal{A} \to Q$ is a transition function,

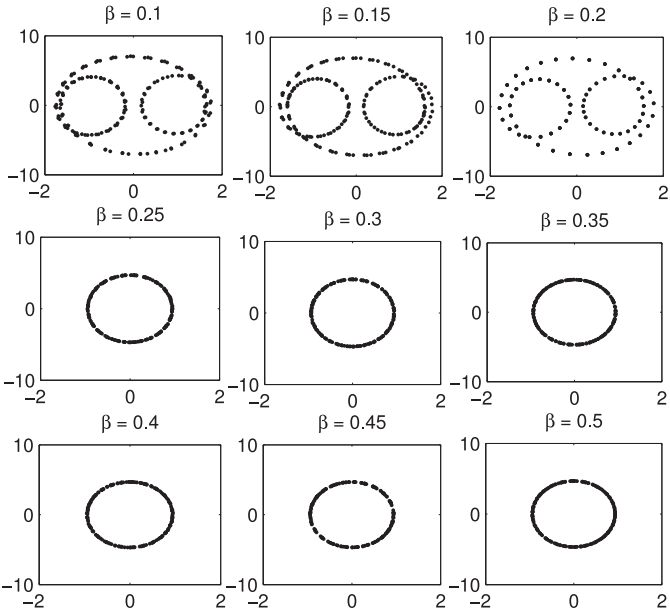

Figure 7: Phase plots of Duffing system for different values of β.

$\beta = 0.1$ is considered. First, the time-series clustering solution (LOGPE, Hirata, K-means, MEP) is learned on a nominal ($\beta = 0.1$) time series for a particular initial condition. Then this solution is used to symbolize (encode) all the other time series generated using the same initial condition with different values of β. Then the D-Markov machine constructed for each time series is used to generate the (steady-state) probability vector of the D-Markov machine's states for that time series. Hence, for each initial condition, there are nine such probability vectors, with one representing the nominal time series (in this case, with $\beta = 0.1$). Then the Kullback-Leibler divergence from each probability vector to its nominal reference point is found. If this divergence is less than a preset threshold the time series is classified as class 0; otherwise to class 1. The performances of LOGPE (Hirata et al., 2004), K-means, and MEP are evaluated by constructing the average receiver operating characteristic (ROC; Duda et al., 2012) curve (averaging over all the initial conditions) using each of these algorithms. The results for alphabet sizes 4 and 6 are shown in Figure 9. As shown in the figure, LOGPE

and $\pi : Q \times \mathcal{A} \to [0,\ 1]$ is a state transition probability matrix (Hopcroft, Motwani, & Ullman, 2001). A D-Markov machine, where $D \in \mathcal{N}$, is a PFSA corresponding to a stochastic symbolic stationary process for which the probability of the next symbol depends only on the previous (at most) D symbols (Ray, 2004).

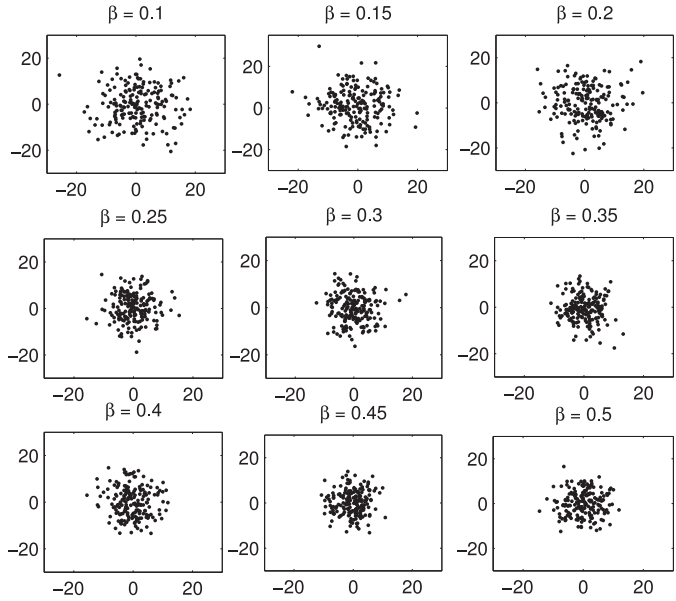

Figure 8: Phase plots of Duffing system with additive noise and different values of β.

