Predicting High-Codimension Critical Transitions In Dynamical Systems Using Active Learning

Authors: Kelly Spendlove, Jesse Berwald, and Tomáš Gedeon

RESEARCH ARTICLE

Predicting High-Codimension Critical Transitions In Dynamical Systems Using Active Learning

Kelly Spendlove[a][*], Jesse Berwald[b] and Tomáš Gedeon[a]

[a]*Department of Mathematical Sciences, Montana State University,
Bozeman, MT 59718, USA*
[b]*Department of Mathematics, The College of William and Mary,
Williamsburg, VA 23185, USA*

Complex dynamical systems, from those appearing in physiology and ecology to Earth systems modeling, often experience critical transitions in their behavior due to potentially minute changes in their parameters. While the focus of much recent work, predicting such bifurcations is still notoriously difficult. We propose an active learning approach to the classification of parameter space of dynamical systems for which the codimension of bifurcations is high. Using elementary notions regarding the dynamics, in combination with the nearest neighbor algorithm and Conley index theory to classify the dynamics at a predefined scale, we are able to predict with high accuracy the boundaries between regions in parameter space that produce critical transitions.

1. Introduction

Critical scientific problems are increasingly marked by the need to understand the dynamical behavior of complex systems. Merely decomposing a problem into isolated subsystems, and attempting to extrapolate the behavior of the whole system from the steady state behavior of each subsystem is often insufficient. The importance of studying the dynamics of complex systems is exemplified by the wide range of fields in which such systems constitute major research foci [3, 19]. Examples of complex systems include scale-free networks in fields ranging from biology and neuroscience to computer science and internet research [7, 26]; analysis of high-dimensional models in ecology and physics [5, 11, 32]; and the study of atmospheric circulation ranging from small scale theoretical studies [20] to the macro scale models of global circulation. The need to study directly the entire aggregate system, instead of decomposing a system into its constituent parts is intuitively clear in climate change models: one is interested in the time-dependent trajectory of the entire planetary system over decades and centuries [18, 27, 29]. The mechanisms driving the climate are integrally coupled, making analysis via decomposition difficult or impossible [10, 14, 27, 28].

A key component in the long term behavior of complex dynamical systems is a sensitive dependence on the parameters chosen for the model, as well as the

---

initial conditions. Furthermore, for physical models the parameter values are often only known approximately, and the amount that they affect the system cannot be precisely determined. This introduces uncertainty in every step of the modeling process.

The classical dynamical system paradigm is to consider particular autonomous systems with one or two parameters, and study invariant sets which may be particularly sensitive to change in parameters. In the face of the aforementioned problems, this approach often falls short. Therefore, there is a need for conceptually new approaches to nonlinear dynamical systems with many parameters [18, 29].

One exciting new approach backs away from the traditional concept of an invariant set as the principle object of study for a dynamical system and focuses instead on the related concept of a *Morse set*. Morse sets have the advantage that they are robust with respect to perturbations, including certain types of noise, while also being computationally tractable [2, 23]. This robustness allows one to make statements about the qualitative behavior of a system over a range of parameters. Morse sets are also naturally ordered by the dynamics. The collection of Morse sets with the dynamics-compatible partial ordering is a *Morse decomposition*, and organizes the long-term behavior of the dynamics. Additionally, Morse decompositions have a natural combinatorial representation as a directed graph, where the vertices correspond to the Morse sets and the edges represent the dynamics-compatible ordering. This object is the *Morse graph* and is a central object of this approach to dynamics [2, 22].

This paper is organized as follows. In Section 2 we summarize the background necessary for constructing Morse graphs. We then show how the Morse graph approach can be used to construct a *database of dynamical systems* that can be queried in order to locate parameters at which global changes in dynamics occur. However, construction of the database at a reasonable grid size in high dimensional parameter spaces is computationally expensive and inherently suffers from the curse of dimensionality. In Section 3, we describe a novel machine learning approach to the construction of such a database that builds on the work of Arai, *et al.* [2, 4]. Sections 4 and 5 describe the model with which we demonstrate our algorithm as well as the learning algorithms used to build the database. In Section 6, we demonstrate that by incorporating our learning algorithms we are able to compute the Morse graphs of as few as 20% of the grid elements in a discretized parameter space, while correctly classifying the dynamics of over 98% of parameter space.

2.   Conley-Morse Database

In this section we provide a brief review on the fundamentals necessary to construct a database of global dynamics. The reader is referred to [2, 15] for further details.

2.1.   *Preliminaries*

Instrumental to this work is Conley index theory, an important tool for analyzing the dynamics of both continuous and discrete dynamical systems. In particular, the Conley Index can be used to find stationary, periodic or heteroclinic orbits, or to prove chaotic behavior of dynamical systems. More specifically, the Conley Index is an algebraic topological invariant of isolated invariant sets.

To simplify our exposition, we will focus primarily on the combinatorial aspects of the database, and refrain from speaking at length on algebraic topology. Furthermore, as our experimental model is a map, we will restrict our exposition to

the discrete case, primarily following the outline of [2]. Further details on Conley index theory can be found in [15, 23].

Let $f : X \times \Lambda \to X$ be a *map* on a locally compact metric space X and a parameter space Λ that is a compact, locally contractible, connected metric space. At a particular parameter value λ we will sometimes denote $f(x, \lambda)$ as $f_\lambda(x)$.

The fundamental mathematical structures in the study of dynamical systems are invariant sets. A subset $Z \subset X$ is *invariant* at the parameter $\lambda \in \Lambda$ if $f_\lambda(Z) = Z$. When considering computational dynamics it is important to note that we cannot compute at each parameter independently, and therefore are most interested in sets which are invariant with respect to a subset of the parameter space.

We use the notation $F : X \times \Lambda \to X \times \Lambda$ for the extension of f to include the parameters as explicit variables, i.e. $F(x, \lambda) = (f(x, \lambda), \lambda) = (f_\lambda(x), \lambda)$. For a subset of the parameter space $\Lambda_0 \subset \Lambda$, let $F_{\Lambda_0} : X \times \Lambda_0 \to X \times \Lambda_0$ denote the restriction of F to $X \times \Lambda_0$. For a set $S \subset X \times \Lambda$ we denote its restriction to Λ_0 by $S_{\Lambda_0} := S \cap (X \times \Lambda_0)$. We say a set $S \subset X \times \Lambda_0$ is *invariant over* Λ_0 if $F_{\Lambda_0}(S_{\Lambda_0}) = S_{\Lambda_0}$.

Within the database framework, the primary use of Conley index theory is to identify a finite collection of invariant sets which determine the global behavior of the dynamics within a particular compact set of parameter values. To this end, we will review two essential structures: isolating neighborhoods and isolated invariant sets.

Definition 2.1: An *isolating neighborhood* is a compact set $N \subset X \times \Lambda_0$ such that its maximal invariant set lies in its interior, i.e.

$$\text{Inv}(N, F_{\Lambda_0}) = \{x \in N \mid F_{\Lambda_0}^n(x) \in N \text{ for all } n \in \mathbb{Z}\} \subset \text{int}_{X \times \Lambda_0}(N),$$

where $\text{int}(\cdot)$ denotes the interior of a set. An invariant set S_{Λ_0} is an *isolated invariant set* if $S_{\Lambda_0} = \text{Inv}(N, F_{\Lambda_0})$ for some isolating neighborhood N.

