Classification strategies in machine learning techniques predicting regime changes and durations in the Lorenz system

Cite as: Chaos 30, 053101 (2020); https://doi.org/10.1063/5.0003892
Submitted: 06 February 2020 . Accepted: 13 April 2020 . Published Online: 04 May 2020

Eduardo L. Brugnago 1, Tony A. Hild 1, Daniel Weingärtner 2, and Marcus W. Beims 2

NEW!
Sign up for topic alerts
New articles delivered to your inbox

© 2020 Author(s).
Classification strategies in machine learning techniques predicting regime changes and durations in the Lorenz system

Eduardo L. Brugnago, Tony A. Hild, Daniel Weingärtner, and Marcus W. Beims

Department of Physics, Federal University of Paraná, 81531-990 Curitiba, Brazil
Department of Informatics, Federal University of Paraná, 81531-990 Curitiba, Brazil
Max Planck Institute for the Physics of Complex Systems, Nöthnitzer Str. 38, 01187 Dresden, Germany

Author to whom correspondence should be addressed: mbeims@fisica.ufpr.br

ABSTRACT

In this paper, we use machine learning strategies aiming to predict chaotic time series obtained from the Lorenz system. Such strategies prove to be successful in predicting the evolution of dynamical variables over a short period of time. Transitions between the regimes and their duration can be predicted with great accuracy by means of counting and classification strategies, for which we train multi-layer perceptron ensembles. Even for the longest regimes the occurrences and duration can be predicted. We also show the use of an echo state network to generate data of the time series with an accuracy of up to a few hundreds time steps. The ability of the classification technique to predict the regime duration of more than 11 oscillations corresponds to around 10 Lyapunov times.

Long-term predictability of Lorenz’s chaotic times series is analyzed using classification strategies and machine learning techniques. The chaotic Lorenz attractor presents two wings named right and left regimes. The prediction of regimes changes and regime durations is considered here using quantities like the number of local extrema inside a regime and the remaining time inside each regime. High accuracies are obtained for the predictions, even for the longest regimes, which include Lyapunov times larger than 10. Predictions based on the classification techniques are compared to the usual prediction of the Lorenz times series itself and have shown to be superior.

I. INTRODUCTION

The ability to predict the future based on the past and present data is of high relevance in distinct areas like climate changes, stock markets, critical transitions, and extreme and rare events, which include giant ocean waves, extreme weather, and laser peaks, among others. Such forecasting has become some of the most required research in actual days. The dynamics in realistic problems is typical of highly non-trivial high-dimensional complex systems. To find a model capable of describing the complete dynamics in such complex systems is almost impossible. Instead of using models, many methods of forecasting rely on data sources from the real world itself. Thus, forecasting in complex systems is an extremely challenging open problem. On the other hand, machine learning strategies have been successfully applied to predict the evolution of variables in chaotic systems and to reproduce spatiotemporal chaos. Furthermore, it has been shown that multilayer neural networks are able to predict events in the Hénon map, while Multi-Layer Perceptrons (MLPs) were applied to solve chaotic problems. Several types of Recurrent Neural Networks (RNNs) have been used to replicate chaotic attractors and to calculate the respective Lyapunov exponents from the phase space data. Among the networks with reservoir, we can highlight the Echo State Network (ESN), a category of RNN whose reservoir neurons provide nonlinear response signals and the network output being generated by a trainable linear combination of these individual signals.

Information associated with time series of a dynamical system is often more relevant than the trajectory in the phase space itself. We mention the Lyapunov exponent and the period of a stable attractor. These quantities, however, are not used in this work. We focus on the property of variables in phase space. When studying
dynamical systems that present a double scroll attractor, such as the Lorenz and Chua’s systems, it is of interest to determine the time length of each scroll (i.e., the time spent inside each scroll) as well as to predict changes from one scroll to the other. Each scroll is a regime of motion and the time spent inside the scroll is the regime duration. Another point of recent investigation is the occurrence of extreme events in dynamical systems. The possibility to use machine learning to predict such extreme events is, without doubt, one of the great challenges nowadays. In the present work, each regime is considered as an event, whose “amplitude” is the regime duration.

We train an ensemble of MLPs to predict regimes changes and durations in the chaotic Lorenz system. For this, we use different strategies of label assignment, using two classification strategies which have not been applied in the context of prediction of time series, namely, quantities like (i) the local maxima and minima along the time series and (ii) the time spend inside each regime. Classification strategies were used recently to predict events in the Hénon map.

