A Strategy for Time Series Prediction Using Segment Growing Neural Gas

Jorge R. Vergara[†‡], Pablo A. Estévez[†‡]
[†]Department of Electrical Engineering, University of Chile, Santiago, Chile
[‡]Millennium Institute of Astrophysics, Santiago, Chile
Email:{jorgever, pestevez}@ing.uchile.cl

***Abstract*—Segment Growing Neural Gas (Segment-GNG) has been recently proposed as a new spatiotemporal quantization method for time series. Unlike traditional quantization algorithms that are prototype-based, Segment-GNG uses segments as basic units of quantization. In this paper we extend the Segment-GNG model in order to deal with time series prediction. First Segment-GNG makes a quantization of the trajectories in the state-space representation of the time series. Then a local prediction model is associated with each segment, which allows us to make predictions. The proposed model is tested with the Mackey-Glass and Lorenz chaotic time series in one-step ahead prediction tasks. The results obtained are competitive with the best results published in the literature.**

I. Introduction

Time series prediction has played an important role in many fields of practical application such as economics, signal processing, traffic flow, biology, physics, meteorology, among others [1]. Given a set of observed data, a model of the underlying dynamics of a system is learned and then used to predict new data. Many prediction methods have been proposed such as adaptive prediction [2], support vector machine (SVM) [3], polynomial estimation [4], and neural networks [5]. The aim of these global approximation methods is to identify a structure, select input variables and find parameters embedded in the structure. The drawback of these global methods is that if new information is taken into account then the parameters of the model may change, and a long time may be required for parameter re-estimation. On the other hand, determining all parameters is very complex, and in many approaches a priori subset selection of input variables is assumed [6].

Local prediction approaches have attracted interest from many researchers due to their good performance and flexibility [7]–[9]. In this kind of approach, the time series is first embedded in a state space, and its underlying dynamics is inferred by local approximations. The assumption behind local models is that the underlying dynamics is locally smooth [10]. From our point of view, to correctly infer the underlying dynamics, the local prediction approaches must have three main properties:

(1) Split data into homogeneous clusters: The state-space quantization must preserve the topological (spatial) distribution of the data. VQ [11], SOM [12], K-means [13], NG [14], GNG [15], among others, are traditional algorithms used for time series quantization. Usually they reduce a large volume of vectors to a smaller number of prototypes or codevectors which are the most representatives of the initial distribution. However the drawback is that they are highly dependent on the number of clusters. Over or under estimation of the correct number can cause the appearance of clusters containing regions with different dynamics or with too few patterns to train the local prediction models.

(2) Determine and quantize temporal regions: The dynamical system embedded in a state space could be decomposed into local subsystems. These subsystems are represented by temporal sequences of data. Some quantization methods seek the spatial correlation in the embedded space of sequences or temporal patterns of the time series. These quantization methods add a context to each quantization unit, allowing them to discriminate data sequences spatially similar but with different dynamics. Some quantization methods in the literature are: SOM with Temporal Activity Diffusion (SOMTAD) [16], Merge-SOM [17], γ-SOM [18], Merge-GNG [19], γ-GNG [20], among other.

(3) Transfer temporal information between local models: The ability to transfer information between different subsystems is fundamental in order to approximate the global dynamical system. There exist recursive quantization models such as Recurrent Self-Organizing Maps (RSOM [21]), Recursive Self-Organizing Maps (RecSOM [22]), Sequential Activation, Retention and Delay Network (SARD-NET [23]), Feedback SOM (FSOM [24]), that create and combine quantization maps of two different times. The goal of this strategy is to be able to quantize transitions between dynamical subsystems (prototypes).

Currently, the vast majority of time series quantization methods are designed either to quantize the global dynamics (property 1) or local dynamics (property 2) of the time series, however, the temporal quantization between local models (property 3) has not been extensively studied [25].

Recently, Segment-GNG has emerged as an extension of the GNG model [26]. Its two main novelties are the replacement of prototypes by segments as units of quantization, and the use of a register of temporal connections between segments. The Segment-GNG is an attractive tool for the prediction of time series, due to the following properties: (1) including iteratively segments in the quantization process allows us to adapt to the complexity of the space and also efficiently split data into homogeneous clusters. (2) The use of segments allows us to quantize temporal patterns (directions of trajectories) in state space. (3) The temporal connection between segments allows us to quantize temporal interactions between segments.

978-1-5090-6638-4/17/$31.00

Exploiting the advantages of Segment-GNG in the quantization of time series, the present work proposes a new methodology based on Segment-GNG time series prediction.

The rest of this paper is divided into 5 sections: Section 2 presents the fundamental concepts used in this paper. Section 3 presents our method for time series prediction based on Segment-GNG. Section 4 shows the results of our method for one-step ahead prediction in two chaotic times series. Finally, section 5 shows the conclusions and future work.

II. Background

A. Time series prediction

A time series is an ordered sequence $\{y_t\}_{t=1}^{n}$ of n observations generally sampled in fixed time intervals of size[1] δ. The dynamics of the underlying process generating the sequence is generally unknown. The goal of time series analysis is to estimate a model that best fits the time series at hand. Having a model makes possible to know the behavior of the times series and to predict the next samples [27]. The multiple step ahead prediction is usually estimated using a set of past observations. The value h is known as the prediction horizon and the general expression of the multi-step ahead prediction is described as follows:

$$\hat{y}_{t+h} = f\left(y_t, y_{t-1}, \cdots, y_{t-L_{max}}\right), \tag{1}$$

where $f(\bullet)$ is a linear or nonlinear model that predicts the sample h-step ahead using the current sample and L_{max} past samples. Many different techniques have been proposed to find a model such as: linear models, polynomial models, neural network models, among others.

