Iterated Function System Models in Data Analysis: Detection and Separation

Zachary Alexander, Elizabeth Bradley, Joshua Garland, and James D. Meiss

Department of Computer Science
University of Colorado
Boulder CO 80309-0430

Technical Report CU-CS-1087-11
October 2011

Iterated Function System Models in Data Analysis: Detection and Separation

Zachary Alexander,* Elizabeth Bradley,† Joshua Garland,‡ and James D. Meiss§
University of Colorado

We investigate the use of iterated function system (IFS) models for data analysis. An IFS is a discrete dynamical system in which each time step corresponds to the application of one of a finite collection of maps. The maps, which represent distinct dynamical regimes, may act in some pre-determined sequence or may be applied in random order. An algorithm is developed to detect the sequence of regime switches under the assumption of continuity. This method is tested on a simple IFS and applied to an experimental computer performance data set. This methodology has a wide range of potential uses: from change-point detection in time-series data, to the field of digital communications.

I. INTRODUCTION

Any approach to time series analysis begins with the question: is the data stochastic or deterministic [1, 2]? Often, the answer may be "both": the data could be generated by a deterministic system with a noisy component, perhaps due to measurement or computer round-off error. In this article, we propose an alternative possibility: the data could be generated by a sequence of deterministic dynamical systems selected by a switching process that itself could be deterministic or stochastic, i.e., by an *Iterated Function System* (IFS). For a detailed review of IFS dynamics, see [3]. If this were the case, then a goal is to identify the times at which switching between regimes occurs as well as the number and forms of the deterministic components themselves. Under the assumption that each deterministic system is continuous, we use topology to detect and separate the components of the IFS that are present in the output data. The main idea behind this approach is that the nearby state-space points can evolve in different ways, depending on the dynamical state of the IFS. Such a model has some relation to the determination of states in a hidden Markov model; however, such models are typically discrete and stochastic—not continuous and deterministic. A primary challenge in this problem is that overlap between the components could cause their trajectories to locally coincide. The use of IFS models for physical systems is not new; for example, Broomhead et al [4] used an IFS to model digital communication channels. In the current paper, we provide new tools to extract IFS models from experimental data and to determine the sequence of switching between regimes in the IFS.

We believe that method proposed here will prove useful in a number of applications. For example, detection and separation of IFS components is closely related to the statistical problem of *event* or *change-point detection* [5] where time-series data is assumed to come from a statistical distribution that changes suddenly. Applications where change-point detection plays a role include fraud detection in cellular systems, intrusion detection in computer networks, irregular-motion detection in computer vision, and fault detection in engineering systems, among many others [5]. Our underlying hypothesis is different—we assume that each regime is deterministic. For example, though change-point detection has been successfully applied to determine brain states from EEG data [6], EEGs have also been shown to exhibit properties of low-dimensional chaos [7]. Indeed, low-dimensional dynamics occurs in diverse areas including physiology, ecology, and economics [8–10]. We expect that the separation technique outlined below could be used to produce more accurate models of regime shifts and the effects of rapid parameter changes that occur, e.g. in the onset of seizures, natural disasters, or the bursting of economic bubbles.

II. DETECTION AND SEPARATION

Given a time series that corresponds to measurements of a dynamical system, our goal is to develop a technique that will detect whether the series is generated by an Iterated Function System (IFS). Formally, an IFS is a discrete-time dynamical system that consists of a finite set of maps $\{f_0, \ldots, f_n, \ldots, f_{N-1}\}$ on a state space X. A trajectory of the IFS is a sequence of state-space points, $\{x_0, \ldots, x_t, x_{t+1}, \ldots\}$, together with a *regime* sequence $\{n_0, \ldots, n_t, n_{t+1}, \ldots\}$ with $n_t \in \{0, 1, \ldots, N-1\}$, such that

$$x_{t+1} = f_{n_t}(x_t) \,, \quad \forall t \in \mathbb{N}.$$

Without loss of generality, we may assume that each map occurs at least once in the regime sequence, since otherwise the missing maps could be eliminated.

In the standard study of IFS dynamics, the regime sequence is often taken to be a realization of some random process [11, 12]; however, we only assume that we have access to a single trajectory that is generated by a particular realization. Consequently, the selection rule for the regime sequence is immaterial; indeed, it could just as well be a discrete, deterministic dynamical system. The standard theory, in addition, often requires that each f_n is a contraction mapping, in which case the IFS is *hyperbolic* and has a unique attractor A that is invariant in the

* Electronic Address: alexanz@colorado.edu
† Electronic Address: lizb@cs.colorado.edu
‡ Electronic Address: joshua.garland@colorado.edu
§ Electronic Address: jdm@colorado.edu

sense that $A = \bigcup_{n=0}^{N-1} f_n(A)$. We do not not need this assumption, and only assume that the trajectory lies in some bounded region of X.