Even though the main focus of this work is the prediction of regime changes and durations, it is also possible to generate time series, for a finite time, using machine learning. For this, we use ESN trained for the chaotic time series of the Lorenz system to predict the time evolution of the dynamical variables. We present results comparing the time series generated by the RNN with the series from the reference data, obtained by direct numerical integration of Lorenz’s equations of motion. A comparison between both methods is also presented.

The paper is divided as follows. In Sec. II, the Lorenz system is presented. Section III explains the classification techniques, together with details about the numerical simulations. Results and discussions are presented in Sec. IV and conclusions in Sec. V.

II. THE LORENZ SYSTEM

The chaotic time series to train and test the machine learning technique were generated from the numerical integration using the RK4 method of the chaotic Lorenz system

\[ x = \sigma(y - x), \]  
\[ \dot{y} = x(\rho - z) - y, \]  
\[ \dot{z} = xy - \beta z, \]  

where \((x, y, z)\) are dynamical variables with dots representing the time derivative and \((\sigma, \rho, \beta) = (10, 28, 8/3)\) are the parameters. The Lorenz system is composed of three coupled differential equations which describe, in a simplified manner, a finite amplitude convection. For more details and discussions about the relation of the model with the atmosphere, we refer the reader to the original work of Lorenz.\(^{13}\) The Lorenz model is a well known prototype of chaotic behavior, which presents a lack of long-term predictability. For the numerical simulations, the dynamics was initialized using random initial conditions for the variables \((x, y, z)\) in the interval \((0, 1)\) and a transient of \(10^6\) integration steps was discarded. The chaotic Lorenz attractor resulting from the integration of Eqs. (1)–(3) presents two regimes of motion (one for each scroll), which can be defined according to the range of values of \(x\) variable: left regime defined for \(x < 0\) and right regime for \(x \geq 0\). Thus, our definition of regime is restricted just to \(x\) variable. Figure 1 illustrates the attractor of the system in phase space, the blue color represents the left regime and the red color the right one. For the predictions with respect to regime changes, we set the integration step at \(10^{-3}\). The largest Lyapunov exponent for this times series is \(\lambda = 1.306\) so that the corresponding Lyapunov time is \(t_\text{L} = (1.306)^{-1} \approx 0.766\). To determine the Lyapunov exponent, we consider \(5 \times 10^6\) points along the attractor and use a version of Wolf’s algorithm.\(^{21}\)

III. CLASSIFICATION STRATEGIES

In this section, we define the two classification strategies used in the simulations, namely, to associate (i) the number of local maxima (or minima) in the time series of the variable \(x\) inside each regime to one class and (ii) the time intervals for which the current regime will end, to another class. Local maxima (or minima) are defined by points which satisfy \(x = 0\). See details bellow. The above classification is in fact the relevant novelty of our work, since the dynamics of the Lorenz systems itself is widely known. Thus, we are not showing new results about the Lorenz system, but a novel way to make predictions with high accuracy in such a system.

A. Duration classes

The duration of the \(n\)th regime is given by \(D_n = \tau_{n+1} - \tau_n\), where \(\tau_n\) is the time at which the respective regime begins. The first regime \(n = 1\) is counted when, for the first time, we observe a sign change in the variable \(x\) along the time series (after discarding a transient). The \(n\)th regime is found when the variable \(x\) changed sign \(n\) times. Our strategy is to transform a time interval determination problem in a classification problem by relating \(D_n\) with a class. The goal is to predict, from instantaneous values of the system variables, the duration of a regime which have just started. In Fig. 2(a), we present an excerpt of the time series of the variables \((x, y, z)\) along the chaotic attractor and the interval of occurrence of the \(n\)th regime is shown. To classify a regime, we consider the \(k_n\) amount of local maxima in the right regimes and the local minima in the left regime.
regimes. Thus, the regime \( n \) has \( k_n \) maxima (or minima) and associated label \( k_n - 1 \). In the example shown in Fig. 2(a), we have \( k_n = 7 \) maxima associated with label 6. We relate a two-dimensional vector \( V_n = (y(t_n), z(t_n)) \) to this label. Note that \( x(t_n) = 0 \) according to the definition of regimes and the continuity of variables in the time series. From \( V_n \), we can determine the label of the correspondent regime.