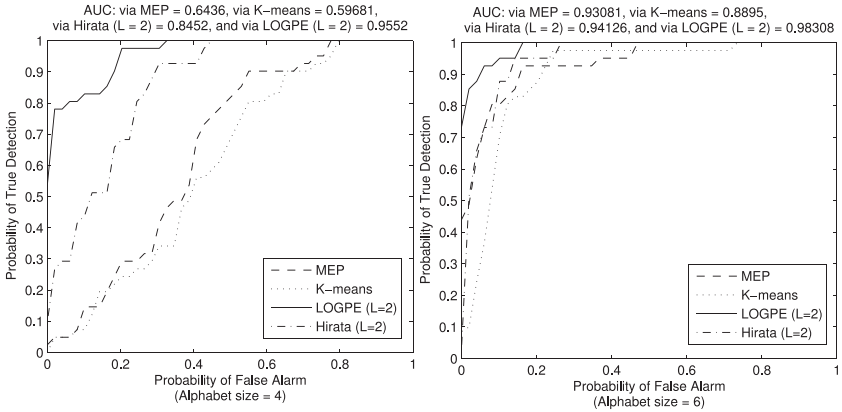

Figure 9: ROC performance for a noisy Duffing system.

performs very well, achieving average AUC = 0.9552 for alphabet size = 4 (much better than the other algorithms). LOGPE is further improved when the alphabet size is increased to 6, achieving AUC = 0.98308, which reflects excellent performance and better than the other algorithms. The phase

transition process in the Duffing system is an excellent example of anomalies occurring in nonlinear systems. In the following section, we consider a real example that frequently occurs in mechanical systems.

3.4 Application to Detection of Fatigue Failure in Mechanical Structures. Fatigue failure is one of the most frequent situations in which mechanical structures fail unpredictably. Modeling this type of failure has received great attention by many researchers (e.g., Gupta & Ray, 2007; Kwofie & Rahbar, 2012; and Aeran, Siriwardane, Mikkelsen, & Langen, 2017). In any mechanical structure, millions of initial materials' defects (such as dislocations, voids, inclusions, and slip bands) exist inside the microstructure even before the structure is used. In general, fatigue damage is critically dependent on these initial defects, from which cracks start to nucleate and merge together, generating bigger cracks, leading to catastrophic failure of the structure (Suresh, 2004). These microstructural initial defects are usually distributed in a highly random fashion, producing large uncertainties in the crack initiation and propagation process even under identical loading. Therefore, fatigue failure is considered an unpredictable and highly stochastic process.

Although structural fatigue damage is not a quantity that is easily measured directly, damage may be correlated with signals that can be measured and used for fatigue damage detection. In this work, we use ultrasonic signals that pass through a metallic specimen undergoing external cyclic loading. These signals are continuously recorded after they pass through the specimen's structure. When cracks occur, part of the signal will be reflected instead of being transmitted to the receiver through the structure, and hence the received signal is attenuated. The received signal keeps attenuating until the specimen breaks. Figure 10 shows such ultrasonic signals, after downsampling, for 16 sample specimens made of steel-aluminum alloy. As shown in the figure, the signal begins to significantly attenuate at a certain time instant. Such time instants roughly estimate phase transitions in fatigue damage, where cracks reach critical lengths and the damage process starts growing aggressively (until the specimen breaks and only a noise signal remains). Although all the specimens used in this work have the same dimensions and are made from the same material, the plots in Figure 10 show that the estimated phase transition instant is different from one specimen to another. This difference is due to initial microstructural defects that are specimen dependent. This situation is analogous to the Duffing system with different initial conditions, where the values of β at which phase transition occurs are sensitive to the initial conditions.

We used 24 specimens and obtained 24 ultrasonic signals (including the 16 signals in Figure 10). Each signal, consisting of 10,000 samples, is segmented into 10 disjoint time series, each with 1000 points. The nominal time series is the first one (for which the specimen has minimum damage). Time series that occur before the estimated phase transition time are labeled class

