Entropy for Time Series Forecasting

João Costa
Fakulteta za matematiko in fiziko
joaocostamat@gmail.com

António Costa
ESN Paris
antoniocbscosta@gmail.com

Klemen Kenda
Jožef Stefan Institut
klemen.kenda@ijs.si

João Pita Costa
IRCAI
joao.pitacosta@quintelligence.com

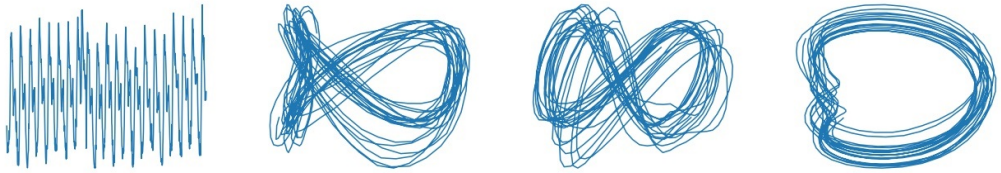

Figure 1: Sample of the time series and projections of the embedding - This plot gives us a geometrical representation of the theory involved in section 3 and shows the reconstructed state space of the given time series. This can be obtained by using Takens' embedding to reconstruct the time series y, given in figure a), as the markovian system Y_K with K time delays and then use Principal Component Analysis in order to perform the change of basis of the data. The obtained projections b), c) and d) attain the dynamics of the system, which gives us the possibility to predict the time series with higher efficiency.

ABSTRACT

In this paper, we present the exploitation of a method to extract information from microscopic samples of time series data in order to provide a representation of optimized stability to a chaotic system [1]. The main goal of this approach is to predict the dynamics of a time series and therefore develop optimized forecasting algorithms. First, we study how to increase the predictability of a system and second, we develop a Deep Learning Algorithm, namely an LSTM, that can recognize patterns in sequential data and accurately predict the future behaviour of a time series.

KEYWORDS

Recurrent Neural Networks, LSTM, Entropy, Markov Chain, Clustering, Time Series

1 INTRODUCTION

Given its intrinsic nature, mathematics concerns with the construction of formal statements and proofs relating the different concepts within it. Its methods are used in countless ways and effectively model the shape of our world. But how is it possible to shape the unknown? Motivated by this question and the upmost need for finding ways of optimizing water resources for future generations, there has been a great development on the study of dynamical systems based on, for example, (Shannon) entropy [9] and phase space reconstruction [4]. In this paper, we provide an approach to water resource management using Deep Learning and Chaos Theory, by studying the dynamics of a time series using the 2 main ideas cited before. This study was developed for the H2020 NAIADES Project [2] with data collected from the Municipality of Alicante (Spain). We will present this study for the Autobus Dataset, related to the Bus Station Areas in Alicante.

Information Society 2020, 5–9 October 2020, Ljubljana, Slovenia

2 STATIONARY AND CHAOTIC NATURE

2.1 Dickey-Fuller Test for Stationarity

In order to proceed with the theory involved in the method, it is necessary to understand the behaviour of the time series and its sensitivity to initial conditions. For studying time series' stationarity, one can use the Augmented Dickey-Fuller test, which is a type of statistical test called a unit root test, where generally the null hypothesis is that the time series can be represented by a unit root, which means that for $y = \{y_t\}_{t=1}^{T}$, the information at point y_{t-1} does not provide us the ability to predict y_t. In our case, we obtained that the p-value of the test was 0, so the null hypothesis was rejected and the time series has no unit root. Therefore, it is stationary and the time delays will provide important information for predicting the dynamics of the time series.

2.2 Lyapunov exponents for understanding chaotic nature

The Lyapunov Exponent is a quantifier for the sensitivity of the time series on initial conditions and therefore for its chaotic nature. The main idea is to select an array of nearest neighbors, i.e, points at minimum distance, and calculate its trajectories in time. By doing so, we can then obtain an average of this divergence exponent which gives us the Lyapunov Exponent. Since the system is bounded, the divergence is also bounded and will reach a plateau after a certain number of timesteps. In our case, the Lyapunov Exponent, given as the initial slope, is ≈ 518 and the initial growth is exponential, as can be seen in figure 5. Therefore, the time series is of a chaotic nature.