The advantage of our focus on isolating neighborhoods is that they are robust to perturbations. This fact is captured in the notion of *continuation*, which states that given an isolating neighborhood, N, then N will also serve as an isolating neighborhood for a map with sufficiently close parameters [23]. For our purposes, the most important feature of isolating neighborhoods is that they are readily computable [9].

2.2. *Morse Decompositions*

We first introduce some elementary concepts from the theory of dynamical systems. A more in-depth treatment can be found in [17].

Definition 2.2: Consider $x_0 \in X$. The ω-limit set of x_0 is

$$\omega(x_0) = \omega(x_0, f) = \bigcap_{n \in \mathbb{N}} \text{cl}(\{f^k(x_0) \mid k \geq n\})$$

while the α-limit of x_0 is

$$\alpha(x_0) = \alpha(x_0, f) = \omega(x_0, f^{-1})$$

where $\text{cl}(\cdot)$ denotes set closure.

The following definition is central to the mathematical and computational aspects of this work. Morse decompositions provide a coarse, global description of the dynamics on S_{Λ_0}.

4

Definition 2.3: A *Morse decomposition* $M(S_{\Lambda_0})$ of S_{Λ_0} is a finite collection of disjoint isolated invariant subsets of S_{Λ_0} indexed by the set $\mathcal{P}_{\Lambda_0}$, i.e.

$$M(S_{\Lambda_0}) = \{M_{\Lambda_0}(p) \subset S_{\Lambda_0} \mid p \in \mathcal{P}_{\Lambda_0}\}$$

On the index set $\mathcal{P}_{\Lambda_0}$ there exists a strict partial order $>_{\Lambda_0}$ such that for all $(x, \lambda) \in S_{\Lambda_0} \setminus \bigcup_{p \in \mathcal{P}} M_{\Lambda_0}(p)$ there exist $p, q \in \mathcal{P}_{\Lambda_0}$, $p >_{\Lambda_0} q$ such that $\omega(x) \in M_{\Lambda_0}(q)$ and $\alpha(x) \in M_{\Lambda_0}(p)$. The sets $M_{\Lambda_0}(p)$ are known as *Morse sets.*

It is worth noting that while Morse sets may not be preserved under perturbation, the decomposition itself is robust in the sense that the decomposition and the associated isolating neighborhoods are preserved. In particular, Morse decompositions inherit the continuation property of isolating neighborhoods. From a computational viewpoint, since $\mathcal{P}_{\Lambda_0}$ is a partially ordered set, a Morse decomposition can be naturally represented as a *Morse graph*, a directed acyclic graph $\mathrm{MG}(\mathcal{F}_{\Lambda_0})$ with vertice $V = \mathcal{P}_{\Lambda_0}$ and edges $E = \{(p, q) \mid p >_{\Lambda_0} q\}$.

Morse graphs are the fundamental objects of the Conley-Morse database. Morse graphs provide a natural combinatorial representation for the dynamical system which can be compared in order to detect changes in the dynamics across regions in parameter space.
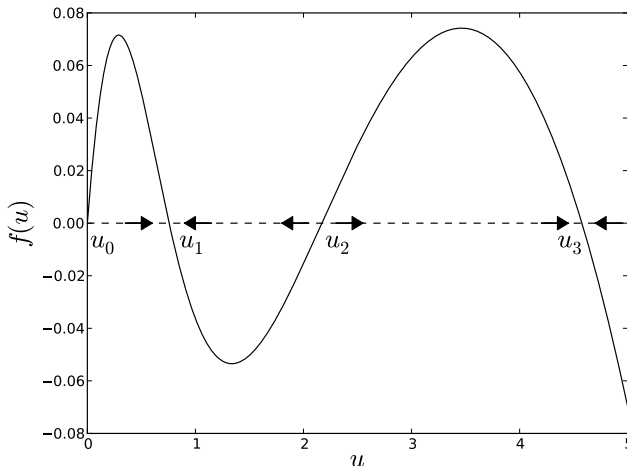

Figure 1.: Dynamics for choice of (q, r) exhibiting three steady states.

2.3. *An Example*

To motivate our methods, we will illustrate some aspects of our approach on the following model of spruce budworm populations, originally introduced by Ludwig [21]. The model is a classical example of insect outbreak as spruce budworm can defoliate the balsam fir, leading to potentially major environmental issues in eastern North America. We follow the exposition of Murray [24], with the budworm population dynamics given by the following equation:

$$\frac{dN}{dt} = r_B N \left(1 - \frac{N}{K_B}\right) - \frac{BN^2}{A^2 + N^2}$$

Here, r_B is the linear birth rate of the budworm and K_B is the carrying capacity of the environment (associated with the density of foliage available on the trees). The

predation term, $\frac{BN^2}{A^2+N^2}$, is that originally suggested by Ludwig for its qualitative behavior (predation saturates for large enough N).

We consider the nondimensional quantities:

$$u = \frac{N}{A}, r = \frac{Ar_B}{B}, q = \frac{K_B}{A}, \tau = \frac{Bt}{A}$$

which leads to the following nondimensionalized equation:

$$\frac{du}{d\tau} = ru\left(1 - \frac{u}{q}\right) - \frac{u^2}{1+u^2} = f(u, (r, q))$$

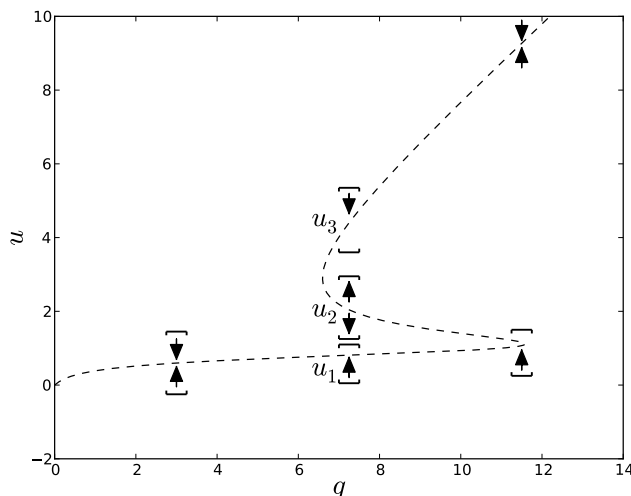The system now has only two parameters, r and q, which are dimensionless quantities.

Figure 2.: Isolating neighborhoods for equilibria u_1, u_2, u_3 from Figure 1 as a function of parameter q and a fixed r. The isolating neighborhood for u_0 is omitted for clarity.

The steady states of the system corresponds to the solutions of

$$f(u, (r, q)) = 0$$

A conventional dynamical analysis reveals that either one or three solutions exist, depending upon the parameters, r and q [24]. For instance, for appropriate values of r and q, the phase portrait has three equilibria as shown in Figure 1.

The smaller stable equilibrium, u_1, is the *refuge* equilibrium while the stable equilibrium, u_3, is the *outbreak equilibrium*. The unstable equilibrium at the origin, u_0, corresponds to *extinction*.