B. Delay Coordinate Embedding

The basis of chaotic time series prediction is that the reconstruction of a state space from a one-dimensional time series is equivalent to the original state space of a system [1]. According to Takens' embedding theorem [28], it is possible to reproduce entirely the properties of such a system (topology and temporal structure) starting from a one-dimensional time series. The time series corresponds to a sequence of scalar measurements of the state space or a single state variable, y_t. To embed a time series, a delay coordinate vector is constructed as follows: $\boldsymbol{\phi}_t = [y_t, y_{(t-\zeta)}, y_{(t-2\zeta)}, \cdots, y_{(t-(m-1)\zeta)}]$, where the delay ζ and dimension m are the embedding parameters. Although the embedding theorems do not provide a way to estimate these parameters, there are some heuristic methods to do so. The parameter ζ is usually estimated by seeking the delay that provides the first minimum of the average mutual information [29], while the dimension m is estimated by the false nearest neighbor algorithm [30].

C. Local linear prediction model

Usually the local models on a m-dimensional embedding space are autoregressive models with L_{max} past samples for each component of the state-space reconstruction [31]. Deterministic predictions assume that the reconstructed trajectory in the state-space is governed by a continuous mapping F between the current state and the future state [10]. In the time series space an autoregressive model is defined as:

$$\hat{y}_{t+1} = \alpha_0 + \sum_{i=1}^{m}\sum_{j=0}^{L_{max}} \alpha_{ij} y_{t-j-(i-1)\zeta}. \tag{2}$$

The parameters α_{ij} can be estimated by least squares or with the LMS algorithm.

D. Performance metrics

In the time series literature, several metrics have been proposed to measure the ability of the model to predict a new data [32]. Depending on the time series prediction problem to be solved, there are two principal types of prediction-error metrics [33]: scale-dependent metrics and scale-independent metrics. The first are designed to analyze data regardless of how the prediction was produced. The range of values obtained by this kind of metrics depend on the magnitudes of the time series. Their main utility is to detect large errors particularly undesirable ones. The second type of metrics is designed to compare prediction performance with different data series. With this aim, each error is divided by the error obtained using some benchmark method of prediction [33]. In this work, the results are evaluated using two different metrics: the scale-dependent metric so-called Root Mean Square Error (RMSE) and the scale-independent metric so-called Normalized Mean Square Error[2] (NMSE). Formally, these metrics are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(y_t - \hat{y}_t\right)^2} \tag{3}$$

$$\text{NMSE} = \frac{\sum_{t=1}^{n}\left(y_t - \hat{y}_t\right)^2}{\sum_{t=1}^{n}\left(y_t - \bar{y}\right)^2}, \tag{4}$$

where y_t and $\hat{y}_t$ are the real and predicted sample at time t and $\bar{y} = {}^1\!/_n \sum_{t=1}^{n} y_t$ is the mean.

E. Segment Growing Neural Gas

Segment-GNG is a recently proposed method for the spatiotemporal quantization of time series [26]. Segment-GNG has two main differences with GNG: (i) the basic quantization unit of GNG (node or prototype) is changed to a segment (connection between two nodes), and (ii) Segment-GNG keeps a register of temporal connections among segments. The Segment-GNG algorithm seeks to approximate trajectories in the state space representation by linear segments. A segment S^i is defined as the line joining two nodes $\mathbf{s}_O^i, \mathbf{s}_F^i \in \mathbb{R}^m$ where $\mathbf{s}_O^i$ and $\mathbf{s}_F^i$ correspond to the initial and final points of segment S^i, respectively. Segment S^i is used to identify and quantize portions of trajectories in the state space that could be locally approximated by a linear segment. A portion of a trajectory $\{\phi\}_t^{t-\tau}$ is associated with the trajectory generated by the sample ϕ at time t and its τ past samples $\{\phi\}_t^{t-\tau} = \{\phi_{t-\tau}, \phi_{t-\tau+1}, \cdots, \phi_t\}$. The size of a portion of trajectory to be quantized by S^i is determined by the linearity of this portion. The latter is evaluated starting with $\tau = 1$,

[1] For notation, we assume in this work that $\delta = 1$. In this way, the sample at time t is y_t and its next sample is y_{t+1}.

[2] A value of NMSE $= 1$ corresponds to the value obtained by simply predicting the average.

where τ is increased iteratively until the area enclosed (AE) between the current portion of a trajectory $\{\phi\}_t^{t-\tau}$ and the line joining the extreme points of this portion $\left(\overline{\phi_{t-\tau}\phi_t}\right)$ reaches a certain threshold E_{max}, as show in Fig. 1.

To avoid the cost of computing AE accurately, Segment-GNG approximates it as the sum of distances between each sample in the portion of a trajectory $\{\phi\}_t^{t-\tau}$ and its projection onto the line $\overline{\phi_{n-\tau}\phi_t}$. To obtain the Best Matching Linear Segment (BMLS) for each trajectory portion $\{\phi\}_t^{t-\tau}$, a distance measure that evaluates two features of each linear segment S^i is used: (i) the closeness (d_{close}) between S^i and the trajectory portion $\{\phi\}_t^{t-\tau}$ is measured through the spatial distance. To measure the spatial distance, first the midpoint ($\boldsymbol{\rho}_t$) of the trajectory portion $\{\phi\}_t^{t-\tau}$ is estimated. Secondly, the distance between this midpoint and the linear segment S^i is computed by the square euclidean distance from a point to a line. (ii) the degree of parallelism (d_{parall}) between S^i and the line $\overline{\phi_{t-\tau}\phi_t}$ is measured through the cosine similarity. The cosine similarity between S^i and the line $\overline{\phi_{t-\tau}\phi_t}$ is computed as:

$$d_{parall}(S^i, \overline{\phi_{t-\tau}\phi_t}) = \frac{\Delta s^i \cdot \Delta\phi}{\|\Delta s^i\| \, \|\Delta\phi\|}, \tag{5}$$

where $\Delta s^i = \mathbf{s}_F^i - \mathbf{s}_O^i$ and $\Delta\phi = \phi_t - \phi_{t-\tau}$.