3 MAXIMUM PREDICTABILITY

Given the high variability of any chaotic system, it is hard to capture the whole set of variables that model the state space. This is characteristic of a non-Markovian system which is highly unpredictable. How do we surpass this issue?

Takens' Embedding Theorem [8] tells us that, under certain conditions, it is possible to use past data to reconstruct a Markovian system, thus giving us the possibility to model the initial time series with higher efficiency. We start by considering a set of ODEs $x = (\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_D)$ and the d-dimensional time series $y(t)$ of duration T which is a set of incomplete measurements of x given by a measure M, i.e., $y = M(x)$. Then, in order to calculate the number of K time delays to feed the LSTM with, the d-dimensional measurements are lifted into the state space $Y_K \in \mathbb{R}^{d\times K}$ consisting of the previously referred K time delays [3]. It is possible to quantify the chaotic measure of the system Y_K by calculating the entropy resulting from clustering. This can be done by partitioning the $d \times K$-dimensional space into N Voronoi cells using K-Means clustering. Having partitioned the state space Y_K, the reconstructed dynamics are encoded as a row-stochastic transition probability matrix $P = [P_{ij}]_{i,j}$ which relates increments on the state-space density p in the following way

$$p_i(t + \delta t) = \sum_j P_{ji} p_j(t). \tag{1}$$

The entropy rate of the initial time series $y(t)$ is then approximated by estimating the entropy rate (Figure 3) of the associated Markov chain on the different time delays K using Kolmogorov's definition

$$h_{p_N}(K) = -\sum_{i,j} \pi_i P_{ij} \log P_{ij}, \tag{2}$$

where π is the estimated stationary distribution of the Markov chain P. This approximation gives an estimate for the conditional entropies (Figure 6), i.e., for a discrete state with delay vectors $\vec{y}^K = \{\vec{y}_i, \ldots, \vec{y}_{i+K-1}\}$, the entropy of the Markov chain provides an estimate for the conditional entropy,

$$\begin{aligned} h_{p_N}(K) &\approx \langle -\log[p_N(y_{i+K}|y_i, \ldots, y_{i+K-1})]\rangle \\ &= H_{K+1}(N) - H_K(N) \\ &= h_K(N), \end{aligned} \tag{3}$$

where H_K is the Shannon Entropy of the sequence obtained by partitioning the $\vec{y}$ space into N partitions.

4 MODEL ARCHITECTURE

4.1 LSTM

Long Short Term Memory (LSTM) Networks are a special type of Recurrent Neural Networks (RNN) which rely on gated cells that control the flow of information by choosing what elements of the sequence are passed on to the next module. This idea was introduced in order to surpass the vanishing gradient problem in conventional RNNs [7]. At each time t, consider f_t as the forget gate, i_t as the input gate and o_t as the output gate, which are functions that depend on the output of the previous LSTM module, given by h_{t-1} and on the input of the current timestep, given by x_t. Then, the next figure shows a representation of how a single LSTM cell performs its computations. The computations

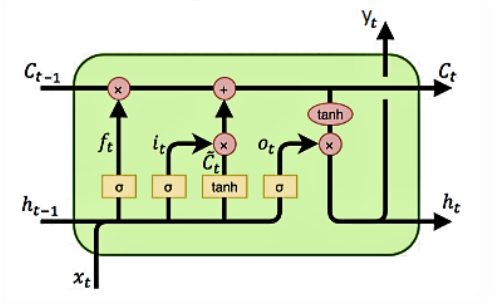

Figure 2: An LSTM performs the following ordered computations: The first step is to forget their irrelevant history. Then, LSTMs perform computation to decide on relevant parts of new information and based on the previous two steps, they selectively update the internal state. Finally, an output is generated.

shown in this figure can be mathematically represented as

$$\begin{aligned} f_t(x_t, h_{t-1}) &= \sigma(w_{f,x}^T x_t + w_{f,h} h_{t-1} + b_f) \\ i_t(x_t, h_{t-1}) &= \sigma(w_{i,x}^T x_t + w_{i,h} h_{t-1} + b_i) \\ o_t(x_t, h_{t-1}) &= \sigma(w_{o,x}^T x_t + w_{o,h} h_{t-1} + b_o), \end{aligned} \tag{4}$$

where $w_{f,x}, w_{i,x}, w_{o,x} \in \mathbb{R}^d$ are weight parameters and σ is an activation function.