While equilibria in this particular example can be computed analytically, more complicated invariant sets in high dimensional spaces are, in general, not accessible to analytic methods. Furthermore, small perturbations can lead to a cascade of changes in the internal structure of these sets, while the overall shape may remain the same. Therefore we focus on isolated invariant sets, which are by definition stable under perturbations and their associated isolating neighborhoods are readily computable. In our simple example where the phase space is one-dimensional, isolating neighborhoods are intervals whose boundary does not intersect an invariant set, i.e. an equilibrium. In Figure 2, we depict a one-parameter family of phase

portraits parameterized by q, for a fixed $r = 0.535$. The curve is the set of zeros for $f(u, (r, q))$. The intervals represent isolating neighborhoods at a given value of r and q with non-trivial isolated invariant sets which in this case are equilibria. Similar intervals isolate the equilibrium at $u_0 = 0$ which we chose not to depict for the sake of clarity.

We now concentrate on the value of $q = 7.5$. At this value there are three isolated invariant sets. The flow between the neighborhoods that isolate u_0, u_1, u_2 and u_3 induce a flow defined partial order on these neighborhoods. Isolated invariant sets, together with this partial order define a Morse decomposition of the invariant set $M(S_q)$ at $q = 7.5$. This information is represented combinatorially in terms of a Morse graph, depicted in Figure 3.

The classical bifurcation theory shows that the budworm model exhibits *hysteresis* which is a basis for predictions of outbreak dynamics [24]. Figure 4a depicts the set of equilibria as a function of both parameters, and the Figure 4b shows the parameter region where three non-zero equilibria coexist. In the language of Morse decompositions, in the shaded region the Morse decomposition is that depicted in Figure 3, while outside of this region in consists of two isolated invariant sets, one corresponding to u_0 and the other to a stable equilibrium.
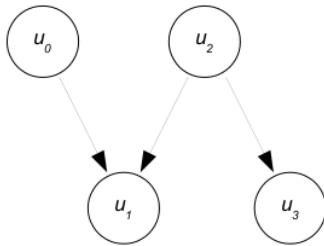

Figure 3.: Morse graph for Morse decomposition for $M(S_q)$, $q = 7.5$, see Figure 2. (u_0 has been included.)

This example shows a key role of classifying regions of parameter space into different types of dynamics. Such an analysis can form a basis for interventions and control of the system to a desired regime. The database approach that we use in this paper, and describe in following sections, aims to compute a partitioning of the parameter space into regions with identical Morse decompositions. In particular, we emphasize that our techniques are amenable to computation. The emphasis on computability is essential as analytical techniques are often of limited use in large size problems, found in real world applications.

2.4.   *Combinatorial Dynamics*

In this section, we review precisely how Morse graphs are computed. From [2], a *grid* on a metric space Z is a collection $\mathcal{Z}$ nonempty compact subsets of Z with the following properties

(1) $Z = \bigcup_{G \in \mathcal{Z}} G$
(2) $G = \mathrm{cl}(\mathrm{int}(G))$ for all $G \in \mathcal{Z}$
(3) $G \cap \mathrm{int}(H) = \emptyset$ for all $G \neq H \in \mathcal{Z}$
(4) If $Y \subset Z$ is compact, then $\{G \in \mathcal{Z} \mid G \cap Y \neq \emptyset\}$ is finite

In the context of a grid, we define the support function $|\cdot|$ from subsets of $\mathcal{Z}$ to subsets of Z by $|\mathcal{A}| := \bigcup_{A \in \mathcal{A}} A$. Further, given $Y \subset Z$, we define

$$\mathcal{Z}(Y) := \{G \in \ \mid \mathrm{int}(G) \cap Y \neq \emptyset\}.$$

For the rest of the paper we denote by $\mathcal{X}$ and $\mathcal{Q}$ the grids on phase space and parameter space, respectively.

We now discuss a method for exploring the dynamics of f computationally. A *combinatorial multi-valued map* $\mathcal{F} : \mathcal{X} \rightrightarrows \mathcal{X}$ assigns to each element $G \in \mathcal{X}$ a finite (and possibly empty) subset of $\mathcal{X}$ as follows. It is defined as follows. Consider $f(|G|)$ and its *combinatorial enclosure* formed by the collection of all grid elements $H \in \mathcal{X}$ such that $|H| \cap f(|G|) \neq \emptyset$. Then the multivalued map $\mathcal{F}$ is defined as

$$\mathcal{F}(G) := \{H \in \mathcal{X} \mid |H| \cap f(|G|)\}.$$

Furthermore, the combinatorial enclosure gives an *outer approximation* of f by defining

$$|\mathcal{F}|(x) := \bigcup_{G \in \mathcal{X} \,:\, x \in G} |\mathcal{F}(G)|.$$

This provides rigorous bounds on the dynamics of f. An essential component of this approach is the use of interval arithmetic software to construct $\mathcal{F}$, providing a rigorous bound on any approximation error as well [34]. Efficient algorithms exist to compute isolating neighborhoods and Conley indices from the combinatorial enclosure $\mathcal{F}$. Furthermore,, $\mathcal{F}$ is represented as a graph, opening up its analysis to efficient graph algorithms as well [15].

In order to construct Morse graphs, we first start with the recurrent set of $\mathcal{F}_Q$, defined by

$$\mathcal{R}_Q := \{G \in S_Q \mid \text{there exists a nontrivial path from } G \text{ to } G \text{ in } S_Q\}$$

The recurrent set $\mathcal{R}_Q$ can be partitioned into equivalence classes $\{\mathcal{M}_Q(p) \mid p \in \mathcal{P}_Q\}$ called combinatorial Morse sets by the equivalence relation $G \simeq H$ if and only if there exists a path in $\mathcal{F}_Q$ from G to H and a path in $\mathcal{F}_Q$ from H to G. This gives a strict partial order on the indexing set $\mathcal{P}_Q$ by setting $p >_Q q$ if there exists $G \in \mathcal{M}_Q(p)$, $H \in \mathcal{M}_Q(q)$ and a path from G to H in $\mathcal{F}_Q$.

By construction, the combinatorial Morse decomposition can be represented by a directed graph. We define the Morse graph $\mathrm{MG}(\mathcal{F}_Q)$ to be the acylic directed graph with vertices consisting of the elements of $\mathcal{P}_Q$ and the minimal set of directed edges (p, q) which generate $p >_Q q$ under transitivity.

2.5.   *Comparing Morse Graphs*

We now review how to use Morse graphs to compare dynamics over different regions in parameter space. Suppose that for each Q in $\mathcal{Q}$ we have computed $\mathrm{MG}(\mathcal{F}_Q)$. In order to compare each MG, we define the *clutching graph* and *clutching function*.

Definition 2.4:   Consider $Q_0, Q_1 \in \mathcal{Q}$ where $Q_0 \cap Q_1 \neq \emptyset$. The clutching graph $\mathcal{J}(Q_0, Q_1)$ is the bipartite graph with vertices $\mathcal{P}_{Q_0} \cup \mathcal{P}_{Q_1}$ and edges $(p, q) \in E$ if $\mathcal{M}_{Q_0}(p) \cap \mathcal{M}_{Q_1}(q) \neq \emptyset$.

If every vertex in P_{Q_0} in the clutching graph $\mathcal{J}(Q_0, Q_1)$ has a unique edge, then we define the *clutching function* $\iota : \mathcal{P}_{Q_0} \to \mathcal{P}_{Q_1}$ by $\iota(p) := q$ for each edge (p, q) of $\mathcal{J}(Q_0, Q_1)$.

