Trabalho apresentado no XXXVIII CNMAC, Campinas - SP, 2018.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Parameter Identification in a Predator-Prey System using Persistent Homology

Sabrina S. Calcina[1]
Instituto de Ciências Matemáticas e de Computação, USP, São Carlos, SP
Marcio Gameiro[2]
Instituto de Ciências Matemáticas e de Computação, USP, São Carlos, SP

Abstract. The present work uses persistent homology combined with machine learning to identify (classify) parameters of system of equations producing complex patterns. Persistent homology is used as a tool to extract topological information from the patterns. This topological information is in turn used as features for the machine learning methods used for the classification. The method is applied to patterns generated by a predator-prey system using the SVM, PLS-DA, and the Naive Bayes machine learning methods.

Keywords. Persistent homology, Parameter identification, SVM classifier, PLS-DA classifier, Naive Bayes classifier.

1 Introduction

Mathematical models, in particular, differential equations, are extensively used to study problems in sciences and engineering, and hence, it is extremely important to develop methods to design and analyze such models. Designing mathematical equations to model problems in sciences and engineering is referred to as *mathematical modeling* [14]. When modeling a problem, it is necessary to decide what type of model to use, i.e., whether to use differential equations or statistical models for example, and then to choose the appropriate equations to describe the problem. Once a suitable model has been chosen, it is necessary to ensure that it is solved correctly and that it solves the correct problem. In scientific computing, this step is referred to as *verification* and *validation* of the model [5, 14]. More specifically, *verification* can be defined as the process of ensuring that the model is correctly implemented and the solution is accurate ("solving the equations right") and *validation* can be defined as the process of determining that the model provides an accurate description of the problem it is intended to describe ("solving the right equations"). The validation process often involves comparing the results from the model to experimental data [9, 14].

Mathematical models often include parameters that need to be determined during the validation process. The process of determining the model parameters, called *parameter*

[1]sabrinas@icmc.usp.br
[2]gameiro@icmc.usp.br

identification, is a very important and challenging problem that is often addressed by comparing the solutions of the model to experimental data [4, 9, 12, 15, 16].

In this paper, we propose to apply techniques from *topological data analysis*(TDA) [3], more specifically, *persistent homology* [6] combined with *machine learning* [2] to study the parameter identification problem in models producing complex spatio-temporal patterns [7, 12]. More precisely, we compute persistent homology of the level sets of the patterns produced by the system and use the corresponding persistence diagrams as features for machine learning algorithms. One important aspect of the proposed method is that it can be applied directly to the patterns (images) generated by the system, and hence can also be applied to experimental data where we have only images representing the state of the system, such as experiments in fluid dynamics for example [11, 12].

2 Persistent Homology

In this section, we present a brief description of persistent homology. For a more in-depth discussion please see [6, 10]. Persistent homology is a tool that provides metric information about the topological properties of an object and how robust these properties are with respect to change in parameters. More specifically, persistent homology counts the number of connected components and holes of various dimensions and keeps track of how they change with parameters.

Suppose that we have a space (object) X that varies as a function of a parameter. Persistent homology provides a way of capturing how the shape of this object changes as we vary this parameter. To make this more precise, we need to describe the type of spaces X we consider and how X changes with the parameter [10].

Let $h \in \mathbb{R}$ be a fixed grid size. Given $j \in \mathbb{Z}$ we denote by $I_j = [jh, (j+1)h]$ the interval of length h with end-points jh and $(j+1)h$. An *n-dimensional cube* (or a *cube of dimension n*) is a set of the form $I_{j_1} \times I_{j_2} \times \cdots \times I_{j_n}$, where $j_1, j_2, \ldots, j_n \in \mathbb{Z}$. An *$n$-dimensional cubical complex* is a finite collection X of n-dimensional cubes.

To a cubical complex X, we associate a collection of groups $H_k(X)$, $k = 0, 1, \ldots$, called *homology groups* of X, that provide the essential topological features of X. For the type of complexes that we consider in this paper, the homology groups are of the form $H_k(X) = \mathbb{R}^{\beta_k}$, where β_k is a non-negative integer called the *k-th Betti number* of X. Therefore, for the cubical complexes we considered in this paper, the homology groups are in fact vector spaces, and the Betti numbers are the dimensions of these vector spaces. The Betti numbers have the very important property that the k-th Betti number β_k is equal to the number of "k-dimensional holes" in X. More specifically, for $k = 0, 1, 2$, β_0 is the number of *connected components* of X, β_1 is the number of *holes* or *tunnels* in X, and β_2 is the number of *cavities* in X. For more details see [6, 10]. In this paper we consider only 2-dimensional cubical complexes, hence we are concerned only with the number of components β_0 and the number of holes β_1. For an example, see Figure 1.