**B. Imminence classes**

Another proposal is to determine when the current regime will end. For this, we define imminence classes \( (\Omega) \) depending on the time interval until the end of the current regime (TITE). All points on the attractor have an imminence class, used as a label in the classification method so that it is possible to approximate TITE. To assign labels to the current time, we define the imminence function \( \Omega_n(t) \) with \( t \in (\tau_n, \tau_{n+1}) \) corresponding to regime \( n \). The function

\[
\Omega_n(t) = \begin{cases} 
3 & \tau_{n+1} - t < 0.01, \\
2 & 0.01 \leq \tau_{n+1} - t < 0.10, \\
1 & 0.10 \leq \tau_{n+1} - t < 1.00, \\
0 & 1.00 \leq \tau_{n+1} - t 
\end{cases}
\]

provides four categories of imminence of a regime change. In Fig. 2(b) is illustrated the behavior of \( \Omega_n \), together with the \( x \) variable, as a function of the current time. To apply a classification technique, we relate the three-dimensional vector \( r(t) = (x(t), y(t), z(t)) \) that indicates the state of the system to the corresponding value of \( \Omega \). Thus, from \( r(t) \), we can determine a label that indicates an approximation to TITE.

**IV. RESULTS**

**A. Simulation details**

As a classifier, we use ensembles of up to 40 MLPs with two hidden layers. The two main training datasets, with \( 3 \times 10^5 \) samples, have been generated to the MLPs training step, one for each objective. The first one, used to the regime duration prediction, is composed by the pairs \( (V_n, D_n) \) and respective labels from \( 3 \times 10^5 \) regimes. In the other case, applied for the end of regime’s prediction, for each regime (from 3000), we randomly selected 100 \( r \)’s together with their imminence values \( \Omega \). From these data, we obtained the statistical information shown in Tables I and II. Based on the statistics obtained for the regimes duration, we assign 12 categories. The last one, label 11, concatenates all quantities with \( k_n \), larger than 11. Specifically for machine learning training, 40 subsets with 2500 samples have been randomly generated from the main training sets in

<table>
<thead>
<tr>
<th>Lab.</th>
<th>( k_n )</th>
<th>Occ. (%)</th>
<th>ARD(( t_d ))</th>
<th>MD(( t_d ))</th>
<th>Hits</th>
<th>(-1)</th>
<th>(+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>47.01</td>
<td>1.037</td>
<td>0.027</td>
<td>99.88</td>
<td>...</td>
<td>0.12</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>22.62</td>
<td>2.057</td>
<td>0.046</td>
<td>99.60</td>
<td>0.27</td>
<td>0.13</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>12.12</td>
<td>2.974</td>
<td>0.050</td>
<td>99.09</td>
<td>0.58</td>
<td>0.34</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>6.90</td>
<td>3.858</td>
<td>0.053</td>
<td>99.03</td>
<td>0.50</td>
<td>0.48</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4.23</td>
<td>4.722</td>
<td>0.053</td>
<td>98.34</td>
<td>0.85</td>
<td>0.81</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>2.61</td>
<td>5.575</td>
<td>0.055</td>
<td>98.13</td>
<td>0.71</td>
<td>1.16</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1.65</td>
<td>6.422</td>
<td>0.058</td>
<td>97.55</td>
<td>0.85</td>
<td>1.60</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>1.08</td>
<td>7.257</td>
<td>0.057</td>
<td>92.38</td>
<td>0.96</td>
<td>0.67</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0.66</td>
<td>8.089</td>
<td>0.057</td>
<td>91.79</td>
<td>0.15</td>
<td>8.06</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>0.42</td>
<td>8.917</td>
<td>0.058</td>
<td>90.31</td>
<td>0.15</td>
<td>9.54</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>0.27</td>
<td>9.748</td>
<td>0.061</td>
<td>98.42</td>
<td>1.23</td>
<td>0.34</td>
</tr>
<tr>
<td>11+</td>
<td>11</td>
<td>0.42</td>
<td>&gt;10.578</td>
<td>...</td>
<td>89.85</td>
<td>10.15</td>
<td>...</td>
</tr>
</tbody>
</table>
both applications, namely $k_{n}$ and $\Omega_{n}$. Each subset was used for a different MLP. It is worth to mention that only labels and vectors are used in the subsets. Thus, we decided not to normalize the data to maintain all the characteristics of the system.

To test each classification strategy, 50 sets with $10^{5}$ elements were generated from different random initial conditions. The test sets are in the same format as the training sets and at this stage, the labels are used only to check the MLP answers. Due to the random aspects of the learning process, to ensure the robustness of the method, a new training was performed before each test, and we calculated the average success rate on the 50 test sets for all ensembles of the investigated MLPs. The ensemble answers were defined by voting. Each MLP in the ensemble can classify differently the same vectors. The ensemble voting combines the answers obtained with an ensemble of 38 MLPs. The last three columns of Table I correspond only to correct predictions. This information corresponds only to correct predictions.