4.2 Our approach

The core idea is to take a list of k training sets $Q_0, Q_1, \ldots, Q_{k-1}$ and testing sets $P_0, P_1, \ldots, P_{k-1}$ in order to generalize the model and do the best estimation for the time series. This is based on translating the testing sets' partitions along the time series, where the first partition $P_0 = \{p_0^0, \ldots, p_0^n\}$ is taken from the zeroth point of the time series data and the last partition $P_{k-1} = \{p_{k-1}^0, \ldots, p_{k-1}^n\}$ until the last point of the time series data and

$$|P_i| = \frac{|y|}{k}, \forall i \in \{0, \ldots, k-1\} \tag{5}$$

where $|y|$ stands for the cardinality of the time series y. This procedure yields k models which will use each of the training sets to make predictions on the respective test sets. Given the erratic nature of the data, which was taken in 15 and 30 minutes samples, a resampling to 30 minute delays had to be done on the 15 minutes delay data points and a masking was added to the time series in order to neglect NaN values that could be created from resampling. Therefore, a masking layer was added and the model is composed by 3 other layers $\mathcal{L}_{n_1}$, $\mathcal{L}_{n_2}$ and $\mathcal{L}_{n_3}$, where $n_1 = n_3 = 1$ (we have a univariate timeseries) and $n_2 = 64$, since it gave the best results in cross validation. A dropout regularization of 0.1 was added for better approximation of training and validation errors and the batch size was set to 128. The mean squared error for the predictions on the training set is ≈ 0.00115 and for the testing set is ≈ 0.00236. One can address the capacity of the model whose predictive results are shown in figure 4.

5 FORECASTING

5.1 Forecasting Methods

Consider a time series $T = \{t_1, \ldots, t_N\}$. The forecasting process can be done in 3 ways:

(1) iterated forecasting

(2) direct forecasting
(3) multi-neural network forecasting

Process number (1) is based on "many-to-one" forecast for which

$$t_{n+1} \approx \mathcal{F}(t_i, \ldots, t_{i+n-1}), i \in \{1, \ldots, N-n\}. \quad (6)$$

Then, a K-step forecast can be iteratively obtained by

$$\hat{t}_{N+j} := \mathcal{F}(\hat{t}_{N+j-n+1}, \ldots, \hat{t}_{N+j-2}, \hat{t}_{N+j-1}), j \in 1, \ldots, K. \quad (7)$$

Process number (2) can be characterized by training a "many-to-many" function $\mathcal{F}$ for which

$$(t_{i+n}, \ldots, t_{i+n+K-1}) \approx \mathcal{F}(t_i, \ldots, t_{i+n-1}), \quad (8)$$

where $i \in \{1, \ldots, N-n-K+1\}$. We can obtain a K-step forecast by

$$(\hat{t}_{N+1}, \ldots, \hat{t}_{N+K}) := \mathcal{F}(t_{N-n+1}, \ldots, t_N). \quad (9)$$

Finally, process (3) is defined by k "many-to-one" functions $\mathcal{F}_1, \ldots, \mathcal{F}_k$ which hold the following relationship

$$\begin{aligned} t_{i+n} &\approx \mathcal{F}_1(t_i, \ldots, t_{i+n-1}) \\ &\vdots \\ t_{i+n+K-1} &\approx \mathcal{F}_k(t_i, \ldots, t_{i+n-1}), \end{aligned} \quad (10)$$

where i ranges from 1 to $N-n-K+1$. Process (1) does not require k a propri while both process (2) and (3) are dependent on the choice of k.

5.2 Our Approach

We chose to do a Direct Forecasting for the next 7 days by taking the last test set partition P_{k-1} and did a prediction on this test set. Although forecasting seems pretty motivating, by choosing a partition that attains more characteristics of the time series, one can achieve even better results. The achieved forecast can be seen on Figure 8 and compared with a 7 days sample on Figure 7.

6 RESEARCH METHODS

6.1 Time Series Reconstruction

Consider the time series y with duration T as given in section 2. The idea is to add K time delays to y in order to obtain a $(t-K) \times Kd$ space $Y_K \in \mathbb{R}^{d \times K}$ and further partition Y_K using k-means Clustering into N Voronoi Cells.