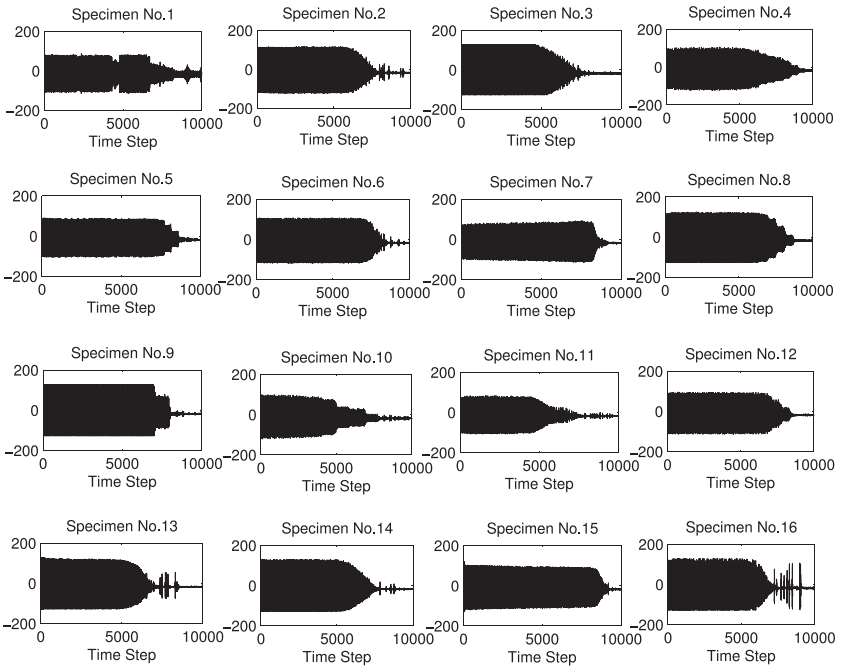

Figure 10: Ultrasonic signals for sample specimens.

0, and those that occur after that time are labeled class 1. Class 0 thus represents all the time series observed before the crack reaches a critical length (which can be visually observed), and class 1 represents all the time series after criticality.[12] Then all the time series are corrupted by additive noise.[13] We proceeded in a similar way as in the previous section to get the ROC performance (averaged over specimens) for LOGPE, K-means, and MEP. The results are shown in Figure 11. It is clear from the figure that LOGPE achieves good performance and better than K-means and MEP when the additive noise variance is one-tenth of the maximum amplitude of the signal (AUC $= 0.97137$). When the noise variance is increased to seven-tenths of the maximum amplitude of the signal, LOGPE achieves an AUC of 0.89232, still better than K-means and MEP.

The actual phase transition of fatigue damage in metals occurs before the received ultrasonic signal gets attenuated. The phase transition starts in the

[12] The signals shown in Figure 10 are in fact obtained after downsampling the original signals, which consist of millions of sample points.

[13] This is additional noise. The original signal is already corrupted by measurement noise as shown in Figure 10.

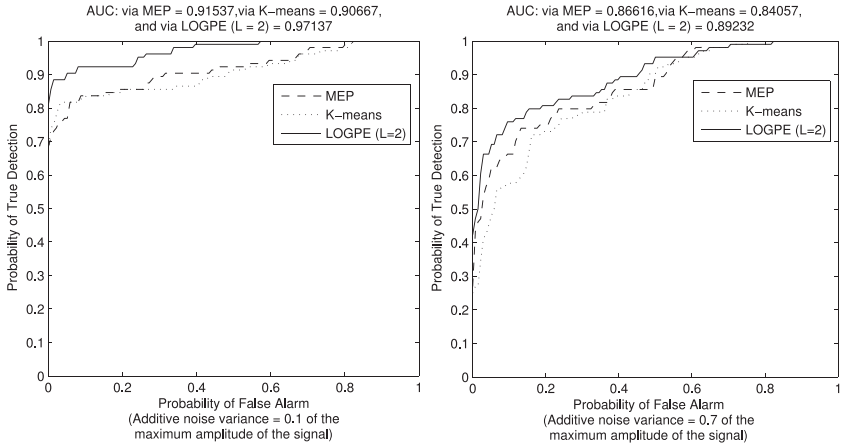

Figure 11: Average ROC performance for noisy ultrasonic signals from 24 specimens.

microscale in the metal's structure (Suresh, 2004). However, the phase transition at this early stage does cause a slight change in the behavior of the signal. The detection of such behavior is highly desirable in health monitoring and fatigue failure prediction of mechanical structures (Singh, Gupta, & Ray, 2009). In this experiment, we consider one of the above steel-aluminum alloy specimens, attempting to indicate early detection based on the ultrasonic signal. Unlike section 3.3, which considered a signal taken at a sampling rate of about 1.4 Hz, we consider the signal taken at a much higher rate—about 218 Hz. This signal, with 1,572,279 sample points, is shown in Figure 12a. We applied MEP, Hirata et al. (2004), and LOGPE, with the latter two initialized by MEP. The purpose is to evaluate whether Hirata et al. (2004) or LOGPE can achieve earlier detection of the fatigue damage phase transition than MEP. We divided the ultrasonic time series into 22 disjoint chunks, with each chunk about 71,467 sample points. The nominal time-series chunk is the first one (for which the specimen has minimum damage). Then MEP, Hirata et al. (2004), and LOGPE were learned on this chunk, with the resulting models used to symbolize all the subsequent chunks. After that, for each resulting symbol sequence, a D-Markov machine with $D = 2$ is constructed, and each D-Markov machine gives a (steady-state) probability vector of the machine's states. Then the anomaly measure for each time-series chunk is defined as the Euclidean distance between the probability vector of that time series and the probability vector of the nominal (reference) one. The results are shown in Figure 12b. As a physical fact, under a constant amplitude cyclic load, the damage in the microstructure grows slowly in its early stages, until cracks propagate and reach critical lengths, where a phase transition occurs and the damage rate rapidly increases,