We will assume that the time-series corresponds to T measurements on a particular state-space sequence,

$$\Gamma = \{x_0, x_1, \ldots, x_{T-1}\}; \tag{1}$$

but that the regime sequence is unmeasurable or hidden. For example, one may be able to measure the position of a forced pendulum at a sequence of times, but the pendulum may have a sealed brake mechanism that sets a friction coefficient and that is controlled externally to the experiment. Measurement of Γ also implicitly includes that of its associated shift map

$$\sigma(x_t) = x_{t+1}. \tag{2}$$

It is often the case that a time series corresponds to a limited measurement, perhaps of one function in multi-dimensional dynamical system. In this case, the first step is to use delay-coordinate embedding to construct, as much as is possible, a topologically faithful image of the orbit a reconstructed state-space [13]. We suppose that (1) is this embedded time series.

The fundamental goal in this paper is *detection and separation*: to detect if Γ is a trajectory of an IFS and to separate the regimes by recovering the sequence $\{n_t\}$. This problem is relatively straightforward when Γ is a subset of some *non-overlapping* region of the IFS, i.e., a region R such that $f_i(R) \bigcap f_j(R) = \emptyset$ for all $i \neq j$. In this paper, we address a more general situation in which Γ could be sampled from an overlapping region of the IFS.

A fundamental requirement for our separation method is that the maps f_n are continuous. In particular, the image of a connected set under each f_n must be connected. Since for finite data sets, the notion of connectivity makes no sense; we will instead use ϵ-connectivity under the assumption that X is a metric space with distance $d(x, y)$.

Definition (ϵ-connected [14]). *A set $\Omega \subset X$ is ϵ-connected if there exists an ϵ-chain connecting the points in Ω, i.e., for each point $x \in \Omega$ there exists a point $y \in \Omega$ such that $0 < d(x, y) \leq \epsilon$.*

Let $\mathcal{N}_k(x_t)$ denote the set consisting of x_t and its k-nearest neighbors in Γ. For each such set there will be a δ such that $\mathcal{N}_k(x_t)$ is δ-connected.

The simple idea of our algorithm is as follows. For each ϵ that is not too small, there must be a $k > 0$ such that such that the image of $\mathcal{N}_k(x_t)$ under a single map will be ϵ-connected. Indeed, continuity implies there is a δ so that a δ-connected set has an ϵ-connected image. For a given ϵ, the minimal δ, will be determined by the maximal distortion of the map. For the algorithm to work, the set Γ must be dense enough so that for this δ, there are nearest neighbors, i.e., $k > 0$.

If ϵ is chosen to reflect this maximal, single-map distortion, then whenever the time-shifted image, $\sigma(\mathcal{N}_k(x_t))$, consists of a number of ϵ-connected components, each component should reflect the action of a different f_n. This idea is expressed visually in Fig. 1. Note that $\sigma(\mathcal{N}_k(x_t))$ is NOT the same as $\mathcal{N}_k(x_{t+1})$, the set of nearest neighbors to the image of x_t.

To obtain reasonable results the parameter ϵ must be selected carefully as it will determine the maximal number of nearest neighbors, k. The number, N, of regimes of the IFS is not more than the maximal number of components of $\sigma(\mathcal{N}_k(x_t))$. However, since sparsely covered portions of the data set could result in spurious components, we will select N to be the number of components in the bulk of the images $\sigma(\mathcal{N}_k(x_t))$.

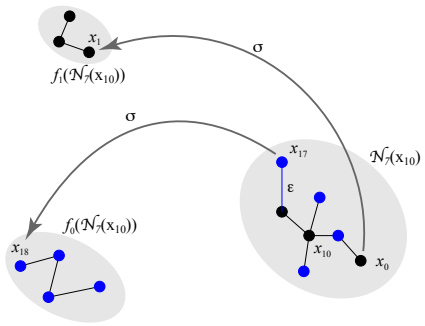

FIG. 1. Sketch of the action of the shift map σ on a 7-nearest neighborhood of a point $x_{10} \subset \Gamma$ results in two ϵ-connected components that can be identified as $f_0(\mathcal{N}_7(x_10))$ and $f_1(\mathcal{N}_7(x_10))$.

Given a time series Γ that we suspect to be generated by an IFS, the detection and separation algorithm requires an appropriate value for ϵ. Here we outline a possible algorithm.

- (Detection) Determine a value for ϵ by computing histograms of the separations between $\mathcal{N}_1(x_t)$ and $\sigma(\mathcal{N}_1(x_t))$. If Γ is sampled from a connected invariant set, then each of the nearest neighbor sets should be δ-connected. If there is more than one regime, their images should be disconnected for some choice of ϵ. The number of regimes N is estimated to be the number of components in the majority of the $\sigma(\mathcal{N}_k(x_t))$ and this should be persistent over a range of ϵ and k values.

- (Separation) Select a set of K-nearest-neighborhoods, $\{\Omega_j = \mathcal{N}_K(x_{t_j}) | j = 0, 1, \ldots J-1\}$, that overlap and cover Γ. Points are identified to be from a common regime if they lie in overlapping neighborhoods and their images lie in ϵ-connected components.

In the next section, we illustrate this method on a simple example.