Given a finite collection of n-dimensional cubical complexes $X_1 \subset X_2 \subset \cdots \subset X_r$, *persistent homology* provides information about the changes in the Betti numbers as we move from one cubical complex X_j to the next one X_{j+1}. The collection of cubical complexes X_i is called a *filtration* and denoted by $\mathcal{X}$. More precisely, the *persistent homology* $PH_k(\mathcal{X})$ of

DOI: 10.5540/03.2018.006.02.0444

$\mathcal{X}$ is characterized by its *persistence diagrams* $PD_k(\mathcal{X})$, $k = 0, 1, \ldots$, where each $PD_k(\mathcal{X})$ is a multi-set of pairs of points of the form (b, d) called *birth-death pairs* [6]. Each point $(b, d) \in PD_k(\mathcal{X})$ represents a k-dimensional hole γ in $\mathcal{X}$. The number $b \in \{1, 2, \ldots, r\}$ is called the *birth time* (*birth index*) of γ and the number $d \in \{1, 2, \ldots, r, +\infty\}$ is called the *death time* (*death index*) of γ. We say that γ was *born* at time b and *died* at time d. The birth time b indicates where in the filtration the hole γ first appears, and the death time d indicates where in the filtration γ disappears. Notice that d is allowed to be $+\infty$, to account for the cases where γ never dies. Software for efficient computation of persistent homology is available [13]. Figure 1 presents an example of a cubical filtration and its persistence diagrams.

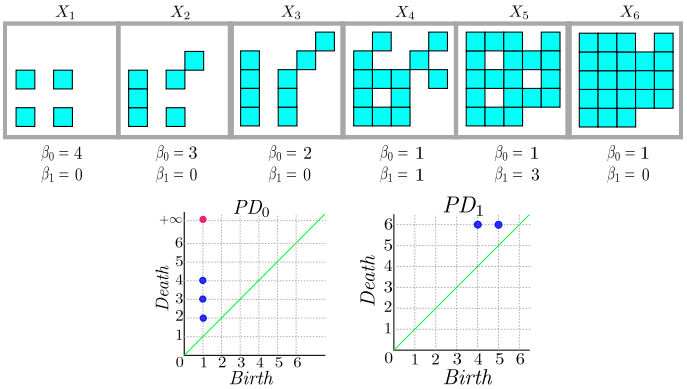

Figure 1: Filtration of cubical complexes $X_1 \subset X_2 \subset \cdots \subset X_6$, and their Betti number β_0 and β_1 (top), and the corresponding persistence diagrams (bottom).

3 Machine Learning

Machine learning is becoming one of the most active areas of research in computer science and data analysis in recent years. One of the reasons for this is the great number of successful applications of machine learning in many different areas of science [2]. Machine learning can be broadly divided into two main areas: *supervised learning* and *unsupervised learning*. In supervised learning we have a dataset, called *training dataset*, for which we know the answers to the questions we are interested in and we use this dataset to "train our machine". We then use our "trained machine" to obtain the answers to our questions for other datasets. In unsupervised learning, on the other hand, we want to extract information (such as clustering information, for example) from our dataset without the aid of a training dataset.

One of the main tasks in supervised learning is *classification*: given a dataset, we want to classify each element of the set as belonging to one of a predetermined collection of classes. This can be described more formally as follows. Let X be a vector space, the elements of which are called *feature vectors* and are meant to represent the features used to describe our objects. Let $C = \{c_1, c_2, \ldots, c_d\} \subset \mathbb{R}$ be a set of *class labels*. The goal of *supervised classification* is to classify each element of X as belonging to one of the classes given by C. To this end, assume that we are giving a set of pairs $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \subset X \times C$ called *training dataset*. Given one such pair

(x_i, y_i) we say that the vector x_i belongs to the class labeled by y_i. Supervised machine learning classifies the elements of X by using the training dataset to "learn" (or "train") a parameter dependent function $g\colon X \times \mathbb{R}^m \to C$ satisfying some given optimality conditions and such that $g(x_i, \alpha) = y_i$ for all $i = 1, \ldots, N$. Learning the function g means finding a value for the parameter $\alpha = \alpha_0$ such that $f(x) := g(x, \alpha_0)$ satisfies all the required conditions. Once we have the trained function $f\colon X \to C$, we define the class to which a vector $x \in X$ belong to be the class whose label is given by $f(x) \in C$.