**TABLE II.** Data about the prediction of the end of regimes. The first three columns show the values of the imminence $\Omega_{n}$ obtained using Eq. (4), and the related time interval ($\Delta_{n} = t_{n} - t_{n-1}$) to the end of regime (TITE), associated with the occurrence (Occ.) of each class. These data were obtained from the main training set with $3 \times 10^{5}$ samples. In the second part of the table, we show the best results obtained with an ensemble of 23 MLPs applied to 100 test sets with $10^{5}$ vectors. The last four columns are a confusion matrix, lines are the correct class and columns are the predicted. The column with hits repeats the main diagonal of the confusion matrix. This information corresponds only to correct predictions.

<table>
<thead>
<tr>
<th>Training data inform.</th>
<th>Prediction results (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Omega_{n}$ Occ. (%) TITE Hits</td>
<td>0 28.61</td>
</tr>
<tr>
<td>$\Delta_{n} \geq 1.0$</td>
<td>1 1.03 0.1 $\Delta_{n} &lt; 1.0$</td>
</tr>
<tr>
<td>$0.1 \leq \Delta_{n} &lt; 0.1$</td>
<td>2 7.53 0.01 $\Delta_{n} &lt; 0.1$</td>
</tr>
<tr>
<td>$0.01 \leq \Delta_{n} &lt; 0.1$</td>
<td>3 0.83 $\Delta_{n} &lt; 0.1$</td>
</tr>
</tbody>
</table>

ARD increases linearly, as demonstrated in Fig. 3 using data from Table I. Label 11 includes all number of local maxima (or minima) larger than 11, which represents 0.42% of the samples. By a proper definition of thresholds, events with label 11 can be considered rare to occur. In fact, we observed (not shown) that the amplitude of the longest regime durations exceed four standard deviations from the average amplitude of the events. Thus, events with label 11 are rare. The relation between $k_{n}$ and ARD can be approximated by the curve shown in Fig. 3. By regression, we obtain the equation

$$D_{e}(k_{n}) = 0.862k_{n} + 0.341 \tag{5}$$

so that $D_{e}(k_{n})$ is the duration of the regime. All quantities are given in respect to the $n$th regime.

In both cases, namely, in predicting the regime change and its duration, we tested ensembles from 2 to 40 MLPs, as well as the performance of a single network. The classification methods were applied to 50 test sets with $10^{5}$ samples. To determine the regimes duration, the best average performance is 99.28% of correct predictions obtained with an ensemble of 38 MLPs. The last three columns of Table I show the rate of classification success and errors in each label. It should be noted that the errors in this application are only one class (up or down), and the error rate is < 1.00% in 15 of the 22 cases presented. We notice that we can predict with high accuracy the regimes duration, especially the longest ones, grouped with the label 11. Considering the relation between $k_{n}$ and ARD [see $D_{e}(k_{n})$], we can predict the permanence of a specific regime with low error.

For the prediction of the regimes changes, the best result is 97.80% using an ensemble of 40 MLPs. The specific success rates of each eminence class is shown in the last four columns of Table II. Also in this application, the errors are very small, < 2.50% in all cases, highlighting the most eminent class $\Omega = 3$ with an accuracy of 99.23%. This imminence value corresponds to the time interval to the end of the regime, which is < 0.01 and occurs only 0.83% of the samples in the main training set, as can be observed in the second column of Table II. Figure 4 displays the classification accuracy for both proposals according to the number of networks in the

**FIG. 3.** Statistical data from the main training set with $3 \times 10^{5}$ regimes. ARD, in Lyapunov times ($t_{i}$), as a function of $k_{n}$ is represented by red circles. The blue line is an approximation to the regime duration as a function $k_{n}$ according to $D_{e}(k_{n})$ from Eq. (5).
is calculated (analogous definition for variable $y$). The cumulative errors are plotted as red continuous curves in Figs. 5(a) and 5(b). The ESN was able to accurately reproduce, in the best scenario, approximately $10\,t_L$ in variable $x$ and four regime changes, as can be observed in Fig. 5(b). The cumulative error $\text{MSE}_x$ starts to increase for times larger than $10\,t_L$. For the variable $y$, we replicate the curve for $7\,t_L$ and got small differences up to $8\,t_L$, as shown in Fig. 5(a). However, the cumulative error $\text{MSE}_y$ starts to increase for shorter times, namely, for times larger than $5\,t_L$.