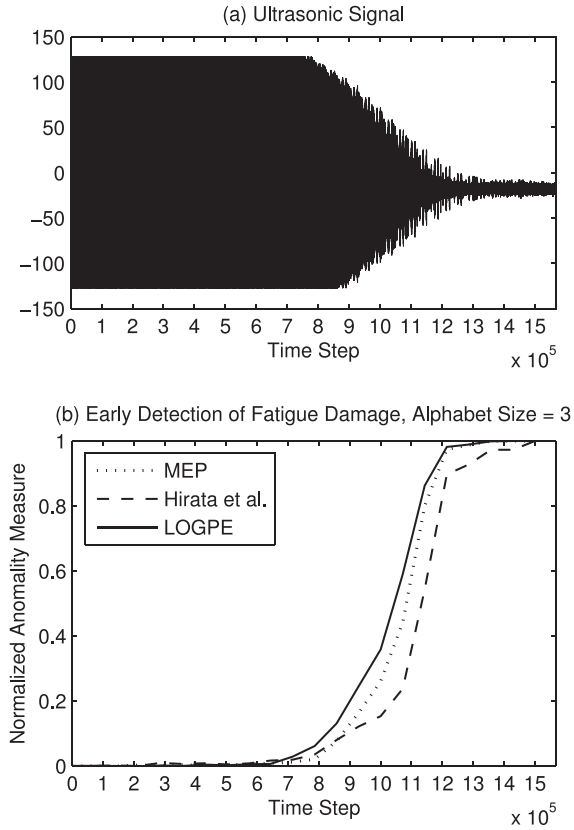

Figure 12: Anomaly detection for fatigue damage modeling.

resulting in a complete specimen failure (Suresh, 2004). Figure 12b shows that both MEP and LOGPE capture this behavior much better than Hirata et al. (2004) do. Furthermore, the figure shows that, in contrast to the curve given by Hirata et al. (2004), the curves given by MEP and LOGPE are in good agreement with the results available from the literature (Bogdanoff & Kozin, 1985; Jin, Gupta, Mukherjee, & Ray, 2011; Rege & Pavlou, 2017; Aeran et al., 2017). In fact, the curve given by Hirata et al. (2004) keeps increasing even after the $13.3 * 10^5$th time step, where only noise signal is left. Moreover, the Hirata et al. (2004) curve lags both MEP and LOGPE. On the other hand, LOGPE's curve leads the MEP curve. While we do not have ground truth on when the failure onset truly occurs, since the signal changes should only be due to fatigue damage, we expect that earlier detection is better than later detection. From the figure, MEP predicts fatigue failure at the $7.861 * 10^5$th sample point, where the damage rate (given by

the slope of the curve) starts a significant increase, while LOGPE predicts it at the $6.432 * 10^5$th sample point—142,900 sample points earlier than MEP. In terms of execution time for this big data experiment, Hirata et al. (2004) required 714 seconds while LOGPE required 11,861 seconds. The reason for this large difference is that Hirata et al. (2004) performs only one sweep in symbolizing a chunk, while for LOGPE, iterative sweeps were performed until convergence for each chunk (this can, of course, be reduced by using a stopping criterion or a fixed computational allowance). The gain for this increased execution time is much earlier (and presumably more accurate) detection of fatigue failure.