The combined distance measure used by Segment-GNG is the following:

$$D\left(\theta, \{\phi\}_t^{t-\tau}, S^i\right) = \left(\theta\ (1 - d_{parall}) + 1\right)\ d_{close}, \tag{6}$$

where $\theta \geq 0$ is a user-defined parameter that controls the trade-off between the cosine value (degree of parallelism) and the spatial distance (closeness). Once the BMLS is selected at time t, a temporal connection is created between this unit and the previous BMLS selected. For more details on the Segment-GNG algorithm see [26].

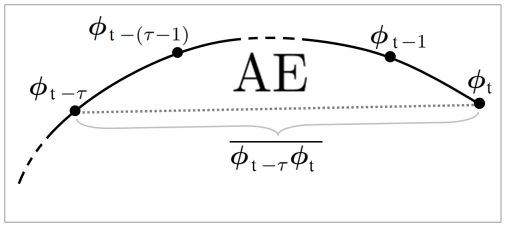


Fig. 1: Area enclosed (AE) between a trajectory portion in the state space representation $\{\phi\}_t^{t-\tau}$ and the line $\overline{\phi_{t-\tau}\phi_t}$.

III. Proposed Method

Traditionally local prediction models for time series do not convey information from one local models to another. An advantage of Segment-GNG is that its quantization is space-directional and temporal. The space-directional information gives us the direction of the trajectories quantized and the temporal information help specify which local prediction models should be considered for the prediction. This allows us to estimate the prediction using domains of two or more temporally connected segments. These characteristics of Segment-GNG are considered in our proposed method for making one-step ahead predictions. Our method is divided into two phases, as follow:

Training Phase

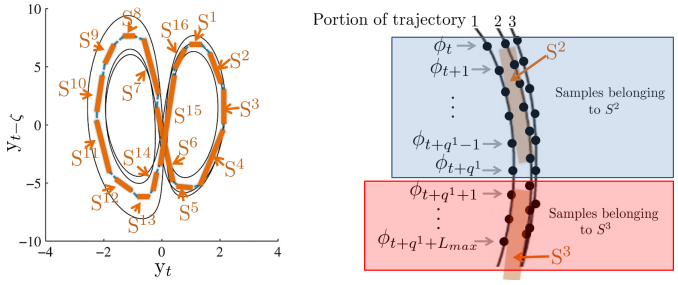
1) *Quantize the State Space Representation* (Fig. 2a): Embed the time series in a state space using delay coordinate embedding. Then use Segment-GNG to quantize the space state representation. Let S be the set of all segments obtained by Segment-GNG.
2) *Create local prediction models* (Fig. 2b): The objective is to predict the sample y_{t+1} using local prediction models built in the state space of time series. A local prediction model is created for each segment $S^i \in \mathsf{S}$. Let's assume that the segment S^i is the BMLS of k portions of trajectories, where the *k-th* portion has q^k samples. Next, we add L_{max} future samples to each portion of trajectory. This is done to ensure a correct prediction through two temporally connected segments. To estimate $\hat{y}_{t+1}$, the local prediction models take as input the sample $\phi_t \in S^i$ and its L_{max} past samples as follows:

$$\hat{y}_{t+1} = f_{S^i}\left(\{\phi\}_t^{t-L_{max}}\right), \tag{7}$$

where the function $f_{S^i}(\bullet)$ is the prediction model. To estimate the function $f_{S^i}(\bullet)$, let z_t be the vector with all components of $\{\phi\}_t^{t-L_{max}}$, i.e., $z_t = [y_t, \cdots, y_{t-L_{max}}, y_{t-\zeta}, \cdots, y_{t-\zeta-L_{max}}, \cdots, y_{t-(m-1)\zeta}, \cdots, y_{t-(m-1)\zeta-L_{max}}]$. Then we can approximate Eq. 7 as:

$$\hat{y}_{t+1} = \alpha_0 + \sum_{l_1=1}^{mL} \alpha_{l_1} z_t(l_1) + \sum_{l_1 \leq l_2}^{mL} \alpha_{l_1,l_2} z_t(l_1) z_t(l_2), \tag{8}$$

where $mL = m \times L_{max}$, $z_t(l_1)$ is the l_1-*th* component of z_t and α_0, α_{l_1} and α_{l_1,l_2} are the constant, linear and quadratic coefficient of $f_{S^i}(\bullet)$ which are estimated using the $\sum_k \left(q^k + L_{max}\right)$ samples associated with S^i. Since the function $f_{S^i}(\bullet)$ is linear in its parameters but quadratic in its terms, the α coefficients are estimated by least squares or LMS algorithm.



(a) (b)

Fig. 2: Example to illustrate the training phase. (a) Quantization of the state space representation of the times series using Segment-GNG. The set of all segments is $\mathsf{S} = \{S^1, S^2, \cdots, S^{16}\}$ (b) Zoom of S^2 and the $k = 3$ portions of trajectories associated with this segment. Each *k-th* portion of trajectory has q^k samples plus L_{max} futures samples. The $\sum_k \left(q^k + L_{max}\right)$ samples are used to train the $f_{S^2}(\bullet)$ according to Eq. 8.