The machine learning methods used in this paper are the *Support Vector Machines* (SVM) classifier [8], the *Partial Least Squares-Discriminant Analysis* (PLS-DA) classifier [1], and the *Naive Bayes* classifier [2]. The performance of a machine learning method on a dataset is measured by the *accuracy*, which measures the overall amount of correct identification from all predictions that were made (percentage) on the dataset.

4 Persistence Homology of Level Sets

Given a function $u\colon \Omega \to \mathbb{R}$ defined on a rectangle $\Omega := [a, b] \times [c, d]$, we can construct a cubical complex filtration by making a grid on the domain Ω and considering the sub-level sets of u given by $U_r := \{(x, y) \in \Omega \mid u(x, y) \leq r\}$.

We then define the cubical complex X_r to be the set of grid elements that intersect U_r. Since there is only a finite number of grid elements, we get a finite filtration of cubical complexes $X_{r_0} \subset X_{r_1} \subset \cdots \subset X_{r_N}$, with $r_0 < r_1 < \cdots < r_N$, where $X_{r_0} = \emptyset$, and X_{r_N} is the full cubical grid. Using this filtration we can compute the persistent homology of the sub-level sets of u (see Figure 3).

Our goal is to use persistent homology level sets to identify parameters in systems producing complicated spatio-temporal patterns. For this purpose we consider the following reaction-diffusion predator-prey system [7]

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta u + u(1-u) - \dfrac{uv}{\alpha + u} \\ \dfrac{\partial v}{\partial t} = \delta \Delta v + \beta \dfrac{uv}{\alpha + u} - \gamma v \end{cases} \tag{1}$$

defined on a rectangular domain Ω with no-flux (Neumann) boundary conditions. Here, $u(x, t)$ and $v(x, t)$ represent the population densities of prey and predators, respectively, at time t and position x. The choice of boundary conditions is equivalent to the assumption that both species cannot leave the domain.

We solve the predator-prey system (1) numerically on a uniform grid in space and time using a semi-implicit (in time) finite-differences method given in [7]. We denote the grid sizes in space by h and in time by Δt. For our experiments we fix the domain size and the parameter values as follows: $\Omega = [0, 400] \times [0, 400]$, $h = 1$, $\Delta t = 1/3$, $\alpha = 0.4$, $\gamma = 0.6$, and $\delta = 1$, and vary the parameter β. Figure 2 shows some solutions of (1) for different values of β. In Figure 3 we present some level sets of the solution on the left of Figure 2 and the persistence diagrams of the corresponding filtration.

DOI: 10.5540/03.2018.006.02.0444

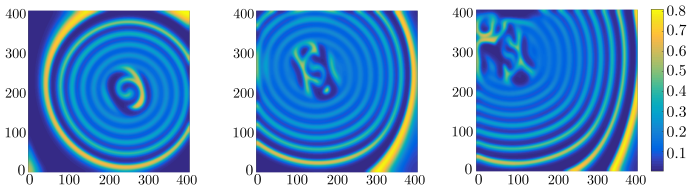

Figure 2: Level sets of solutions $u(x,t)$ for $t = 300$ of the predator-prey system (1) starting with a random initial condition for $\beta = 2.0$ (left), $\beta = 2.1$ (middle), and $\beta = 2.2$ (right).