Consider $Q_0, Q_1 \in \mathcal{Q}$ and their corresponding Morse graphs $\mathrm{MG}(\mathcal{F}_{Q_0})$, $\mathrm{MG}(\mathcal{F}_{Q_1})$, where $Q_0 \cap Q_1 \neq \emptyset$. If the clutching function ι is defined for the clutching graph $\mathcal{J}(Q_0, Q_1)$, and ι gives a directed graph isomorphism from $\mathrm{MG}(\mathcal{F}_{Q_0})$ to $\mathrm{MG}(\mathcal{F}_{Q_1})$, then we consider $\mathrm{MG}(\mathcal{F}_{Q_0})$ and $\mathrm{MG}(\mathcal{F}_{Q_1})$ to be *equivalent*. As in Arai *et al.* [2],

we say that the equivalence classes of $\{\mathrm{MG}(\mathcal{F}_Q) \mid Q \in \mathcal{Q}\}$ with respect to the transitive closure of this relation are *continuation classes*.

The purpose for requiring that ι define a directed graph isomorphism as opposed to merely a bijection is that differences in the partial order may indicate differences in the dynamics. In [2] it is shown that if $\mathrm{MG}(\mathcal{F}_{Q_0})$ and $\mathrm{MG}(\mathcal{F}_{Q_1})$ belong to the same continuation class then there is a path in parameter space along which the underlying Morse decompositions are related by continuation. However, compared to classical notions of equivalence, two Morse graphs belonging to the same continuation class is a relatively weak form of equivalence. In general, this is due to investigating the dynamics at the level of grid elements. We refer the reader to [2] for further elaboration.

The position we take in this paper is that continuation classes are a significant and useful notion for constructing a database of global dynamics. In other words, the database consists of a combinatorial representation of the set of continuation classes, and their relative connectivity. This concept is formalized in an efficient manner in the following definition [2].

Definition 2.5: The continuation graph of $\mathcal{F}$ is a graph whose vertices are the continuation classes $\{(\mathrm{MG}(j), \mathcal{Q}(j)) \mid j = 1, \ldots, J\}$ where $\mathcal{Q}(k) \subset \mathcal{Q}$ is the set of parameter boxes associated with the k-th continuation class, and $\mathrm{MG}(k) = \mathrm{MG}(\mathcal{F}_Q)$ for some $Q \in \mathcal{Q}(k)$.

2.6.   *Conley-Morse Database*

We will now summarize the process to construct the complete database for a given dynamical system. The first step is to choose the resolution for the grids $\mathcal{X}$ and $\mathcal{Q}$, on the phase and parameter spaces, respectively. The next step is to construct the combinatorial multi-valued map $\mathcal{F}$ on $\mathcal{X} \times \mathcal{Q}$. Then the Morse graph $\mathrm{MG}(\mathcal{F}_Q)$ must be determined for each $Q \in \mathcal{Q}$. For each pair of adjacent grid elements, Q_0, Q_1 in the parameter space, the clutching graph $\mathcal{J}(Q_0, Q_1)$ must then be formed, with the
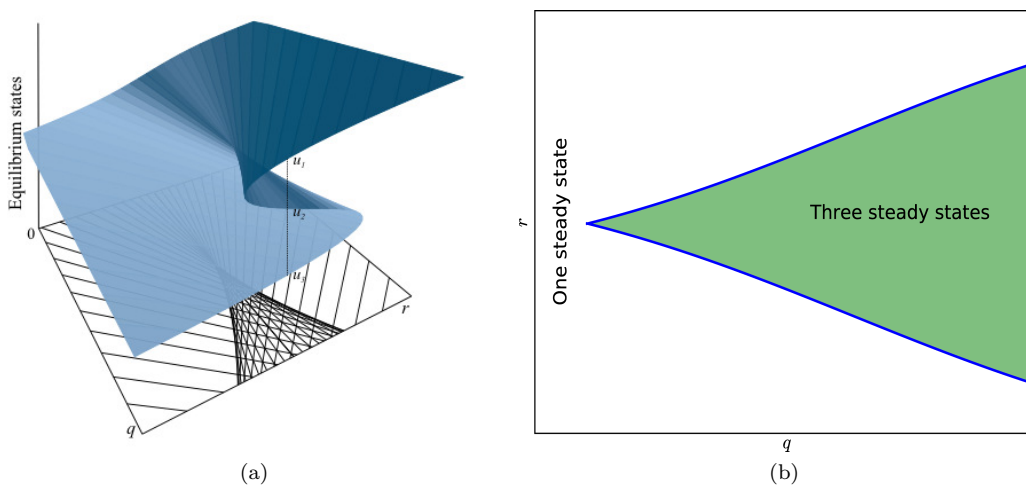

Figure 4.: (a) Equilibria states in a general cusp catastrophe model. In the region of the fold, three equilibrium states exist. One equilibrium exists in other regions. The dashed vertical line shows the arrangement of the equilibria in phase space for an arbitrary pair of parameters (q, r) in the cusp region. The repeller and attractors in the Morse graph in Figure 3 correspond to u_1, u_2, and u_3. (b) Parameter space showing the projection of the folded region on the right.

clutching function ι tested to determine if it defines a directed graph isomorphism.

Unfortunately, exhaustively computing upon parameter space in this fashion is very computationally demanding. In fact, the computational burden of producing the Conley-Morse database has inspired previous machine learning approaches [4]. We expand upon this work by introducing novel machine learning algorithms and ideas for approximating the database computation.

Now that we have introduced the Conley-Morse database, recall that in this paper our focus is on predicting the locations of potential bifurcations. In the setting of the database, this corresponds to discerning the edges between regions of grid elements which belong to the same continuation class.

3.   Learning the Conley-Morse Database

As we want to refrain from computing the entire parameter space, we do not form the clutching graph to check the pairwise intersection of Morse sets in the phase space. Thus, we cannot assign the correct continuation classes for intersecting grid elements in parameter space. Instead, we define two grid elements to belong to the same equivalence class if there exists a directed graph isomorphism between their corresponding Morse graphs.

Definition 3.1: Let G and H be directed graphs. A directed graph isomorphism between G and H is a bijection h that maps $V(G)$ to $V(H)$ and $E(G)$ to $E(H)$ such that each $(u,v) \in E(G)$ is mapped to $(h(u), h(v)) \in E(H)$.

For $Q_0, Q_1 \in \mathcal{Q}$, we define $\mathrm{MG}(\mathcal{F}_{Q_0}) \sim \mathrm{MG}(\mathcal{F}_{Q_1})$ if there exists a directed graph isomorphism between them. As a result of our changes to the Conley-Morse database framework, our equivalence classes are coarser than continuation classes. Nevertheless, we show that our algorithms still detect a significant portion of phase transitions.

4.   Leslie Model

We illustrate our approach on a nonlinear Leslie population model. This model and its relevance to population biology is discussed further in [35]. In the Leslie model, the population is partitioned into d generations, each with population $x_1, \ldots, x_d$, and an associated reproduction rate. The nonlinearity stems from the assumption that fertility decreases exponentially with the total size of the population.