III. EXAMPLE: A HÉNON IFS

As a simple example, consider the IFS generated by the two quadratic, planar diffeomorphisms

$$\begin{aligned} f_0\,(x,y) &= \left(y+1-1.4x^2,\, 0.3x\right), \\ f_1\,(x,y) &= \left(\, y+1-1.2(x-0.2)^2,\, -0.2x\right). \end{aligned} \tag{3}$$

The map f_0 is Hénon's quadratic map with the canonical choice of parameter values [15]; the map f_1 is conjugate, via an affine change of coordinates, to Hénon's map with parameters $(a,b) = (0.912, 0.2)$. We generate a single trajectory of this IFS by using a Bernoulli process with equal probability to generate a sequence $n_t \in \{0,1\}$. A trajectory with $T = 30,000$ points, shown in Fig. 2, has the appearance of two overlapping Hénon-like attractors. Note however, that since most points on Γ are not iterated more than a couple of consecutive steps with the same map, Γ is not just the union of the attractors of f_0 and f_1. Indeed the attractor for f_1 is simply a fixed point at $(0.63909, -0.1279)$.

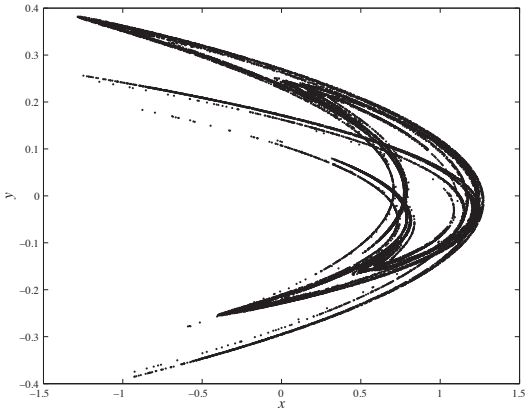

FIG. 2. A trajectory of the IFS generated by (3) with $T = 30,000$ points. Here $n_t \in \{0,1\}$ are chosen with equal probability.

To recover the regime sequence from Γ we must check for ϵ-disconnected images of δ-connected components. As a first step to determine an appropriate value for ϵ, we compute the distance between each point in Γ and its nearest neighbor, i.e., the diameter of $\mathcal{N}_1(x_t)$. This is shown in panel (a) of Fig. 3 as a histogram. Note that all but two points in Γ have a nearest neighbor within 0.02; and the vast majority within 0.002. Panel (b) of Fig. 3 indicates how these distances grow upon iteration: it shows the distance between the iterates of each of these nearest neighbors, i.e., the diameter of $\sigma(\mathcal{N}_1(x_t))$. There are now two distinct distributions separated by a gap $[0.02, 0.032]$. This suggests that the dynamics underlying Γ is discontinuous, and that a choice of ϵ in the gap may be appropriate.

Suppose that we did not know that the IFS (3) had two regimes—that only the trajectory Γ of Fig. 2 was available. To detect the number of regimes, we look at the

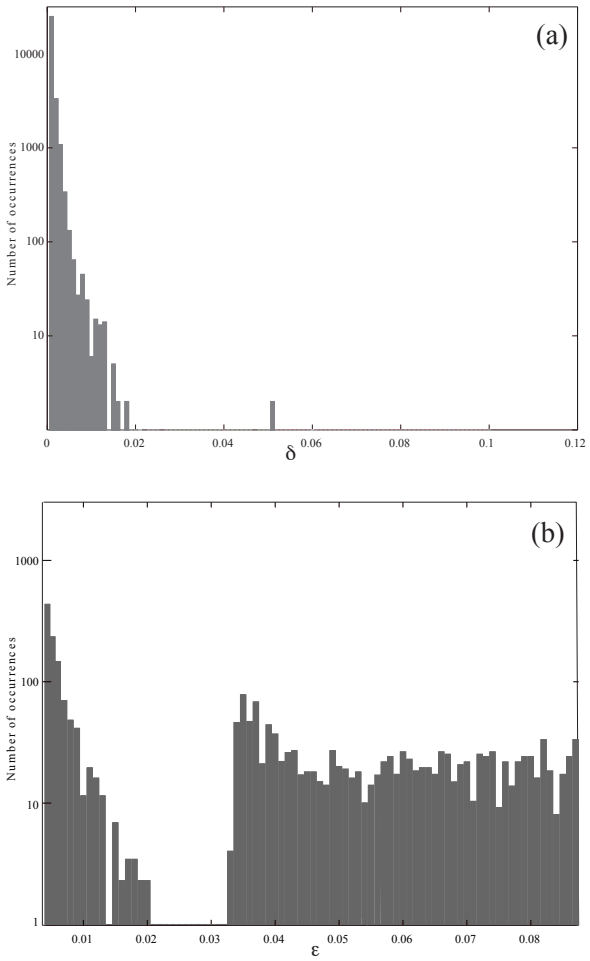

FIG. 3. Distance between each point on Γ of Fig. 2 and its nearest neighbor (a) and between the images of these two points (b).

number of ϵ-components in the image of the sets of five nearest-neighbors, $\mathcal{N}_5(x_t)$. Histograms of the number of ϵ-components of $\sigma(\mathcal{N}_5(x_t))$ are shown in Fig. 4 as ϵ varies from 0.005 to 0.05. The vast majority of these neighborhoods split into at most two ϵ-components. When ϵ is as small as 0.005 about 3% split into four or more components and when $\epsilon \geq 0.02$, only 0.3% split into three or more components. Note that with the equal probability rule that we used for (3), the probability that all five points in $\mathcal{N}_5(x_t)$ will be iterated with the same map is $\frac{2}{32}$, which is confirmed in Fig. 4, since about 6% of the images have one ϵ-component.