4 Conclusion and Future Work

The partition algorithm proposed in this letter addresses the issue of estimating a generating partition, from an observed time series, that is optimal in the sense of reconstructing the time series from the symbol sequence. The underlying concept has been validated and successfully applied to symbolization of time series generated by chaotic maps and to anomaly detection, achieving improvements over popular methods used in the literature. While the proposed algorithm ensures locally optimal solutions, simulated annealing or an extension of deterministic annealing (Rose, Gurewitz, & Fox, 1992) may find better solutions, albeit at the cost of increased computational complexity. A validation approach (e.g., applied to a (held-out) second half of the time series), or a cross-validation approach, could also be used to best choose the hyperparameters m, l, and $|\mathcal{A}|$ for symbolizing chaotic maps. Moreover, as a more theoretical direction, one could aim to achieve almost certain convergence results (as the length of the time series grows) for our LOGPE algorithm akin to those that have been obtained for K-means clustering (Pollard, 1981).[14]

We also note that the approach here does not build a generative stochastic model. Such a model would be appropriate, for example, for nonlinear dynamical systems observed in additive noise. LOGPE is in fact closely related to (generative) hidden Markov models that involve high-order

[14] One concern with the result in Pollard (1981), however, is that it assumes that the clustering algorithm finds the globally optimal solution, given a random data sample of a given size. Even for K-means, this assumption does not hold in practice. K-means is only guaranteed to find a locally optimal solution, even though it is based on two steps that are each globally optimal given the other held fixed: the centroid step, given the partition held fixed, and the nearest-neighbor partition step, given the centroids held fixed. LOGPE does not even have a globally optimal partition step. Such a step is computationally intractable, as discussed earlier in the letter. Thus, LOGPE uses a cyclical symbol assignment step that is guaranteed to descend in the clustering objective, but only to find locally optimal symbol assignments, given fixed centroids. Thus, we expect that LOGPE may be even less likely, in practice, to find globally optimal solution than K-means.

state transitions (depending on $l + m + 1$ states) and with gaussian state-conditional densities. However, it does not appear to be trivial to realize this generative extension of LOGPE since l may be positive, with the resulting model noncausal. By contrast, HMMs have a causal data generation mechanism. Developing a suitable stochastic model generalization of LOGPE, one that is valid for positive l, should be a good subject for a future study.

Acknowledgments

This work was partially supported by the U.S. Air Force Office of Scientific Research under DDDAS grants FA550-17-1-0070 and FA9550-15-1-0400.

References

Aeran, A., Siriwardane, S., Mikkelsen, O., & Langen, I. (2017). A new nonlinear fatigue damage model based only on S-N curve parameters. *International Journal of Fatigue*, *103*, 327–341.

Aghabozorgi, S., Shirkhorshidi, A., & Wah, T. (2015). Time-series clustering: A decade review. *Information Systems*, *53*, 16–38.

Ahmad, S., Alexander, L., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, *262*, 134–147.

Baragona, R., & Battaglia, F. (2007). Outliers detection in multivariate time series by independent component analysis. *Neural Computation*, *19*, 1962–1984.

Barron, A., Rissanen, J., & Yu, B. (1998). The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, *44*(6), 2743–2760.

Beck, C., & Schlogl, F. (1993). *Thermodynamics of chaotic systems: An introduction*. Cambridge: Cambridge University Press.

Bogdanoff, J., & Kozin, F. (1985). *Probabilistic models of cumulative damage*. New York: Wiley.

Buhl, M., & Kennel, M. (2005). Statistically relaxing to generating partitions for observed time-series data. *Phys. Rev. E*, *71*(4), 046213-1–046213-14.

Chamroukhi, F., Same, A., Govaert, G., & Aknin, P. (2009). Time series modeling by a regression approach based on a latent process. *Neural Networks*, *22*, 593–602.

Cornfeld, I., Fomin, S., & Sinai, Y. (1982). *Ergodic Theory*. Berlin: Springer-Verlag.

Davidchack, R. L., Lai, Y.-C., Bolt, E. M., & Dhamala, M. (2000). Estimating generating partitions of chaotic systems by unstable periodic orbits. *Phys. Rev. E*, *61*(2), 1353–1356.

Daw, C. S., & Finney, C. E. A. (2003). A review of symbolic analysis of experimental data. *Review of Scientific Instruments*, *74*, 915–930.

Daw, C. S., Kennel, M., Finney, C. E. A., & Connolly, F. T. (1998). Observing and modeling nonlinear dynamics in an internal combustion engine. *Phys. Rev. E*, *57*(3), 2811–2819.

Duda, R., Hart, P., & Stork, D. (2012). *Pattern classification* (2nd ed.). New York: Wiley.

Gersho, A., & Gray, R. (1993). *Vector quantization and signal compression*. New York: Kluwer.