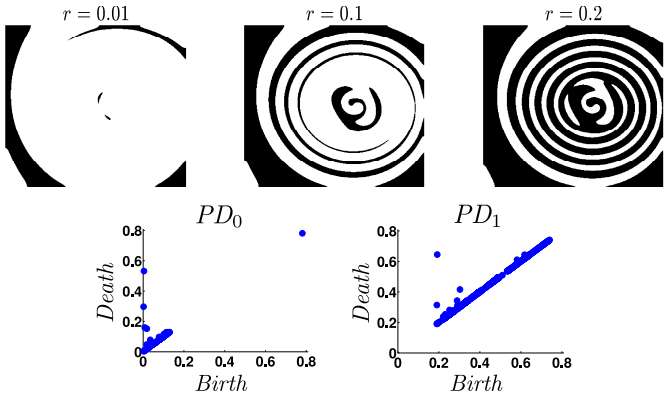

Figure 3: Some complexes on the filtration of level sets of the solution corresponding to $\beta = 2.0$ on Figure 2 (top) and the corresponding persistence diagrams (bottom).

5 Proposed Method and Results

The goal of this paper is to apply machine learning to identify parameters of solutions of (1). More specifically, we use the persistence diagrams of the level sets of the solutions to extract features from the solutions and apply machine learning to these features.

Recall from Section 2 that the k-dimensional persistence diagram of a level set filtration $\mathcal{X}$ is a multi-set $PD_k(\mathcal{X})$ of pairs of points of the form (b, d), where each pair correspond to the birth and death values of a given k-dimensional hole γ in terms of the level set values r for which γ appears and disappears in the filtration $\mathcal{X}$. To extract a feature vector from the persistence diagram $PD_k(\mathcal{X})$, we fix $r_{min} < r_{max}$ and consider only the persistence points whose birth values are in the interval $[r_{min}, r_{max}]$. Now consider a uniform grid $r_{min} = r_0 < r_1 < \cdots < r_m = r_{max}$ consisting of m subintervals of $[r_{min}, r_{max}]$, and let v_j be the number of pairs in the persistence diagram $PD_k(\mathcal{X})$ whose birth value b is in the interval $[r_{j-1}, r_j]$. We define the k-dimensional *persistence feature vector* of size m to be the vector $v^k(\mathcal{X}) = (v_1, v_2, \ldots, v_m) \in \mathbb{R}^m$.

5.1 Experiments and Results

As described in Section 4, we solve the predator-prey system (1) numerically on the domain $\Omega = [0, 400] \times [0, 400]$ with spatial grid size $h = 1$, time step $\Delta t = 1/3$, and the parameters values: $\alpha = 0.4$, $\gamma = 0.6$, and $\delta = 1$. For the computations in this paper, we consider three

values of the parameter β, namely, $\beta_1 = 2.0$, $\beta_2 = 2.1$, and $\beta_3 = 2.2$. For each value of the parameter β, we solve the system (1) up to $t = 300$, and consider the solutions $u(x,t)$ for t varying from $t = 100$ to $t = 300$ to form our dataset. Hence, we have three datasets of solutions corresponding to β_1, β_2, and β_3 that we denote by S_1, S_2, and S_3, respectively. Since $\Delta t = 1/3$, each dataset consists of 600 solutions of (1).

Now given values of r_{min}, r_{max}, and m, for each solution $u(x,t)$ in the datasets S_1, S_2, and S_3, we constructed a level set filtration $\mathcal{X}$, computed its persistence diagram using the software Perseus [13], and constructed the 0-dimensional and the 1-dimensional persistence feature vectors $v^0(\mathcal{X})$ and $v^1(\mathcal{X})$. Finally, we concatenated these two vectors and define the *persistence feature vector* $w(\mathcal{X}) := \big(v^0(\mathcal{X}), v^1(\mathcal{X})\big) \in \mathbb{R}^{2m}$. Therefore, we have three datasets of feature vectors, that we denote by $P_1(m)$, $P_2(m)$, and $P_3(m)$, each one consisting of 600 feature vectors of size $2m$.

We fix the values of $r_{min} = 0$ and $r_{max} = 0.792$ for the 0-dimensional and the 1-dimensional persistence diagrams, and compute the datasets $P_1(m)$, $P_2(m)$, and $P_3(m)$ for several values of m. For each value of m we apply the methods SVM, PLS-DA, and Naive Bayes to classify all possible pairs $P_i(m)$ and $P_j(m)$, and also to classify the three datasets $P_1(m)$, $P_2(m)$, and $P_3(m)$. For each run, we randomly selected 80% of the dataset as the training set and the remaining 20% as the test set. We run each computation 30 times and compute the average accuracy among these 30 computations. Figure 4 shows the plots of the average accuracy as a function of m. As we can see from these results, the classification is successful in all the cases. Hence, the method is effective in identifying the parameter values corresponding to each dataset.