Thus in the detection phase of the algorithm, we confirm that the underlying dynamics has two regimes, $N = 2$ and obtain a reasonable choice, $\epsilon = 0.03$.

For the separation phase of the algorithm, we wish to classify which points on Γ are images of which map. To do this we choose larger, overlapping neigbhorhoods that cover Γ so that we can connect the subsets for each regime. To distribute these neigbhorhoods, more-or-less

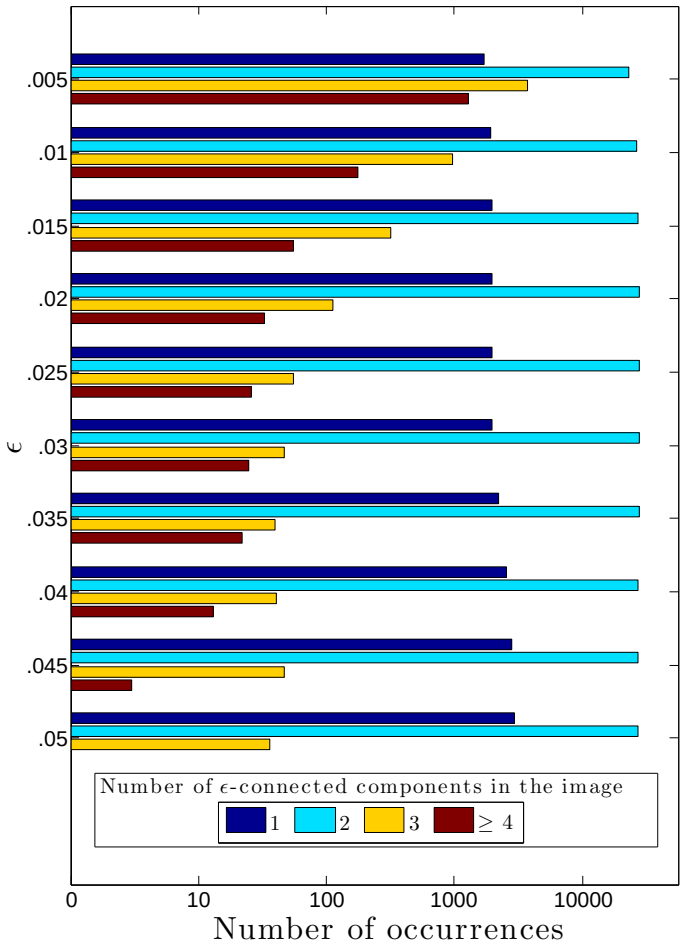

FIG. 4. Detection of the number of components of the images of sets of $K = 5$ nearest neighbors for the trajectory Γ of Fig. 2. The histograms show the number of the images $\{\sigma(\mathcal{N}_5(x_t)) \mid 0 \le t < 29,999\}$ that have N ϵ-connected components for various ϵ.

evenly over Γ, we select J points $\{y_0, y_2, \ldots, y_{J-1}\}$ by first choosing $y_0 \in \Gamma$ arbitrarily, and subsequently incrementing j and selecting y_j to be the point of Γ farthest from the previously selected points. Each selected point is the nexus of the K-nearest-neighborhood

$$\Omega_j = \mathcal{N}_K(y_j).$$

We choose $K = 40$ and $J = 10^4$, so that most of the Ω_j overlap with other neighborhoods, in the sense that they share points in Γ. In this case, each of the Ω_j is 0.03-connected.

The separation into regimes is accomplished as follows: whenever two "overlapping" Ω_j's have ϵ-connected image components that intersect, we identify them as being generated by the same f_n, see the sketch in Fig. 5. More specifically, suppose that $\Omega_{j,k}$ is the set of points in Ω_j that generate the k^{th} ϵ-component of $\sigma(\Omega_j)$. These are distinguished by Whenever $\Omega_{j_1,k_1} \cap \Omega_{j_2,k_2} \neq \emptyset$, then the union of their images $\sigma(\Omega_{j_1,k_1}) \cup \sigma(\Omega_{j_2,k_2})$ will share a point as well, and thus be ϵ-connected. In this case, the points in these images are selected as being generated by the same regime f_n, thus we set $k_1 = k_2 = n$.