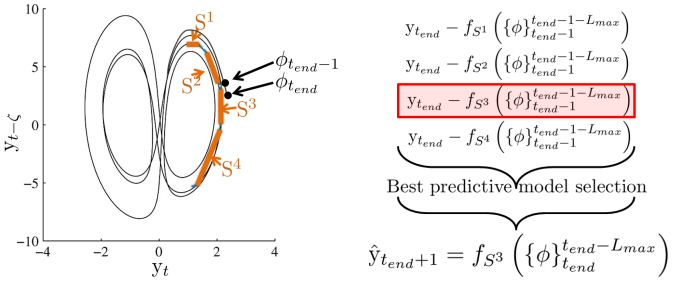
PREDICTION PHASE

To estimate the next point $\hat{y}_{t_{end}+1}$, we get $\phi_{t_{end}}$ and its past L_{max} samples from the state space representation of time series and we perform the following steps:

1) *Nearest segments*: Find the p nearest segments $S^i \in \mathsf{S}$ of the current sample $\phi_{t_{end}}$ using Eq. 6, and store them in N.
2) *Temporal neighbor segments* (Fig. 3a): Find those segments that temporally connected to each $S^i \in \mathsf{N}$. Add these new segments to N.
3) *Best prediction model selection*: Each segment $S^i \in \mathsf{N}$ has associated a prediction model $f_{S^i}(\bullet)$. In order to select the best prediction model from N, we predict $\hat{y}_{t_{end}}$ instead of $\hat{y}_{t_{end}+1}$, since at time t the sample $y_{t_{end}}$ is known. Then we evaluate all models associated with segments $S^i \in \mathsf{N}$ on the portion of trajectory $\{\phi\}_{t_{end}-1}^{t_{end}-1-L_{max}}$. We select the index of the best model $idx_{opt}(\phi_{t_{end}-1})$ as follows:

$$idx_{opt}(\phi_{t_{end}-1}) = \arg\min_{i \in \mathsf{N}} \left\| y_{t_{end}} - \hat{y}^i_{t_{end}} \right\|, \quad (9)$$

where $\hat{y}^i_{t_{end}} = f_{S^i}\left(\{\phi\}_{t_{end}-1}^{t_{end}-1-L_{max}}\right)$.
4) *Prediction* (Fig. 3b): Use the best prediction model $f_{S^{idx_{opt}}}(\bullet)$ to predict $\hat{y}_{t_{end}+1}$ from the current sample $\phi_{t_{end}}$, i.e., $\hat{y}_{t_{end}+1} = f_{S^{idx_{opt}}}(\{\phi\}_{t_{end}}^{t_{end}-L_{max}})$.

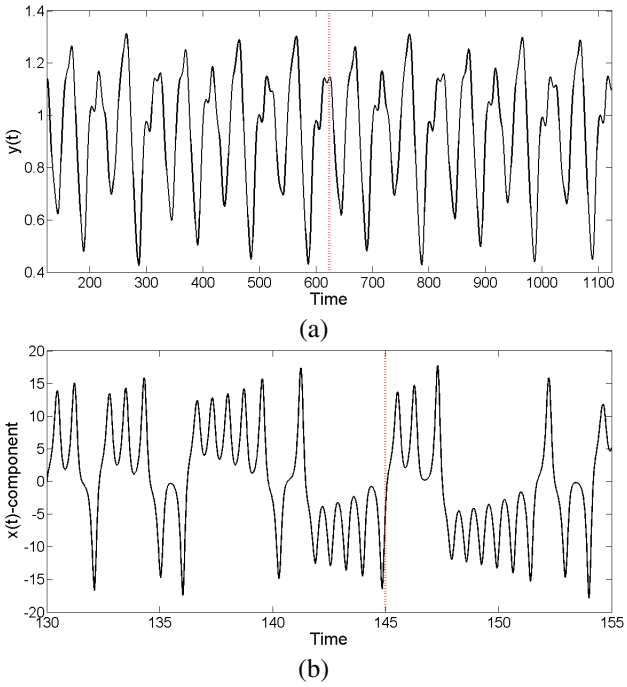


(a) (b)

Fig. 3: Example to illustrate the prediction phase. The objective is to predict y_{tend+1} using the information of $\phi_{t_{end}}$ and its L_{max} past samples. (a) Assuming that $p = 2$, the two nearest segments of $\phi_{t_{end}}$ (using Eq. 6) are S^2 and S^3 ($\mathsf{N} = \{S^2, S^3\}$). The segments that are temporally connected to S^2 and S^3 are $\{S^1, S^3\}$ and $\{S^2, S^4\}$ respectively. Then the updated set is $\mathsf{N} = \{S^1, S^2, S^3, S^4\}$. (b) Best prediction model selection and prediction. Evaluation of $f_{S^1}(\bullet)$, $f_{S^2}(\bullet)$, $f_{S^3}(\bullet)$ and $f_{S^4}(\bullet)$ on the portion of trajectory $\{\phi\}_{t_{end}-1}^{t_{end}-1-L_{max}}$ to estimate $\hat{y}_{t_{end}}$. The best model is chosen as the one giving the minimum error prediction of $y_{t_{end}}$, in this case $f_{S^3}(\bullet)$ (showed in the red box). Then $y_{t_{end}+1}$ is predicted using model $f_{S^3}(\bullet)$.

IV. EXPERIMENTS

In order to evaluate the proposed prediction strategy based on Segment-GNG in one-step ahead prediction, we designed



(a) (b)

Fig. 4: Time series used in the experiments. (a) The Mackey-Glass time series and (b) the Lorenz time series. The red line indicate the limit between training dataset (to the left) and testing dataset (to the right).