For our experiments we consider the two generation Leslie model with a three dimensional parameter space $\Lambda := \{(\theta_1, \theta_2, p) \in [8, 37] \times [3, 50] \times [0.5, 0.9]\}$, given by:

$$(x, \lambda) = \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} \theta_1 \\ \theta_2 \\ p \end{bmatrix} \right) \mapsto f(x, \lambda) = \begin{bmatrix} (\theta_1 x_1 + \theta_2 x_2) e^{-0.1(x_1 + x_2)} \\ p x_1 \end{bmatrix}$$

Mirroring [2], we select a $80 \times 80 \times 40$ grid for the three-dimensional parameter space, which we label traditionally as x, y, and z axes, respectively. Since in [2] they perform performed an exhaustive computation on this subset of parameter space, we are able to compare our weaker notion of equivalence with the result provided by their continuation classes. Fixing z, we obtain a brute-force, or 'ground truth', result without the use of learning algorithms, but with our notion of weaker equivalence. We compare this with the continuation class results from [2] in the

continuation diagram in Figure 5. In the continuation diagram, regions of parameter space which are same color correspond to an continuation or equivalence class. For instance, one difference between our two notions of equivalence can be seen upper left region in Figure 5b, which is detected to be of the same class as other regions of the same color, whereas in Figure 5a the same region is distinguished from the others. However, in all of these figures the colors are arbitrary, thus the difference in color between Figure 5a and 5b is not representative of anything.

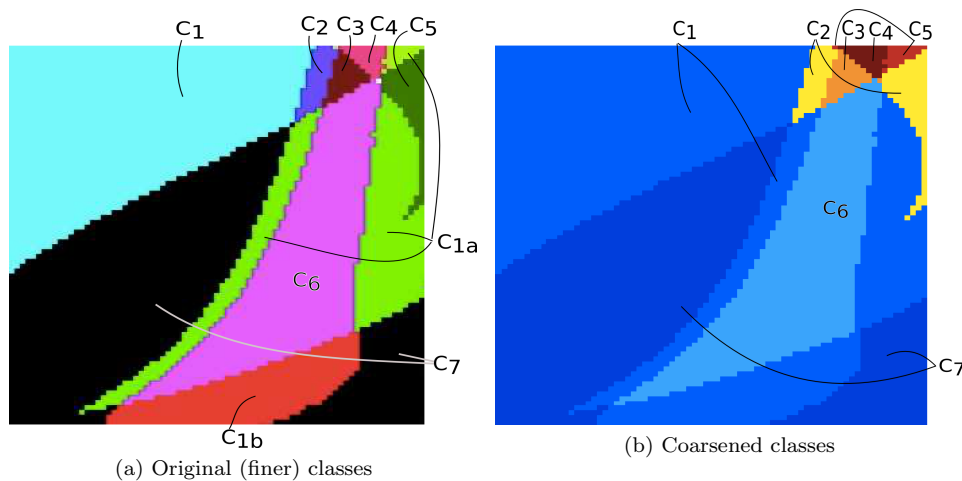(a) Original (finer) classes

(b) Coarsened classes

Figure 5.: A slice of parameter space at $z = 1$. (a) Continuation classes from the original paper by Arai *et al.*; (b) Classes obtained without machine learning algorithms, but using the coarser notion of equivalence. Classes labels, C_*, correspond between (a) and (b) as well as to those in figures below. (color online)

Recall that in Arai *et al.*, a Morse graph $\mathrm{MG}(\mathcal{F}_{Q_0})$ is computed for each grid element $Q_0 \in \mathcal{Q}$. The clutching graph $\mathcal{J}(Q_0, Q_1)$ is then formed for all adjacent boxes $Q_0, Q_1 \in \mathcal{Q}$, and the clutching function ι is tested to determine if it defines a directed graph isomorphism. The equivalence classes with respect to the transitive closure form the continuation classes. In the continuation diagram, grid elements of identical colors indicate that they belong to the same continuation classes.

In contrast, in Figure 5b, we present classification using a coarser equivalence relationship, where two Morse graphs are equivalent if they are isomorphic. This represents our 'ground-truth' for classification. The boundaries between the clusters of identically colored grid elements in parameter space are potentially critical transitions in the dynamical behavior. It should be noted that while Conley-Morse database results are rigorous at the pre-specified resolutions, the results depend upon resolution in both parameter space and phase space. For the purposes of this paper, our resolutions are identical to those in [2]. However, for future analysis the efficiency of machine learning allows us to use much finer resolutions in both parameter and phase space.

5.   Methodology

In this section we provide a review of the requisite machine learning techniques, and give a description of our algorithms.

5.1.   *Active learning*

Active learning is a form of supervised learning in which it is possible for the learning algorithm to perform interactive queries of the data in order to generate a training set [31, 33]. The rationale for active learning is that a machine learning classifier can achieve greater accuracy with a smaller training set if it is allowed to choose which samples to learn from.

Active learning is primarily motivated by instances where labeling data is expensive. In the setting of predicting bifurcations, labeling data corresponds to computing Morse graphs. As this is one of the most computationally intensive parts of the construction of the Conley-Morse database, the active learning paradigm applies naturally to building the database.

The most popular form of active learning is *uncertainty sampling*, where samples are chosen based on where the classifier is least certain of classification. In other words, samples which change the potential classification by small amounts are of little value, whereas samples which can change the classification substantially are of high value.

In order to incorporate this philosophy into the framework of the Conley-Morse database, we rely on the continuation properties of the Morse graphs. We make a heuristic assumption that regions enclosed by grid elements with isomorphic Morse graphs likely belong to the same continuation class. As such, queries within such a region are thought to be of low value. In selecting a random query, an active learning subroutine decides whether the query is of high value, and if so, adds it to the training set. The goal of our active learning subroutine is to iterate this procedure in order to build up a small, yet informative training set.

5.2.   *Algorithmic Preliminaries*

The k-Nearest Neighbor (k-NN) classification algorithm is a well-established method for classifying or labeling objects based on the closest training examples in a feature space. For more detail, see [1].

For our purposes, the feature space is the discretized parameter space $\mathcal{Q}$ of a dynamical system. The class of a grid element Q_0 in the $\mathcal{Q}$ is the equivalence class to which the associated Morse graph $MG(\mathcal{F}_{Q_0})$ belongs.

5.3.   *Delaunay Triangulation*

To interactively construct an effective training set, our goal is to choose a minimal number of informative samples. To do this, we consider the geometry of the current training set in parameter space, and accept queries from boundary regions.

In order to judge whether a query is in such a region, it is natural to turn to computational geometry. Our computational geometry approach is to consider the Delaunay triangulation of the elements of the current training set.

Definition 5.1: A Delaunay triangulation for a set P of points in a plane is a triangulation $\mathcal{D}(P)$ such that no point in P is inside the circumcircle of any triangle in $\mathcal{D}(P)$.

The benefit of the Delaunay triangulation is that it produces 'fat' simplices by maximizing the minimum angle of all the angles in the triangulation. Such triangles are more effective for our purposes. Using point location algorithms, it is straightforward to determine the location of the query in the triangulation, and compare Morse graphs of the vertices that make up the encompassing simplex. If the vertices of the encompassing simplex have isomorphic Morse graphs, the query is discarded

as a low-information point, otherwise it is accepted into the training set. Computational geometry, in the form of the Voronoi diagram, has previously been used in conjunction with the nearest-neighbor classification algorithm [12]. However, our approach is disparate in that our active learning techniques are tailored to the dynamics rather than the classification algorithm.