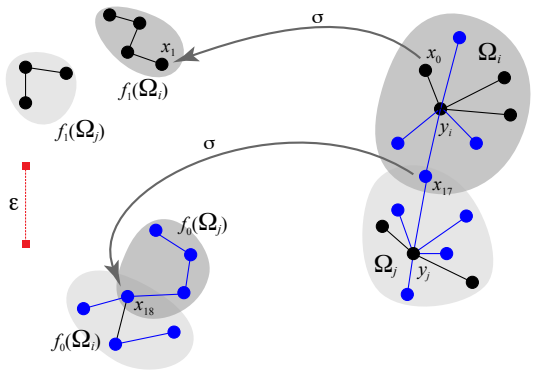

FIG. 5. Separation of the time series Γ into regimes. Here Ω_i and Ω_j represent the 7-nearest neighborhoods of y_i and y_j, respectively. They overlap, having x_{17} in common. Each of the neighborhoods has two ϵ-connected images under the shift σ. The pair that share $\sigma(x_{17}) = x_{18}$ are identified to be in the same regime, say $n = 0$ so the 7 points in this ϵ-connected image set (and their preimages) are colored blue to indicate the common regime.

The $\Omega_{j,k}$ can be thought of as nodes on an abstract graph. Whenever two of these neighborhoods share a point, an edge linking these nodes is added to the graph. Using this construction, the connected components of the resulting graph are selected as images of a fixed regime. Of course, we do not know which of the f_n's is associated with which graph component unless we have prior knowledge of some values of the functions.

For the trajectory of Fig. 2 and using the covering by the 10^4 neighborhoods Ω_j, this algorithm generates two large connected graph components, one contains $14,724$ points and the other $14,815$ points. These points are shown in the panels (a) and (b) of Fig. 6, respectively. Comparing these results with the known values of n_t, shows that every point in the first graph component has $n_t = 0$ and every point in the second has $n_t = 1$; that is, both the separation had *no false positives*. There are an additional 465 points of Γ that are not in these two graph components. These unidentified points represent sparsely visited regions of the trajectory.

It is no coincidence that the points identified to be images of f_0 in Fig. 6(a) appear to lie close to the attractor of the standard Hénon map, which is shown in light red in the figure. Note, however, that though the attractor for f_1 is a fixed point, the intersection of the red cross in the figure, the strong perturbation due to f_0 iterations causes the points in Fig. 6(b) to range far from its attractor.

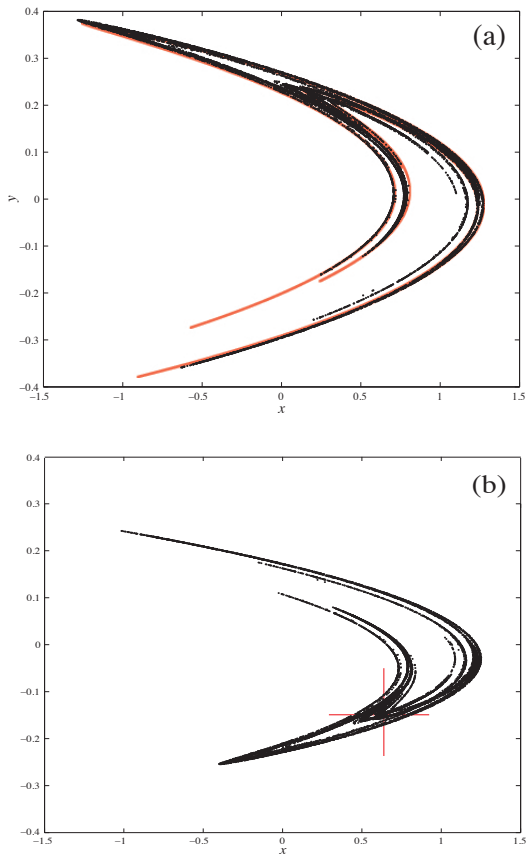

FIG. 6. Panel (a) (panel (b)) shows the 14,724 (14,815) points identified to reflect the action of f_0 (f_1). These points can be approximately interpreted as a sampling of $f_0(\Gamma)$ ($f_1(\Gamma)$) Also shown, in red, are points on the attractor of the Hénon map f_0, and a cross at the position of the fixed point of f_1.

IV. COMPUTER PERFORMANCE DYNAMICS

In this section, we describe the application of the regime separation algorithm to a time series obtained from an experimentally obtained a computer performance analysis data set. A critical performance bottleneck in modern computer systems occurs in the efficient management of memory. The cache is the level of memory closest to the processor; it is preloaded with the data a program *thinks* it will need. When the processor looks for a necessary piece of data in the cache and does not find it, it must load the data from main memory, resulting in a major performance slowdown. Such an event is called a *cache miss*.

The experiment to investigate the frequency of cache misses consists of looping over the simple C program:

```
for(i = 0; i < M; i++)
  for(j = i; j < M; j++)
    data[i][j] = 0;
```

running on an Intel Core2® processor. This code initializes the upper triangular portion of a matrix in row-major order. As the program runs, the hardware performance monitors built into the processor chip monitor the memory usage patterns—in particular, the rate of cache misses. The program is interrupted every 10^6 instructions and the number of cache misses that occurred over that interval is recorded. A segment of the resulting time series of $3(10)^4$ points, along with a two-dimensional time delay embedding, is shown in Fig. 7.