6.2 Entropy Calculation

Consider the N Voronoi Cells given as the number of partitions of Y_K and consider the joint probability $p(c_{i_1}, \ldots, c_{i_l}), \{i_1, \ldots, i_l\} \in \{0, \ldots, N-1\}$. Then, the Shannon Entropy [6] is given by

$$H_l = -\sum p(c_{i_1}, \ldots, c_{i_l}) \log p(c_{i_1}, \ldots, c_{i_l}) \quad (11)$$

and the conditional probabilites are given by

$$p(c_{i_{l+1}} | c_{i_1}, \ldots, c_{i_l}), \quad (12)$$

where ci_{l+1} is the next Voronoi Cell after c_{i_l}. We can calculate the entropy rate growth by considering the conditional probabilities of the system given the previous l cells, when visiting the $(l+1)$-th cell, via

$$h_l = \langle -\log[p(c_{i_{l+1}} | c_{i_1}, \ldots, c_{i_l})] \rangle = H_{l+1} - H_l \quad (13)$$

Taking the supremum limit over all possible partitions P of Y_K, we obtain the Kolmogorov-Sinai invariant of the system,

$$h_{KS} = \sup_P \lim_{l \to \infty} h_l(P). \quad (14)$$

6.3 Data and Code Git Repository

The complete work can be found in:
https://github.com/johncoost/JoaoModelsForAlicante.

7 PLOT OF RESULTS

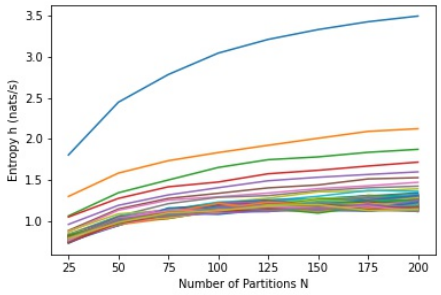

Figure 3: Entropy Rate h - The entropy rate h is given as the function of the number of partitions N for increasing number of delays K (given by the different colors in a descendent mode). It is possible to observe that the entropy rate is a non-decreasing function on the number of partitions N. The idea is to choose the value of N for which the entropy is maximum so that we have the maximum possible information about the system's dynamics.

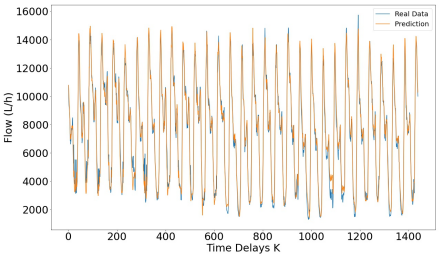

Figure 4: Prediction on the last test set - This shows a sample of the last test set and its prediction. We can observe the effectiveness of the LSTM in modelling the given time series by having a deep understanding of its inherent dynamics.

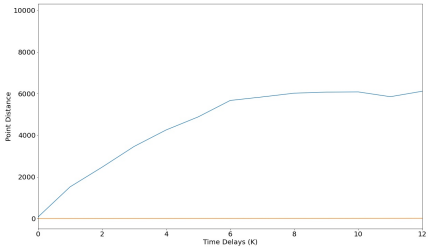

Figure 5: In this figure, we can understand the initial exponential growth on distance between points (given in blue), relative to a curve of slope 1 (given in orange).

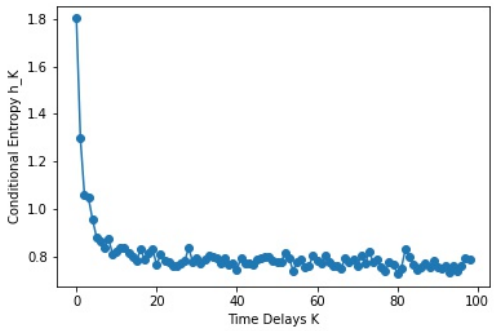

Figure 6: Conditional Entropies - In this plot we can see the entropy rate for number of partitions $N = 200$ which maximizes this entropy. This function reaches a plateau at ≈ 24 timesteps, which gives us an idea about which is the optimal K to choose. Given that we have 30 minutes timesteps, this plot shows that the optimized time delay is of 12h which corresponds to the day and night cycles