Godelle, J., & Letellier, C. (2000). Symbolic sequence statistical analysis for free liquid jets. *Phys. Rev. E*, *62*(6), 7973–7981.

Gray, R. (1975). Sliding-block source coding. *IEEE Transactions on Information Theory*, *21*(4), 357–368.

Gray, R., Kieffer, J., & Linde, Y. (1980). Locally optimal block quantizer design. *Information and Control*, *45*, 178–198.

Gupta, S., & Ray, A. (2007). Symbolic dynamics filtering for data-driven pattern recognition. In E. A. Zoeller (Eds.), *Pattern recognition: Theory and applications* (pp. 17–71). Hauppauge, NY: Nova Science.

Hirata, Y., Judd, K., & Kilminster, D. (2004). Estimating a generating partition from observed time series: Symbolic shadowing. *Phys. Rev. E*, *70*(1), 016215-1–016215-11.

Hopcroft, J., Motwani, R., & Ullman, J. (2001). *Introduction to automata theory, languages, and computation* (2nd ed.). Reading, MA: Addison-Wesley.

Ikeda, K. (1979). Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Optics Communications*, *30*(2), 257–261.

Jin, X., Gupta, S., Mukherjee, K., & Ray, A. (2011). Wavelet-based feature extraction using probabilistic finite state automata for pattern recognition. *Pattern Recognition*, *44*, 1343–1356.

Kantz, H., & Schreiber, T. (2004). *Nonlinear time series analysis* (2nd ed.). Cambridge: Cambridge University Press.

Kennel, M., & Buhl, M. (2003). Estimating good discrete partitions from observed data: symbolic false nearest neighbors. *Phys. Rev. Letters*, *91*(8), 084102-1–084102-4.

Kennel, M., & Mees, A. (2000). Testing for general dynamical stationarity with a symbolic data compression technique. *Phys. Rev. E*, *61*(3), 2563–2568.

Kennel, M., Shlens, J., Abarbanel, H., & Chichilnisky, E. (2005). Estimating entropy rates with Bayesian confidence intervals. *Neural Computation*, *17*, 1531–1576.

Kwofie, S., & Rahbar, N. (2012). A fatigue driving stress approach to damage and life prediction under variable amplitude loading. *International Journal of Damage Mechanics*, *22*, 393–404.

Lind, D., & Marcus, B. (1995). *An introduction to symbolic dynamics and coding*. Cambridge: Cambridge University Press.

Linde, Y., Buzo, A., & Gray, R. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications*, *COM-28*(1), 84–95.

Miller, D., Ghalyan, N., & Ray, A. (2017). A locally optimal algorithm for estimating a generating partition from an observed time series. In *IEEE International Workshop on Machine Learning for Signal Processing*. Tokyo.

Mukherjee, K., & Ray, A. (2014). State splitting and merging in probabilistic finite state automata for signal representation and analysis. *Signal Processing*, *104*, 105–119.

Nguyen, H., McLachlan, G., Orban, P., Bellec, P., & Janke, A. (2017). Maximum pseudolikelihood estimation for model-based clustering of time series data. *Neural Computation*, *29*, 990–1020.

Petridis, V., & Kehagias, A. (1996). A recurrent network implementation of time series classification. *Neural Computation*, *8*, 357–372.

Pollard, D. (1981). Strong consistency of *k*-means clustering. *Annals of Statistics*, *9*(1), 135–140.

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

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

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

Rege, K., & Pavlou, D. (2017). A one-parameter nonlinear fatigue damage accumulation model. *International Journal of Fatigue*, *98*, 234–246.

Rose, K., Gurewitz, E., & Fox, G. (1992). Vector quantization by deterministic annealing. *IEEE Transactions on Information Theory*, *38*(4), 1249–1257.

Singh, D., Gupta, S., & Ray, A. (2009). In-situ fatigue damage monitoring using symbolic dynamic filtering of ultrasonic signals. *Proc. of the Institution of Mechanical Engineers: Part G: J. Aerospace Engineering*, *223*, 643–653.

Suresh, S. (2004). *Fatigue of materials* (2nd ed.). Cambridge: Cambridge University Press.

Thompson, J. M. T., & Stewart, H. B. (2002). *Nonlinear Dynamics and Chaos* (2nd ed.). New York: Wiley.

Willems, F., Shtarkov, Y., & Tjalkens, T. (1995). The context-tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, *41*(3), 653–664.

Wong, C., & Li, W. (2000). On a mixture autoregressive model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *62*(1), 95–115.

Received November 6, 2017; accepted March 5, 2018.