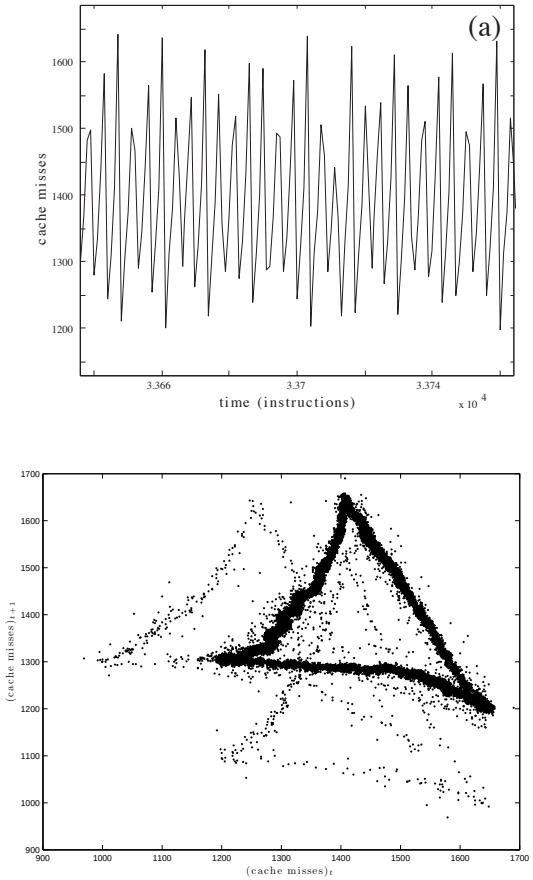

FIG. 7. (a) Cache misses per 10^5 instructions observed during the execution of *rowmajor*. (b) Two dimensional time delay embedding of the time series from Fig. 7(a)

This data set has been studied previously and shown to exhibit chaotic dynamics [16, 17].

The observation of the *ghost triangles* in Fig. 7(b)—seemingly reminiscent of three overlapping attractors from an IFS—prompted us to apply our regime separation algorithm to this data. Because the two ghosts are much more lightly sampled than the *main* triangle, our assumption was that the IFS consisted of three functions and that the switching process prioritized one of the three.

The histograms shown in Fig. 8 confirm that for the vast majority of points, at most three components are generated over for $\epsilon > 10$, thus setting $N = 3$ is reasonable.

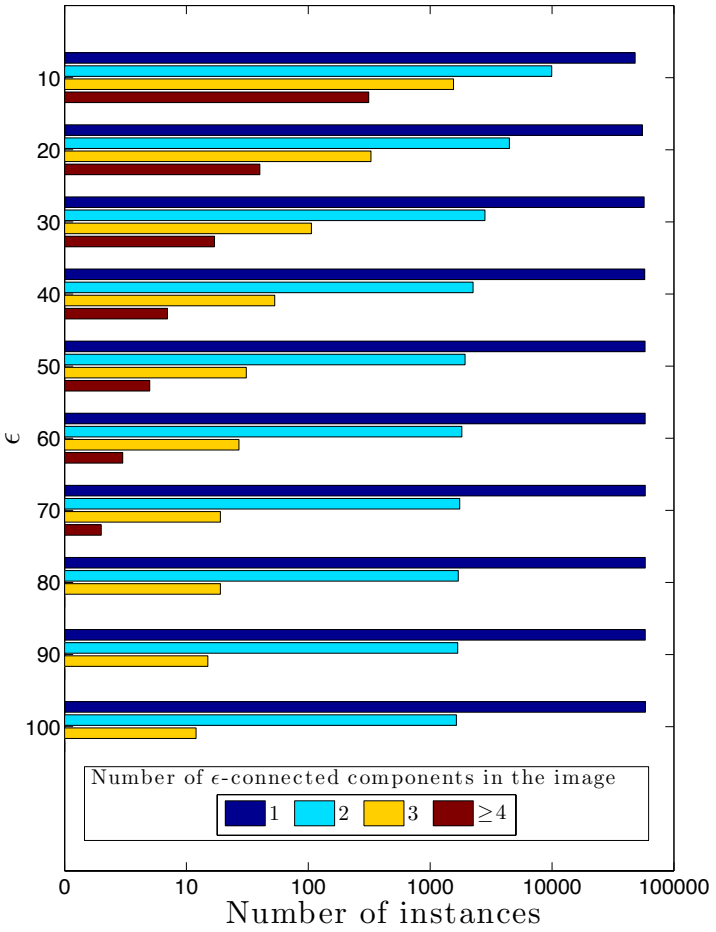

FIG. 8. A total of $6(10)^5$ groups of 5 nearest neighbors were chosen. Each cluster was iterated forward by the dynamics and the number of ϵ-connected components in the image were counted for various ϵ. The histogram shows that the number of clusters with two ϵ-connected components remains fairly constant, and for $\epsilon > 70$, the number of clusters with more than two ϵ-connected components levels off. This indicates that $\epsilon = 75$ is an appropriate parameter to catch the secondary dynamics while minimizing the number of false positives due to noise on the main attractor.