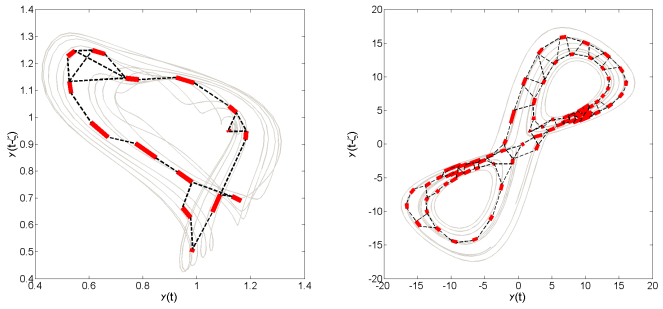
two experiments using two traditional time series: Mackey-Glass [34] and Lorenz [35]. The first experiment use Mackey-Glass time series and aims to show the advantages of Segment-GNG and give some insight into how the proposed method works. The second experiment use Lorenz time series and allows us to apply the guidelines to set the parameters of our method and to evaluate its performance. For comparison purposes we use the results of earlier works [36]–[39] that have used the same time series under similar conditions (data, embedding space, parameters, etc.).

A. Parameter setting

The most sensitive parameters in our method are the number of delays L_{max} to construct the local prediction models and the number of segments $n_{segment}$ to quantize the state space. To find the best model for each dataset, we tried all the combinations of varying L_{max} from 1 to 10 and $n_{segment}$ as defined for each dataset in what follows. The rest of parameters in Segment-GNG are set using the default parameters in [26]. The last 50 points in the training set were used to select the best parameter combination (best prediction model) through five-fold cross validation. The testing set was only evaluated in the best selected model. We calculate the NMSE and RMSE to compare our proposed method with other methods in the literature.

B. Mackey-Glass Time Series

Mackey-Glass is a chaotic time series commonly used in the literature as a benchmark model [34]. Mackey-Glass time



(a) (b)

Fig. 5: Results obtained with Segment-GNG using the best combination of parameters for each dataset. The thin lines correspond to the state-space representation, the thick lines represent the quantization segments and the dotted lines illustrate the temporal connections between segments. (a) Quantization of the state space for Mackey-Glass time series ($n_{segment} = 15$, $L_{max} = 7$). (b) Quantization of the state space for the Lorenz time series ($n_{segment} = 75$, $L_{max} = 5$).

series is generated by the following equation:

$$\frac{dy_t}{dt} = \frac{0.2y_{t-17}}{1 + y_{t-17}^{10}} - 0.1y_t. \tag{10}$$

In this experiment, we generated 1000 samples from $t = 124$ to $t = 1123$. The first 500 samples were used for training and the last 500 samples were used for testing. Figure 4a shows the Mackey-Glass time series. The samples were generated using a 4th order Runge–Kutta method with initial condition $y_0 = 1.2$. The goal of this experiment is to make a one-step ahead prediction. To generate the embedding state space we use $\zeta = 17$ and $m = 1$. The number of Segment-GNG quantization units is $n_{segment} = \{10, 15, 25, 40, 65, 105\}$.

Table I shows the results obtained with the top 5 parameter combinations ($\{n_{segment}, L_{max}\}$) for the Mackey-Glass time series. The first 4 combinations have in common that the number of segments is relatively low and the NMSE in validation is under 1.3×10^{-8}. However augmenting the number of segments to $n_{segment} = 40$ and $L_{max} = 6$ increases the validation error rapidly to 5.371×10^{-8}. Figure 5a illustrates the distribution of segments obtained by Segment-GNG quantization when using the best combination of parameters ($n_{segment} = 15$, $L_{max} = 7$). This distribution ensures that the number of portions of trajectories quantized by each segment are enough to train the local prediction models.

An interesting aspect that can be observed in Fig. 5a is that the temporal connections faithfully capture the sense of movement of trajectories. Another interesting aspect is that only a small number of segments (fifteen) is needed to make a good prediction. This is because the Mackey-Glass time series have a relatively smooth dynamics and most of the trajectories can be approximated with linear segments.

Figure 6 shows the absolute error when using the best parameter combination for the testing set. Table II shows a comparison of results between the proposed model and other models published in the literature using NMSE and RMSE metrics. According to this table our method outperforms the

TABLE I: The top five parameter combinations in the one-step ahead prediction for the Mackey-Glass time series. The ranking is made according to the validation error.

Parameter		Validation	Test	
$n_{segment}$	L_{max}	NMSE $\times 10^{-8}$	NMSE $\times 10^{-8}$	RMSE$\times 10^{-5}$
15	7	0.103	0.23	1.09
25	7	0.117	0.67	1.84
15	10	0.954	1.93	3.14
15	5	1.251	0.59	1.73
40	6	5.371	1.45	2.72

TABLE II: Performance comparison between the results obtained with our method and those of other models published in the literature for Mackey-Glass time series.

Model	NMSE $\times 10^{-8}$	RMSE $\times 10^{-5}$
Proposed method	0.23	1.09
RNN [40]	23500	–
SOM+RNN (4 units) [36]	5120	–
SOM+RNN (40 units) [36]	3600	–
ERNN (2007) [37]	3.15	4.20
HE-NARXNN (2010) [38]	2.70	3.72
KLPP (2015) [39]	1.04	2.46

combinations of SOM with recurrent NN. This is because the Segment-GNG quantization captures nearby spatial trajectories with similar directions, thus preventing local prediction models to learn information about trajectories with different directions.

Segments versus centroids: To evaluate the advantage of the space-directional quantization produced by Segment-GNG, we compare it with the traditional K-means quantization algorithm [13]. The aim of this comparison is to evaluate the prediction accuracy when using the quantization made by Segment-GNG and K-means. The local prediction models are trained in the same way as described above. The only difference is the quantization method used to quantize the state space. The prediction task is one-step-ahead and the evaluation of the models is done with NMSE in the validation dataset.