The time complexity of the implemented Delaunay triangulation algorithm is $O(n \log n)$ in the planar case, while the complexity of the point location algorithm is $O(\log n)$ [8]. We consider this an acceptable computational overhead in order to generate an effective training set. However, for dimension d, the complexity of the Delaunay triangulation is $O(n^{\lfloor \frac{d+1}{2} \rfloor + 1})$ [8]. Above two dimensions this overhead is quite large, and this approach becomes infeasible for large training sets. In order to circumvent this problem, we consider two different directions. First, we split the three dimensional domain into slices along the z axis. Second, we develop a similar active learning algorithm that performs at a lower complexity.

We denote the Delaunay triangulation based algorithm ALGORITHMD, and give pseudocode below. First, recall $\mathcal{Q}$ denotes the discretized parameter space. We will denote the training set by $\mathcal{T}$ and for a grid element $Q \in \mathcal{Q}$, we will abuse notation slightly by defining $\mathrm{MG}(Q)$ as the label of the Morse graph of Q. We define the test set as $\mathcal{R} := \mathcal{Q} \backslash \mathcal{T}$, and apply the 1-NN algorithm to $\mathcal{R}$ to obtain the classification of all the grid elements in $\mathcal{R}$.

ALGORITHMD$(\mathcal{Q}, m)$

```
1    M ← m              ▷ Max size of training set
2    𝒯 ← uniform random samples from 𝒬.
                         ▷ Initial training set used to construct
                         ▷ a preliminary triangulation.
3    𝒞(𝒯) ← {MG(Q) | Q ∈ 𝒯}.
4    Compute 𝒟(𝒯)       ▷ Delaunay triangulation of the current training set 𝒯.
5    repeat  Q₀ ← uniform random sample from 𝒬\𝒯
6            Find S₀ ∈ 𝒟(𝒯), such that Q₀ ∈ S₀
7            if All the vertices of S₀ have isomorphic Morse graphs
8                then 𝒯 ← 𝒯 ∪ {Q₀} and 𝒞(𝒯) ← 𝒞(𝒯) ∪ {MG(Q₀)}.
9        until |𝒯| = M
10   return 𝒯
```

Remark 1 : After creating the training set we utilize the 1-NN classification algorithm to classify the remaining test set. However it is worth noting that this approach is not limited to a nearest neighbor classification. For instance, our approach can be adopted to be compatible with a variety of different machine learning classifiers.

Remark 2 : In general the problem determining if two directed graphs are isomorphic is notoriously difficult, though it is not known to be NP-Complete [30]. In practice, a Morse graph has only a few vertices, and the isomorphism problem does not present a computational issue. For moving beyond the Leslie model, we are investigating heuristics for the graph isomorphism problem.

5.4.   *Nearest Neighbor Sampler*

Our second active learning algorithm also considers the elements of the current training set which surround a query. However, the approach this time is not computational geometry, but combinatorial. To discern which elements surround a random query, we use a k-NN algorithm to obtain the k nearest neighbors. We

then check whether the majority of the corresponding Morse graphs of the neighbors are isomorphic. If so, the query is discarded, otherwise the query is accepted into the training set.

This approach benefits from the wealth of exact and approximate k-NN algorithms, and their respective data structures, which have been recently developed for dealing with high dimensional data [13, 16, 36]. For dynamical systems with high dimensional parameter spaces, this method scales more efficiently than a Delaunay triangulation approach.

We now introduce our nearest-neighbor based algorithm, entitled ALGORITHMN, and which is very similar in spirit to ALGORITHMD above.

ALGORITHMN$(\mathcal{Q}, k, m)$

1 $M \leftarrow m$ ▷ Max size of training set
2 $\mathcal{T} \leftarrow$ uniform random samples from $\mathcal{Q}$
                   ▷ Initial training set used to construct
                   ▷ a preliminary triangulation.
3 **repeat** $Q_0 \leftarrow$ uniform random sample from $\mathcal{Q} \backslash \mathcal{T}$
4          $\mathcal{C}(\mathcal{T}) \leftarrow \{\mathrm{MG}(Q) \mid Q \in \mathcal{T}\}$.
5          $\mathcal{N}_k(Q_0) \leftarrow k$ nearest neighbors of Q_0 in $\mathcal{T}$.
6   **if** *majority* $\mathrm{MG}(Q)$ for $Q \in \mathcal{N}_k(Q_0)$ not isomorphic
                   ▷ Using specified majority rule
7   **then** $\mathcal{T} \leftarrow \mathcal{T} \cup \{Q_0\}$ and $\mathcal{C}(\mathcal{T}) \leftarrow \mathcal{C}(\mathcal{T}) \cup \{\mathrm{MG}(Q_0)\}$
8   **until** $|\mathcal{T}| = M$
9 **return** $\mathcal{T}$

Remark 3 : In Line 6 of ALGORITHMN, any majority voting rule can be used. In our implementation, we used a simple majority rule, requiring at most half of the elements of $\mathcal{N}$ to be isomorphic in order to add the associated Q_0 to the training set.

6.   Results

In this section we describe the results of our experiments using the parameter space of the three dimensional Leslie Model shown in Section 4. In accordance with [2], we set the resolution of the grid on the three-dimensional parameter space of the Leslie Model as $80 \times 80 \times 40$.

We validated our algorithms on both two dimensional slices of the domain and the entire three dimensional domain. As previously stated, the 1-NN classifier with Euclidean distance is used to assign labels to the grid elements of the test set.

6.1.   *Two Dimensional Slices*

In order to efficiently use the Delaunay triangulation we consider $\mathcal{Q}$ in slices along the z axis. Considering the domain in two dimensions lends itself to parallelization of the classification process, which further increases efficiency. Each two dimensional domain is 80×80, creating a total size of 6400 grid elements.

First a size n is chosen for the training set, which for our experiments was 640 or 1280 samples, i.e. 10% or 20% of the domain. As described above, we start with a small initial set of m random samples. For our experiments we set $m = 75$, which allows for sufficient initial coverage. The rest of the training set, or $n - m$ elements, is then generated with an active learning subroutine. The remaining $6400 - n$ grid elements determine $\mathcal{R} = \mathcal{Q} \backslash \mathcal{T}$. For our experiments, vectors in $\mathcal{R}$ are classified

14

(a) $z = 1$, $|\mathcal{T}| = 640$ (b) $z = 9$, $|\mathcal{T}| = 1280$
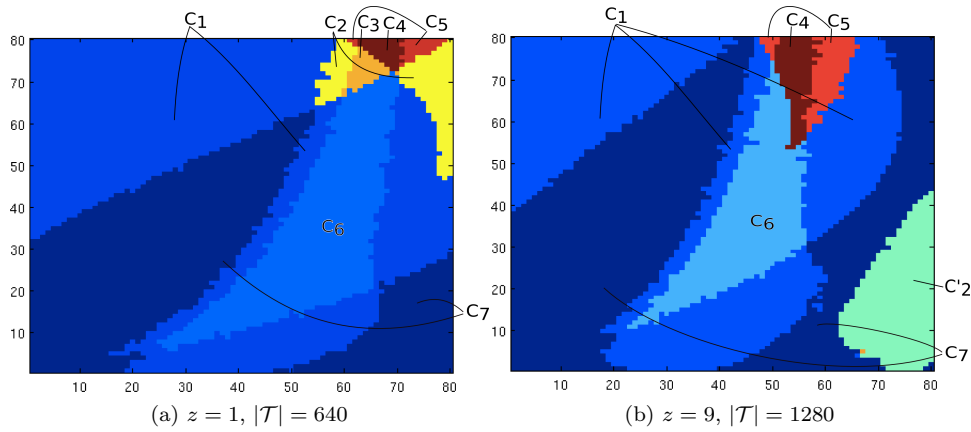
Figure 6.: (a) z slice 1, training set size 10% of $\mathcal{Q}$, misclassification of 244 points, or 3.8125% classification error; (b) z slice 9, training set size 20% of $\mathcal{Q}$, misclassification of 79 points, or 1.234% classification error. As in Figure5, classes that correspond across the slices are labeled the same. Notice that in (b) class C_2' is a newly detected class, while the dynamics detected in classes C_2 and C_3 do not show up in the $z = 9$ slice of parameter space.