We chose $\epsilon = 75$ for the analysis, which is roughly $\frac{1}{3}$ of the apparent offset of each ghost. We counted the number of ϵ-connected components in the images of clusters of points for various ϵ and made a choice that allows us to identify a 'jump' to a ghost triangle, while ignoring relatively small noise vectors on the main triangle.

Points in the ghost were initially identified by selecting the component of the image of each ϵ-connected neighborhood that contained the smallest number of data points. The points thus identified had a strongly periodic component with period 215 so we narrowed our search to find points in a window of length 20 around this period. The set of points identified by the narrowed window is shown in Fig. 9(a). These points correspond to the lower ghost of Fig. 7(b); the second ghost is just an image of the first—a necessary result of the symmetry inherent in the time-delay embedding process.

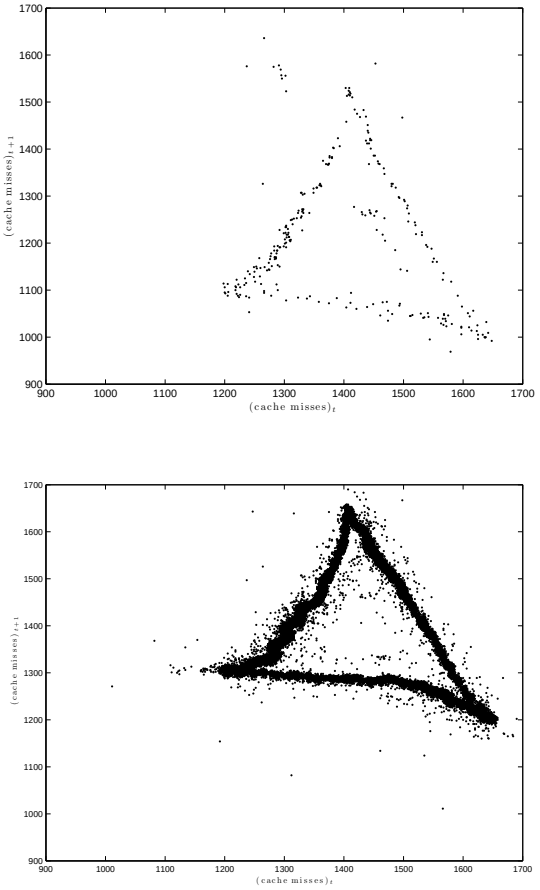

FIG. 9. (a) The lower ghost triangle separated from the data of Fig. 7(b). Each of the 277 points was identified as being ϵ-disconnected ($\epsilon = 75$) from the images of an ϵ-connected set of points. (b) A two dimensional time delay embedding of the *adjusted* time series obtained by adding 200 cache misses to the time series values corresponding to each of the points from (a).

This analysis revealed a direct correspondence between the points on the ghosts and points in the time series that are periodically spaced by 215 measurements—up to a very small error tolerance. Moreover, each ghost point appears to be shifted exactly 200 cache misses from the main triangle. Indeed, upon adding 200 cache misses to each of the ghost values, the full time series now has the embedding shown in Fig. 9(b). Thus, for this case, not only is the regime identified, but the dynamics of the two components is shown to be simply related: by just a shift.

There is one issue remaining before we can model this data as an IFS: we only have access to *measurements* of the states of the IFS. That is, if X is the state space of the computer system and $\{f_0, \ldots, f_k\}$ is a collection of maps on X, the observations correspond to the functions $\{h \circ f_0, \ldots, h \circ f_k\}$ with a continuous measurement function

$h : X \to \mathbb{R}$ that maps the state of the computer system to the number of cache misses that occur over the given time interval. It can be shown that the functions $h \circ f_i$ are sufficient for studying topological and geometric properties of the f_i—a more-detailed treatment of the function h can be found in [16].

Thus, if $h \circ f_0$ denotes the dynamics associated with the main triangle, we define $h \circ f_1(x) = h \circ f_0(x) - 200$, and $h \circ f_2(x) = h \circ f_0(x + 200)$. The IFS consists of the state space X, the collection $\{f_0, f_1, f_2\}$ of continuous maps, and the sequence $\{n_j\} \subset \{0, 1, 2\}$, where

$$n_j = \begin{cases} 1 & \text{if } j \equiv 0 \mod 215 \\ 2 & \text{if } j - 1 \equiv 0 \mod 215 \\ 0 & \text{otherwise} \end{cases}$$

This model rests on the assumption that f_1 and f_2 can be described completely in terms of f_0. To verify this assumption, we tested for determinism in the *adjusted* dynamical system of Fig. 9(a). We found that out of the 277 points so identified, only 17 fail to lie in an ϵ-connected image set in the adjusted dynamical system. Consequently, f_0 appears to be a continuous function and the IFS is an accurate model for this data set.