As it was shown in Table I, the best combination of parameters for Segment-GNG prediction on Mackey-Glass is $n_{segment} = 15$ and $L_{max} = 7$. For K-means we use the same L_{max} and vary the number of quantization units (centroids) as $n_{centroids} = \{10, 15, 25, 40, 65, 105\}$.

Table III shows the performance of K-means on the

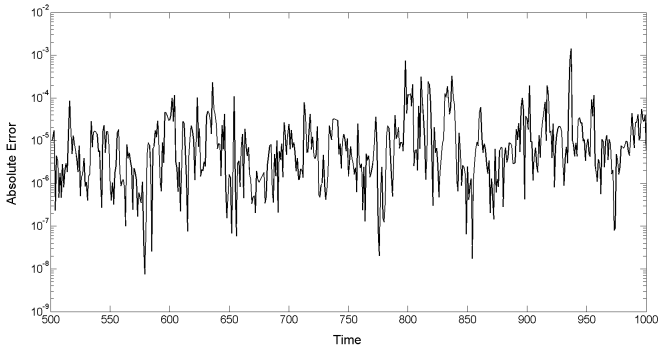
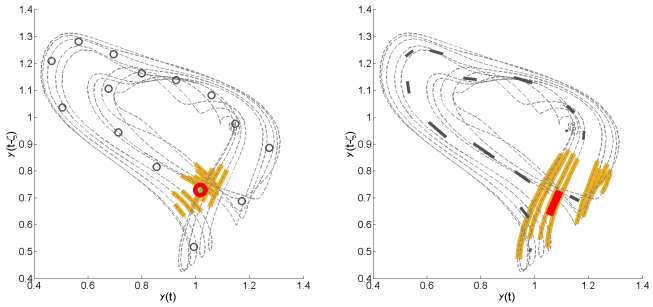


Fig. 6: Prediction absolute test error for the Mackey-Glass time series. The NMSE $= 0.23 \times 10^{-8}$ and RMSE $= 1.09 \times 10^{-5}$.



(a) (b)

Fig. 7: Portions of trajectories (orange lines) belonging to a quantization unit (red) on (a) K-means, and (b) Segment-GNG. The thin lines correspond to the state-space representation and the black marker shows the quantization unit in each method. Note that K-means associates all kind of trajectories (even those with perpendicular directions) with a given unit, whereas Segment-GNG associates only parallel trajectories with a given segment.

TABLE III: K-means performance on Mackey-Glass time series with different numbers of centroids and $L_{max} = 7$.

$n_{centroids}$	10	15	25	40	65	105
NMSE $\times 10^{-8}$	480.97	249.87	115.97	60.52	650.55	333.06
RMSE$\times 10^{-5}$	49.49	35.67	24.30	17.56	57.56	41.18

Mackey-Glass time series with different number of centroids. Regardless of the number of centroids, the performance of K-means is worse than the results obtained with our method. Figures 7a and 7b shows the quantization obtained with K-means and Segment-GNG respectively using 15 units. Figure 7a shows that K-means does not distinguish among trajectories with perpendicular directions, unlike Segment-GNG (Figure 7b). As the prediction strategy is the same in both methods, the results highlight the usefulness of Segment-GNG space-directional quantization.

Although it could be possible to improve K-means results by using more sophisticated local prediction models, the advantage of Segment-GNG is that we can use simpler local prediction models, ensuring a rapid convergence in training and reaching a better understanding of the behavior of local models.

C. Lorenz time series

The Lorenz system was derived from a model of the Earth's atmospheric convection flow heated from below and cooled from above [35]. The Lorenz system is described using nonlinear differential equations as follow:

$$\frac{dx}{dt} = \sigma(y - x) \tag{11}$$

$$\frac{dy}{dt} = (r - z) - y \tag{12}$$

$$\frac{dz}{dt} = xy - bz, \tag{13}$$

where $\sigma = 10$, $b = 8/3$ and $r = 28$. A 4th-order Runge-Kutta method is applied to generate the time series with initial conditions: $x(0) = 1$ and $y(0) = z(0) = 0$. The x-coordinate of the Lorenz time series is considered in this experiment. 2501 samples are generated from $t = 130$ to $t = 155$ with a sampling time of 0.01. The first 1501 samples are used for training and the last 1000 samples for testing. The objective in this experiment is to make a one-step ahead prediction. To generate the embedding space we use $\zeta = 0.012$ and $m = 1$. The number of Segment-GNG quantization units is $n_{segment} = \{50, 75, 125, 200, 325, 525\}$.

TABLE IV: The top five parameter combinations in the one-step ahead prediction for the Lorenz time series. The ranking is made according to the validation error.

Parameter		Validation	Test	
$n_{segment}$	L_{max}	NMSE $\times 10^{-8}$	NMSE $\times 10^{-8}$	RMSE$\times 10^{-5}$
75	5	2.25×10^{-6}	6.90×10^{-3}	0.63
50	3	5.12×10^{-5}	1.38×10^{-4}	0.09
75	3	7.46×10^{-4}	4.58×10^{-3}	0.52
125	5	1.26×10^{-4}	3.99×10^{-2}	1.53
125	4	2.16×10^{-3}	3.02×10^{-5}	1.33

Table IV shows the prediction performance of Segment-GNG in the Lorenz dataset for different combinations of L_{max} and $n_{segment}$. The best combination corresponds to $n_{segment} = 75$ and $L_{max} = 5$. The small number of segments needed to quantize the state space representation is because the trajectories are relatively smooth. Thus each segment captures a great number of portions of trajectories. Similar to the Mackey-Glass case, the good performance of our method with low number of segments is explained because a sufficient number of trajectories is captured for each segment. When increasing the number of segments, the number of trajectories captured by a segment decreases and the prediction power of local models is lost.