using the 1-NN algorithm, which gives $\mathcal{C}_{\mathcal{R}}$. We define the prediction error ϵ for P as the number of elements that are misclassified:

$$\epsilon(P) = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} \delta(\mathrm{MG}(Q), \mathcal{C}_{\mathcal{R}}(Q))$$

where $\mathcal{C}_{\mathcal{R}}(Q)$ is the predicted class of Q, $\mathrm{MG}(Q)$ is the actual class of Q, and δ is the Kronecker delta function, given by

$$\delta(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{if } x \neq y \end{cases}$$

6.1.1.   ALGORITHMD

In this section we review the results of ALGORITHMD for the two dimensional case. In Figure 6 we show the classification for the two dimensional domains, $z = 1$ and $z = 9$. We test our algorithms on training sets of size 10% and 20% of the total domain, or 640 and 1280 points, respectively.

Figure 7 shows the Delaunay triangulation of the final training set over the classifications given by the 1-NN algorithm. This image illustrates the algorithm's ability to enhance resolution around the boundaries of the equivalence classes in parameter space.

In Table 1 is the misclassification of the database by compiling all of the z slice calculation, along with a comparison to a training set generated uniformly at random. For brevity, we will not discuss the results of the indvidual slices, however there are noticeable differences between slices. Due to these differences, we are currently investigating incorporating an adaptive budget to allocate the training set based on the complexity of the slices.
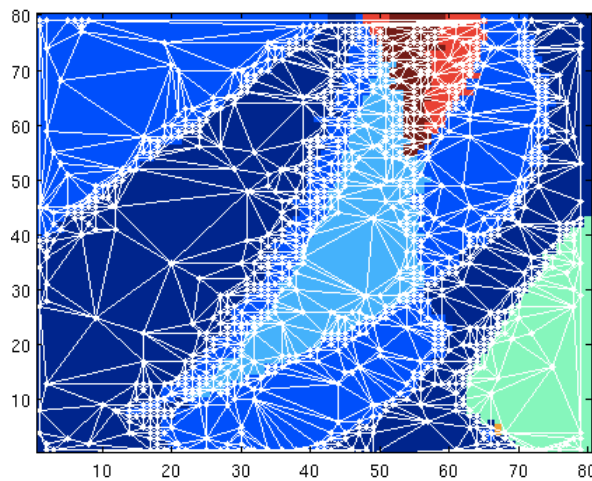
Figure 7.: Delaunay triangulation of $\mathcal{T}$ generated using AlgorithmD; $z = 9$, $|\mathcal{T}| = 1280$.

6.1.2. AlgorithmN

In this section we review the results of AlgorithmN for the two dimensional case. As expected, a slightly elevated misclassification rate is the price for increased computational efficiency. Figure 8 show a plot of misclassification compared to a training set generated uniformly at random. In order to highlight the regions of the parameter space AlgorithmN focuses on, we plot the training set over a 1-NN classification below. The color of the training set elements darken in proportion to the size of the current training set.
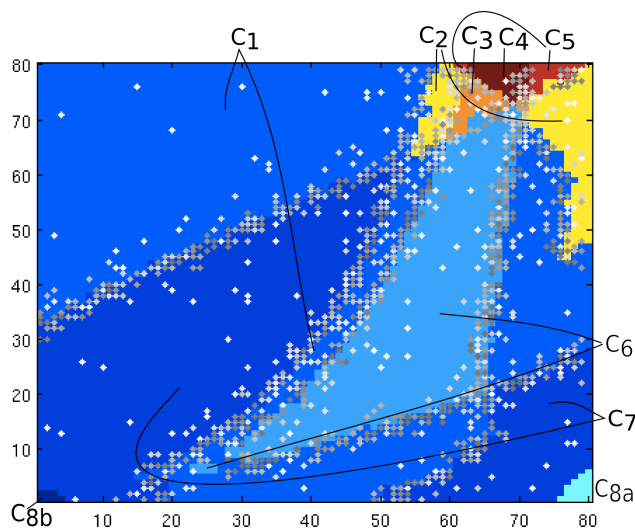

Figure 8.: Plot of training set for AlgorithmN on $z = 1$, $|\mathcal{T}| = 1280$, Misclassification of 123 grid elements, or 1.92% of $\mathcal{Q}$; The shade of a training set element depends upon when it was selected in the process of constructing $\mathcal{T}$. Classes correspond to those in Figures 5 and 6.

6.1.3. Slices in x and y Directions.

In order to further validate this algorithm, we ran both AlgorithmD and AlgorithmN on slices in the x and y direction as well. For brevity, we only show the results for 20% of the domain.

Training Set %	AlgorithmD	AlgorithmN	Uniform
10%	4.5	4.9	7.13
20%	1.1	2.78	4.98

Table 1.: Results for compilation of 2D z slices for AlgorithmD, AlgorithmN and Uniform Sampling.

6.2. *Three Dimensional Domain*

In this section we display results from AlgorithmN applied to the entire three dimensional domain, instead of dividing the domain into slices. We only include results for AlgorithmN, as the overhead of constructing the Delaunay triangulation in three dimensions yields AlgorithmD unwieldy.

Training Set %	AlgorithmN	Uniform
10 %	5.74	7.46
20%	3.19	5.86

Table 2.: Results for 3D Variants of AlgorithmN and Uniform Sampling.

7.   Discussion

In this paper we introduced an approach to predict bifurcations using machine learning incorporated into the Conley-Morse database framework. This approach to the study of complex dynamical systems provides a rich description of the system in both the phase space and the parameter space. This is especially relevant in the biological sciences, where obtaining precise values of parameters often requires extensive collection of experimental data. For instance, certain models of chemotaxis [25] involve upwards of seventeen parameters, some only known approximately, and that range across eight orders of magnitude. In the database framework described in Section 2, the parameters vary over a range of values, with the rigorous results valid over that entire region in parameter space. Usually a substantial effort must be spent validating and tuning model parameters [6]. The database framework shifts this emphasis to the analysis of the database results to find a range of parameters where model prediction is consistent with the data.

Even though computationally efficient, the database description of the parameter space still suffers from the curse of dimensionality. We have shown that active sampling techniques based on assumptions of the underlying dynamics in combination with a straightforward k-NN classifier can lead to very accurate predictions of regions around potential bifurcations, thereby alleviating some of the computational burden due to high-dimensional parameter spaces. In particular, one can approximate the Conley-Morse database with greater than 97% accuracy through computing only 20% of the total Morse graphs.