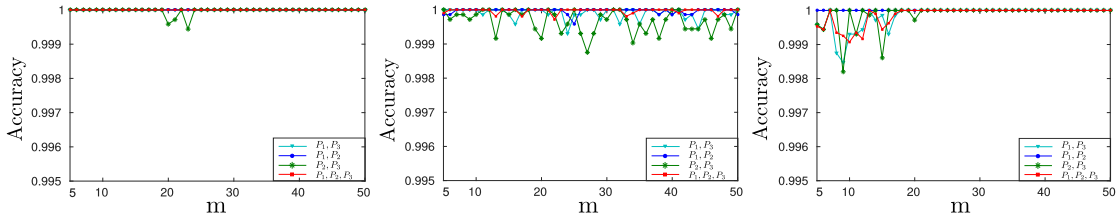

Figure 4: Average accuracy values versus the parameter m for SVM (left), PLS-DA (middle), and Naive Bayes (right) classifiers.

6 Conclusions

We use persistent homology as a feature extractor for machine learning methods to identify parameter in systems of equations exhibiting complex spatio-temporal patterns. The method is applied to the patterns generated by the system, hence it can be applied directly to experimental (image) data. The method presents excellent results on the datasets considered in our experiments.

Acknowledgment

M.G. was partially supported by FAPESP grants 2013/07460-7, 2016/08704-5, and 2016/21032-6, and by CNPq grants 305860/2013-5 and 310740/2016-9, Brazil. S.C. was partially supported by CAPES and FAPESP. We would like to thank CeMEAI for the use of the Euler cluster.

References

[1] M. Barker, and W. Rayens, Partial least squares for discrimination, *Journal of Chemometrics*, 17(3), 166–173, 2003.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2011.

[3] G. Carlsson, Topology and data, *Bull. Amer. Math. Soc.*, 46, 255–308, 2009.

[4] C. Coles, Parameter identification using mollification for predator-prey models in spatially heterogeneous environments, *Computers & Mathematics with Applications*, 48(3), 505–515, 2004.

[5] A. Cuesta, O. Abreu, and D. Alvear, *Evacuation Modeling Trends*, Springer, 2015.

[6] H. Edelsbrunner, *A Short Course in Computational Geometry and Topology*, Springer, 2014.

[7] M. R. Garvie, Finite-Difference Schemes for Reaction-Diffusion Equations Modeling Predator-Prey Interactions in MATLAB, *Bulletin of Mathematical Biology*, 69, 931–956, 2007.

[8] M. A. Hearst, Support Vector Machines, *IEEE Intelligent Systems*, 13(4), 18–28, 1998.

[9] A. Ives, A. Einarsson, V. Jansen, and A. Gardarsson, High-amplitude fluctuations and alternative dynamical states of midges in Lake Myvatn, *Nature*, 452, 84–87, 2008.

[10] T. Kaczynski, K. Mischaikow, and M. Mrozek, *Computational Homology*, Applied Mathematical Sciences 157, Springer, 2004.

[11] M. Kramàr, R. Levanger, J. Tithof, B. Suri, M. Xu, M. Paul, M. F. Schatz, and K. Mischaikow, Analysis of Kolmogorov flow and Rayleigh–Bénard convection using persistent homology, *Physica D: Nonlinear Phenomena*, 334, 82–98, 2016.

[12] K. Krishan, M. Gameiro, K. Mischaikow, M. F. Schatz, H. Kurtuldu, and S. Madruga, Homology and Symmetry Breaking in Rayleigh-Bénard Convection: Experiments and Simulations, *Physics of Fluids*, 19(11), 2007.

[13] V. Nanda, Perseus, `http://people.maths.ox.ac.uk/nanda/perseus/`, Accessed 28/03/18.

[14] W. L. Oberkampf, and C. J. Roy, *Verification and Validation in Scientific Computing*, Cambridge University Press, 2010.

[15] V. Pink, Identification of a predator-prey model parameters, *IOSR Journal of Mathematics*, 10(1), 89–94, 2014.

[16] J. G. Restrepo, and C. M. V. Sánchez, Parameter estimation of a predator-prey model using a genetic algorithm, *ANDESCON, 2010 IEEE*, 1–4, 2010.

DOI: 10.5540/03.2018.006.02.0444