Considering the traditional analogue of the Lorenz attractor with a butterfly, in each wing of the attractor in Figure 5b, we can observe that there are areas densely populated by segments. In these areas, the trajectories are divided into those that are maintained in the same wing and those that jump in to the other wing. These areas need more directional information with respect to other areas and therefore more segments.

Table V shows a performance comparison between our method and other models in one-step ahead prediction task for the Lorenz time series. The best results are obtained by our method. Figure 8 shows the absolute error on the testing set when using the best model obtained. It is interesting to note that while we use local prediction models with quadratic terms, other methods use recurrent neural networks or Gaussian kernels to obtain a comparable performance. The good performance of Segment-GNG quantization shows the importance of the space-directional information and temporal quantization to generate a reliable local prediction model.

V. Conclusions

In this work we have proposed a new method for times series prediction using Segment-GNG. The main novelty of this work is to take advantage of the temporal and space-directional information captured by Segment-GNG to create simple local prediction models. A segment, unlike nodes or

TABLE V: Performance comparison between the results obtained with our method and those of other models published in the literature for Lorenz time series.

Model	NMSE $\times 10^{-8}$	RMSE $\times 10^{-5}$
Proposed method	6.90×10^{-3}	0.63
ERNN (2007) [37]	9.90×10^{-2}	0.88
HE-NARXNN (2010) [38]	1.98×10^{-2}	10.8
KLPP (2015) [39]	1.04	2.03

prototypes used in other quantization methods (SOM, K-means, etc), quantizes trajectories with similar directions on the embedding space, allowing us to create simpler prediction models associated with each segment. The latter was confirmed when using Segment-GNG versus K-means quantization on the Mackey-Glass time series. A segment's domain includes trajectories with regular behavior and directions in contrast with a centroid's domain which includes trajectories in any direction. In addition when using the same kind of local prediction models, Segment-GNG obtained a higher prediction performance than K-means. Furthermore, the temporal connection information among segments is relevant to know how the different prediction model (segment) interact and create more accurate local prediction models. In some cases a predicted sample maybe out of the current segment's domain, so the next segment would be necessary to predict.

The prediction results obtained for Mackey-Glass and Lorenz time series using our method show that Segment-GNG is very competitive with the state of the art methods. In addition our local prediction models are less complex than those used by SOM-RNN, KLPP and ERNN.

The future work to improve the proposed method is summarized as follows:

(i) *On-line creation of local models*: The current method generates the local prediction models after the quantization made by Segment-GNG. The limitation of this strategy is that the local predictors are fitted to a previous quantization. Actual time series can be more dynamic in time and change their behavior constantly, therefore it is convenient to produce prediction models that are more flexible to quick changes.

(ii) *Multiple step-ahead prediction*: This work focused on one-step ahead prediction, however, in the near future we plan to extend the horizon of prediction in order to make multiple-steps prediction, and iterated predictions.

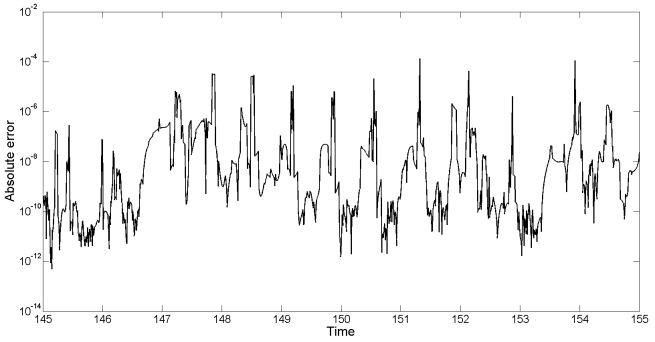


Fig. 8: Prediction absolute test error for Lorenz time series on the one-step ahead prediction task. The NMSE $= 6.90 \times 10^{-11}$ and RMSE $= 0.63 \times 10^{-5}$

(iii) *Combination of temporally connected segments*: In our method, the prediction is made using either the nearest local predictor model or the following one temporally connected. When predicting multiple-steps ahead the current local model and next temporal model would not be enough. It would be necessary to use local prediction models that are more distant temporally.

(iv) Build local prediction models both in the state-space and output space, and use the error information (covariance) as done by the Kalman Filter.

ACKNOWLEDGMENT

This research was supported by Conicyt-Chile under grants Fondecyt Postdoctorate 3160772 and Fondecyt Regular 1171678.

REFERENCES

[1] H. Kantz and T. Schreiber, *Nonlinear time series analysis*. Cambridge university press, 2004, vol. 7.

[2] L. Cao, Y. Hong, H. Fang, and G. He, "Predicting chaotic time series with wavelet networks," *Physica D: Nonlinear Phenomena*, vol. 85, no. 1, pp. 225–238, 1995.

[3] U. Thissen, R. Van Brakel, A. De Weijer, W. Melssen, and L. Buydens, "Using support vector machines for time series prediction," *Chemometrics and intelligent laboratory systems*, vol. 69, no. 1, pp. 35–49, 2003.

[4] L. Y. Su, "Prediction of multivariate chaotic time series with local polynomial fitting," *Computers & Mathematics with Applications*, vol. 59, no. 2, pp. 737–744, 2010.

[5] R. J. Frank, N. Davey, and S. P. Hunt, "Time series prediction and neural networks," *Journal of Intelligent & Robotic Systems*, vol. 31, no. 1, pp. 91–103, 2001.

[6] P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*. springer, 2016.

[7] J. Walter, H. Riter, and K. Schulten, "Nonlinear prediction with self-organizing maps," in *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*. IEEE, 1990, pp. 589–594.