One advantage of the ESN, compared to the training of an ensemble of MLPs, was the computing time. While for the imminence, the training of one MLP is 27 times slower than for the ESN, for the regime duration it is 28 times slower. In addition, since we consider an ensemble of MLPs, parallel computing is highly recommended. However, ESN is not superior to the classification strategies to predict the remaining time interval in the current regime or the duration of the next one. When a MLP is trained, it can predict with great accuracy, from an unique vector, the imminence $\Omega_L$ to the end of the current regime or the $k_0$ to the next regime.

Furthermore, our results are the best obtained concerning the correct generation of the time series for longer time intervals. Other configurations of ESN, using different number of oscillators in the reservoir, or changing the quantity of points for the training, entail only for shorter time intervals an agreement between the generated series and reference data.

To finish this last analysis, we would like to furnish some technical details, so readers can reproduce our results. The network was created using the easyesn library, with the following hyperparameters: reservoirs: 600; density: 0.2; leaking rate: 0.08; regression parameters: $1 \times 10^{-5}$; solver: lsqr; and spectral radius: 0.2.

ensemble. Even though the accuracies are astonishingly good when using just one network, it is noted that they increase rapidly with the number of networks, until stabilization occurs near the maximum value.

C. Reproducing the chaotic times series

The success of the classification technique to predict regime changes and durations becomes convincing when compared to other techniques, which use machine learning in the generation or reconstruction of chaotic time series. Thus, in addition to the prediction using MLPs, we performed experiments using ESN, a kind of RNN, to reproduce chaotic time series. Our purpose with these experiments is to replicate Lorenz’s chaotic time series without using Eqs. (1)–(3). To do so, a dataset with 3500 points along the chaotic Lorenz attractor was used in the training stage. The time interval of the training set is approximately $45.7\,t_L$. We would like to reproduce not only the chaotic behavior, but also the specific drawn curves by each variable in the time series. Our focus is in the variables that change signal, with emphasis in $x_i$, because it defines the regimes. After training, we start to count and compare, during $18\,t_L$, the time series generated by ESN with the reference data, obtained using RK4.

As an example, Figs. 5(a) and 5(b) show the comparison between the ESN and the reference data for the generation and reconstruction of the chaotic time series for the variables $x$ and $y$, respectively. Orange continuous curves display the errors, point by point, obtained from

$$E_i(t = \delta) = |x(\delta)_{\text{RN}} - x(\delta)_{\text{ESN}}|.$$  

(6)

Analogous definition for variable $y$. The error is calculated at discrete times $t = \delta$, where $\delta = 10^{-3}$ is the time step and $i = 1, \ldots, m$, is the number of steps considered. We start to compute the errors from the first point of the time series generated by the ESN. Interesting to observe in Figs. 5(a) and 5(b) is that, along the reproduction of the time series, these errors tend to increase when a regime change occurs. This becomes more evident when the mean square error (MSE)

$$\text{MSE}(t = m\delta) = \frac{1}{m} \sum_{i=1}^{m} \left[ E_i(\delta) \right]^2.$$  

(7)
The optimization of hyperparameters was performed manually and empirically. For each parameters configuration, 10 rounds were performed, and the result with the lowest mean square error was chosen.

V. CONCLUSION

This paper shows how it is possible to use classification strategies and machine learning techniques to predict events in chaotic time series and to generate with accuracy a few hundred steps of such series. Specifically for Lorenz’s time series, we show a correlation between the amount of local maxima for each regime (left and right) with the regime duration, allowing us to treat these characteristics as a classification problem and solve it with MLPs ensembles. High accuracies are obtained for the predictions, even for the longest regimes, which include Lyapunov times larger than 10. ESN has proved to be a good choice to generate few hundreds time steps with great accuracy and performance. The classification technique showed to be much superior than the generation or reconstruction of chaotic time series to predict regime changes and durations.

The dynamics of the Lorenz systems is relatively simple and widely known, and this is the main reason we have chosen this model. We know very well the underlying dynamics of this system, and this allows us to apply and check new methods like the classification strategy and machine learning. We are not showing new results about the Lorenz system itself, but a novel way to make predictions with high accuracy in such a system. This includes the prediction of rare events, which, in our approach, are the large amplitudes of the regime durations.

ACKNOWLEDGMENTS

This work was supported by the Max-Planck Institute for the Physics of Complex Systems, Dresden, in the framework of the Advanced Study Group on Forecasting with Lyapunov vectors. E.L.B. and M.W.B. acknowledge CNPq for financial support (Grant No. 310792/2018-5).

REFERENCES

25. See https://github.com/kaleku/easyesn for information about the easyesn library.