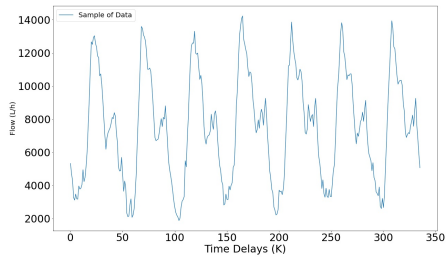

Figure 7: 7 Days Sample

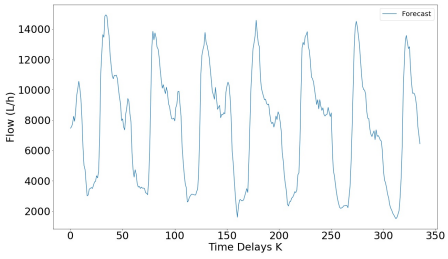

Figure 8: Prediction for 7 days ahead - Actual forecast using 336 timesteps that gives a 7 day future forecast sample using the LSTM model and direct forecasting. It is possible to observe that, as in figure 6, the values vary between ≈ 2000 to ≈ 14000 flow units and the essential dynamics of the time series were understood by the LSTM.

8 CONCLUSION

Having developed all the necessary machinery for constructing a coherent forecasting engine, we come to the conclusion that although the cardinality of the time series data was relatively small, the obtained results are promising and the model will certainly show satisfying results when applied in real time.
For the future, we want to continue developing the project by building other algorithms, such as Transformer neural network, that would provide even better results. Another idea is to use weather data and build a multivariate LSTM that optimally gives better results than the univariate one.

9 ACKNOWLEDGMENTS

I greatly thank to António Carlos Costa for working in cooperation and giving me the possibility to use the powerful machinery he built in order to obtain the desired K time delays and understand the complex dynamics of the system. Also, to the NAIADES team at Jožef Stefan Institute for all the knowledge exchange and, in particular, to Klemen Kenda for giving me the possibility of writing this paper and João Pita Costa for giving me insights on how to write and structure the paper.

This paper is supported by European Union's Horizon 2020 research and innovation programme under grant agreement No. 820985, project NAIADES (A holistic water ecosystem for digitisation of urban water sector).

REFERENCES

[1] Tosif Ahamed, Antonio Carlos Costa, and Greg J. Stephens. 2019. Capturing the continuous complexity of behavior in c. elegans. (2019). arXiv: 1911.10559 [q-bio.NC].

[2] 2019-2022. Cordis, "naiades project". In CORDIS. \url{https://cordis.europa.eu/project/id/820985}.

[3] Antonio Carlos Costa, Tosif Ahamed, David Jordan, and Greg Stephens. 2021. Maximally predictive ensemble dynamics from data. (2021). arXiv: 2105.12811 [physics.bio-ph].

[4] Vicente de P. Rodrigues da Silva, Adelgicio F. Belo Filho, Vijay P. Singh, Rafaela S. Rodrigues Almeida, Bernardo B. da Silva, Inajá F. de Sousa, and Romildo Morant de Holanda. 2017. Entropy theory for analysing water resources in northeastern region of brazil. *Hydrological Sciences Journal*, 62, 7, 1029–1038. DOI: 10.1080/02626667.2015.1099789. eprint: https://doi.org/10.1080/02626667.2015.1099789. https://doi.org/10.1080/02626667.2015.1099789.

[5] David A. Dickey and Wayne A. Fuller. 1979. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74, 366a, 427–431. DOI: 10.1080/01621459.1979.10482531. eprint: https://doi.org/10.1080/01621459.1979.10482531. https://doi.org/10.1080/01621459.1979.10482531.

[6] Robert M. Gray. 2011. *Entropy and Information Theory*. (2nd edition). Springer Publishing Company, Incorporated. ISBN: 9781441979698.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9, 8, 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[8] Floris Takens. 1981. Detecting strange attractors in turbulence. In *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 366–381. DOI: 10.1007/bfb0091924.

[9] Peyman Yousefi, Gregory Courtice, Gholamreza Naser, and Hadi Mohammadi. 2020. Nonlinear dynamic modeling of urban water consumption using chaotic approach (case study: city of kelowna). *Water*, 12, 3. ISSN: 2073-4441. DOI: 10.3390/w12030753. https://www.mdpi.com/2073-4441/12/3/753.