[8] G. Simon, J. A. Lee, M. Cottrell, and M. Verleysen, "Forecasting the cats benchmark with the double vector quantization method," *Neurocomputing*, vol. 70, no. 13, pp. 2400–2409, 2007.

[9] J. Vesanto, "Using the som and local models in time-series prediction," in *Proc. Workshop on Self-Organizing Maps 1997*. Citeseer, 1997, pp. 209–214.

[10] J. C. Principe, L. Wang, and M. A. Motter, "Local dynamic modeling with self-organizing maps and applications to nonlinear system identification and control," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2240–2258, 1998.

[11] R. Gray, "Vector quantization," *IEEE Assp Magazine*, vol. 1, no. 2, pp. 4–29, 1984.

[12] T. Kohonen, *Self-organizing maps*. Springer, Heidelberg, 1995.

[13] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[14] T. Martinetz, S. Berkovich, and K. Schulten, "'Neural-gas' network for vector quantization and its application to time-series prediction," *Neural Networks, IEEE Transactions on*, vol. 4, no. 4, pp. 558–569, 1993.

[15] B. Fritzke, "A growing neural gas network learns topologies," *Advances in neural information processing systems*, vol. 7, pp. 625–632, 1995.

[16] N. R. Euliano and J. C. Principe, "A spatio-temporal memory based on soms with activity diffusion," in *Kohonen Map*, E. Oja and S. Kaski, Eds. Elsevier, Amsterdam, 1999, pp. 253–266.

[17] M. Strickert and B. Hammer, "Merge som for temporal data," *Neurocomputing*, vol. 64, pp. 39 – 71, 2005.

[18] P. A. Estévez and R. Hernández, "Gamma som for temporal sequence processing," in *Advances in SOM's*. Springer, 2009, pp. 63–71.

[19] A. Andreakis, N. Hoyningen-Huene, and M. Beetz, "Incremental unsupervised time series analysis using merge growing neural gas," in *Advances in Self-Organizing Maps*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5629, pp. 10–18.

[20] P. Estévez and J. Vergara, "Nonlinear time series analysis by using gamma growing neural gas," in *Advances in Self-Organizing Maps*, ser. Advances in Intelligent Systems and Computing, P. A. Estévez, J. C. Príncipe, and P. Zegers, Eds. Springer Berlin Heidelberg, 2013, vol. 198, pp. 205–214.

[21] M. Varsta, J. Heikkonen, J. Lampinen, and J. D. R. Millán, "Temporal kohonen map and the recurrent self-organizing map: Analytical and experimental comparison," *Neural processing letters*, vol. 13, no. 3, pp. 237–251, 2001.

[22] T. Voegtlin, "Recursive self-organizing maps," *Neural Networks*, vol. 15, no. 8, pp. 979–991, 2002.

[23] D. L. James and R. Miikkulainen, "Sardnet: A self-organizing feature map for sequences," *Advances in neural information processing systems*, vol. 7, p. 577, 1995.

[24] F. Mizushima and T. Toyoshima, "Language learnability by feedback self-organizing maps," in *International Conference on Neural Information Processing*. Springer, 2006, pp. 228–236.

[25] J. Principe, N. Euliano, and S. Garani, "Principles and networks for self-organization in space-time," *Neural Networks*, vol. 15, no. 8-9, pp. 1069 – 1083, 2002.

[26] J. R. Vergara, P. A. Estévez, and Á. Serrano, "Segment growing neural gas for nonlinear time series analysis," in *Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of the 11th International Workshop WSOM 2016, Houston, Texas, USA, January 6-8, 2016*, E. Merényi, M. J. Mendenhall, and P. O'Driscoll, Eds. Springer International Publishing, 2016, pp. 107–117.

[27] G. A. Barreto, "Time series prediction with the self-organizing map: A review," in *Perspectives of neural-symbolic integration*. Springer, 2007, pp. 135–158.

[28] F. Takens, *Detecting strange attractors in turbulence*, ser. Lecture Notes in Math. New York: Springer, 1981, vol. 898.

[29] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical review A*, vol. 33, no. 2, p. 1134, 1986.

[30] M. B. Kennel, R. Brown, and H. D. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical review A*, vol. 45, no. 6, p. 3403, 1992.

[31] S.-K. Oh, K.-H. Seo, and J.-J. Lee, "Time series prediction by mixture of linear local models," in *Industrial Electronics Society, 2003. IECON'03. The 29th Annual Conference of the IEEE*, vol. 2. IEEE, 2003, pp. 1905–1908.

[32] J. S. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons," *International journal of forecasting*, vol. 8, no. 1, pp. 69–80, 1992.

[33] R. J. Hyndman *et al.*, "Another look at forecast-accuracy metrics for intermittent demand," *Foresight: The International Journal of Applied Forecasting*, vol. 4, no. 4, pp. 43–46, 2006.

[34] M. C. Mackey, L. Glass *et al.*, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[35] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[36] A. Cherif, H. Cardot, and R. Boné, "Som time series clustering and prediction with recurrent neural networks," *Neurocomputing*, vol. 74, no. 11, pp. 1936–1944, 2011.

[37] Q.-L. Ma, Q.-L. Zheng, H. Peng, T.-W. Zhong, and L.-Q. Xu, "Chaotic time series prediction based on evolving recurrent neural networks," in *Machine Learning and Cybernetics, 2007 International Conference on*, vol. 6. IEEE, 2007, pp. 3496–3500.

[38] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid elman–narx neural networks," *Neurocomputing*, vol. 73, no. 13, pp. 2540–2553, 2010.

[39] L. Su and C. Li, "Local prediction of chaotic time series based on polynomial coefficient autoregressive model," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[40] R. Boné, "Recurrent neural networks for time series forecasting," Ph.D. dissertation, Université de Tours, France, 2000.