Much of the usefulness of this model comes from the fact that we have isolated the continuous function f_0. In light of this fact, it is reasonable to assume f_0 is representative of some low-dimensional dynamics that are present in the computer system, while f_1 and f_2 represent a secondary piece of dynamics—in this case, perhaps best described as 'deterministic additive noise'.

V. CONCLUSION

Many techniques for time-series analysis, such as in [18], explicitly require the time series to be generated by a continuous function, and almost all of them implicitly require that it be generated by a *single* function. For example, in [17], time-series analysis of the data of Fig. 7(a) showed that it has a positive Lyapunov exponent and fractal correlation dimension. However, our results show that that time series interleaves trajectories from different dynamical systems—a property that can trip up traditional time-series analysis techniques. In the data studied in [17], this proved not to be an issue because a single f_0 overwhelmingly dominated the dynamics. When that is not the case, problems can arise with traditional time-series analysis, which is often formulated assuming the existence of long, uninterrupted deterministic trajectories. Some techniques, such as those in [18], can be reimplemented using "snippets" of a trajectory. Using our topology-based approach, however, one could pull apart and study the dynamics of each of the f_i independently from the rest of the system.

In conclusion, we have described an algorithm for detection and separation of a signal that is generated by continuous, deterministic dynamics punctuated by regime shifts. The algorithm handles shifts that result from stochastic or deterministic processes—whenever the dynamics are described by an iterated function system. Time-series data from a computer performance analysis experiment were shown to fit this model. More generally, we claim that iterated function systems are a natural model for complex computer programs, which have regime shifts as they move from subroutine to subroutine. Furthermore, IFS models provide a natural framework for data analysis in a wide range of fields: whenever the physical system generating the data is prone to discontinuous regime shifts.

Another area in which the regime separation technique is particularly appropriate is digital communication channels. Indeed, (hyperbolic) iterated function systems are known to provide useful models of these channels [4]. A channel corresponds to an electrical circuit externally driven by a digital signal, and the discrete input signal corresponds to the regime sequence. Thus, the behavior of the circuit corresponds to a discrete set of continuous dynamical systems. A fundamental problem in this context is *channel equalization*, the reversal of distortion incurred by transmission through a channel. This is precisely the determination of the input signal sequence from a sequence of output values—i.e., regime separation. Channel equalization is straightforward for linear dynamics because the IFS attractors in these situations tend to be non-overlapping. However, more-realistic, nonlinear IFS models have overlapping attractors. We believe that our methods can be successfully used for channel equalization in this context.

Challenges that remain to be addressed include finding an efficient implementation for high-dimensional data, and dealing with systems that have traditional noise in addition to regime shifts. Furthermore, we have not addressed the nature of the switching process itself. Once the2 on of the regime shifts have been determined, the next natural question to ask is whether or not there is determinism present in the switching, and if so, if one can determine the rule for switching between regimes.

[1] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. Farmer, Physica D **58**, 77 (1992).

[2] G. Sugihara and R. May, Nature **344**, 734 (1990).

[3] P. Diaconis and D. Freedman, SIAM review , 45 (1999).

[4] D. Broomhead, J. Huke, M. Muldoon, and J. Stark, Proc. Roy. Soc. London. Series A **460**, 3123 (2004).

[5] Y. Kawahara and M. Sugiyama, in *Proceedings of 2009 SIAM International Conference on Data Mining (SDM2009)* (2009) pp. 389–400.

[6] A. Kaplan and S. Shishkin, Nonparametric Statistical Diagnosis: Problems and Methods , 333 (2000).

[7] W. Klonowski, W. Jernajczyk, K. Niedzielska, A. Rydz,

and R. Stepien, Acta Neurobiologiae Experimentalis **59**, 315 (1999).
[8] L. Glass and D. Kaplan, Medical Progress Through Technology **19**, 115 (1993).
[9] P. Turchin and A. Taylor, Ecology **73**, 289 (1992).
[10] P. Chen, System Dynamics Review **4**, 81 (1988).
[11] M. Barnsley, *Fractals Everywhere* (New York: Academic Press, 1988).
[12] K. Falconer, *Fractal Geometry: Mathematical Foundations and Applications* (John Wiley, 2003).
[13] T. Sauer, in *Time Series Prediction: Forecasting the Future and Understanding the Past*, edited by A. S. Weigend and N. A. Gershenfeld (Addison-Wesley, 1994).
[14] V. Robins, J. Meiss, and E. Bradley, Physica D **139**, 276 (2000).
[15] M. Hénon, Comm. Math. Phys. **50**, 69 (1976).
[16] Z. Alexander, T. Mytkowicz, A. Diwan, and E. Bradley, Adv. in Intelligent Data Analysis IX , 18 (2010).
[17] T. Mytkowicz, A. Diwan, and E. Bradley, Chaos **19**, 033124 (2009).
[18] K. Mischaikow, M. Mrozek, J. Reiss, and A. Szymczak, Physical Review Letters **82**, 1144 (1999).