We are currently working to incorporate adaptive computation into the database construction in order to provide rigorous continuation classes at lower computational cost. Furthermore, we are also investigating additional nonlinear systems with higher dimensional parameter spaces. The presentation and exposition of this paper has focused on dynamical systems in the form of discrete maps. However, similar theoretical ideas apply to dynamical systems represented by differential equations, and extending the Conley-Morse database to differential equations is

currently being investigated. In contrast to maps, constructing the combinatorial representation requires rigorous numerical calculation, which incurs a much greater computational cost than evaluating a map. Therefore, a machine learning approach such as the one presented in this paper becomes even more relevant in such a setting.

References

[1] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, Cambridge, MA, 2nd edition, 2010.

[2] Zin Arai, William Kalies, Hiroshi Kokubu, Konstantin Mischaikow, Hiroe Oka, and Pawel Pilarczyk. A Database Schema for the Analysis of Global Dynamics of Multiparameter Systems. *SIAM Journal on Applied Dynamical Systems*, 8(3):757, 2009.

[3] Yaneer Bar-Yam. *Dynamics of Complex Systems*. Westview Pr, Cambridge, MA, 1 edition, 1997.

[4] Jesse Berwald, Tomáš Gedeon, and John Sheppard. Using Machine Learning to Predict Catastrophes in Dynamical Systems. *Journal of Computational and Applied Mathematics*, 236(9):2235–2245, November 2012.

[5] Carlo Bianca. Onset of nonlinearity in thermostatted active particles models for complex systems. *Nonlinear Analysis: Real World Applications*, 13(6):2593–2608, December 2012.

[6] Carlo Bianca. Thermostatted models Multiscale analysis and tuning with real-world systems data. *Physics of Life Reviews*, 9(4):418–425, December 2012.

[7] M. Costa, Ionita Ghiran, CK Peng, A Nicholson-Weller, and A.L. Goldberger. Complex dynamics of human red blood cell flickering: Alterations with in vivo aging. *Physical Review E*, pages 1–4, 2008.

[8] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. Delaunay Triangulations. In *Computational Geometry: Algorithms and Applications*, chapter 9, pages 191–218. Springer Verlag, Berlin, 2008.

[9] Michael Dellnitz, Gary Froyland, and Oliver Junge. The Algorithm Behind GAIO – Set Oriented Numerical Methods for Dynamical Systems. In *Ergodic theory, analysis, and efficient simulation of dynamical systems*, pages 145–174. Springer, 2000.

[10] Peter D. Ditlevsen and Sigfus J. Johnsen. Tipping points: Early warning and wishful thinking. *Geophysical Research Letters*, 37(19):2–5, October 2010.

[11] Sarah M. Glaser, Michael J. Fogarty, Hui LiuIrit Altman, Chih-hao Hsieh, Les Kaufman, Alec D. MacCall, Andrew A. Rosenberg, Hao Ye, and George Sugihara. Dynamic complexity in coupled human-marine resource systems. 2011.

[12] Martina Hasenjger and Helge Ritter. Active learning with local models. *Neural Processing Letters*, 7:107–117, 1998. 10.1023/A:1009688513124.

[13] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM.

[14] Ola M Johannessen, Kirill Khvorostovsky, Martin W Miles, and Leonid P Bobylev. Recent ice-sheet growth in the interior of Greenland. *Science (New York, N.Y.)*, 310(5750):1013–6, November 2005.

[15] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek. *Computational Homology*. Springer, 2004.

[16] Norio Katayama and Shin'ichi Satoh. The sr-tree: an index structure for high-dimensional nearest neighbor queries. *SIGMOD Rec.*, 26(2):369–380, June 1997.

[17] Anatole Katok and Boris Hasselblatt. *Introduction to the Modern Theory of Dynamical Systems*. Cambridge University Press, Cambridge, 1995.

[18] Timothy M Lenton, Hermann Held, Elmar Kriegler, Jim W Hall, Wolfgang Lucht, Stefan Rahmstorf, and Hans Joachim Schellnhuber. Tipping elements in the Earth's climate system. *Proceedings of the National Academy of Sciences of the United States of America*, 105(6):1786–93, February 2008.

[19] Simon A Levin. Complex Adaptive Systems: Exploring the known, the unknown and the unknowable. *Bulletin of the American Mathematical Society*, 40(1):3–19, 2002.

[20] Edward N Lorenz. Irregularity : a fundamental property of the atmosphere. *Tellus A*, 36A(2):98–110, 1984.

[21] D. Ludwig, D. D. Jones, and C. S. Holling. Qualitative Analysis of Insect Outbreak Systems: The Spruce Budworm and Forest. *Journal of Animal Ecology*, 47(1):315–332, February 1978.

[22] Konstantin Mischaikow. The Conley Index Theory: A brief introduction. *Banach Center Publications*, 47, 1999.

[23] Konstantin Mischaikow and Marian Mrozek. Conley index theory. In *Handbook of Dynamical Systems II: Towards Applications*, chapter 9, pages 393–460. North-Holland, 2002.

[24] J.D. Murray. *Mathematical Biology: I. An Introduction*. Interdisciplinary Applied Mathematics. Springer, 2002.

[25] Matthew P Neilson, Douwe M Veltman, Peter J M van Haastert, Steven D Webb, John a Mackenzie, and Robert H Insall. Chemotaxis: a feedback-based computational model robustly predicts multiple aspects of real cell behaviour. *PLoS biology*, 9(5):e1000618, May 2011.

[26] M.E.J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[27] Stefan Rahmstorf, Michel Crucifix, Andrey Ganopolski, Hugues Goosse, Igor Kamenkovich, Reto Knutti, Gerrit Lohmann, Robert Marsh, Lawrence a. Mysak, Zhaomin Wang, and Andrew J. Weaver. Thermohaline circulation hysteresis: A model intercomparison. *Geophysical Research Letters*, 32(23):1–5, 2005.

[28] E. Rignot, I. Velicogna, M. R. van den Broeke, a. Monaghan, and J. Lenaerts. Acceleration of the

contribution of the Greenland and Antarctic ice sheets to sea level rise. *Geophysical Research Letters*, 38(5):1–5, March 2011.

[29] Marten Scheffer, Jordi Bascompte, W.A. Brock, Victor Brovkin, S.R. Carpenter, Vasilis Dakos, Hermann Held, E.H. Van Nes, Max Rietkerk, and George Sugihara. Early-warning signals for critical transitions. *Nature*, 461(7260):53–59, 2009.

[30] Uwe Schöning. Graph isomorphism is in the low hierarchy. *J. Comput. Syst. Sci.*, 37(3):312–323, December 1988.

[31] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[32] G Sugihara. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, 344:734–741, 1990.

[33] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, MULTIMEDIA '01, pages 107–118, New York, NY, USA, 2001. ACM.

[34] Warwick Tucker. *Validated Numerics*. Princeton University Press, Princeton, NJ, 1st edition, 2011.

[35] Ilie Ugarcovici and Howard Weiss. Chaotic dynamics of a nonlinear density dependent population model. *Nonlinearity*, 17:1689–1711, 2004.

[36] Peter N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, SODA '93, pages 311–321, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.