New Learning Technique Applied to Time Series Forecasting

K. Ramírez-Amaro and J. C. Chimal-Eguía
Centro de Investigación en Computación
Instituto Politécnico Nacional
Av. Juan de Dios Bátiz s/n Unidad Profesional Adolfo López Mateos
Col. Nueva Industrial Vallejo, Mexico City, 07738, Mexico.
kramireza@ipn.mx, chimal@cic.ipn.mx

N. Sánchez-Salas
Departamento de Física, Escuela Superior de Física y Matemáticas del IPN
UP Zacatenco, Mexico City, 07738, Mexico
norma@esfm.ipn.mx

Abstract

Forecasting is an important activity in physics, economics, commerce, marketing and various branches of science. This article is concerned with a new forecasting method based on the information obtained from the image axes of a time series. A time series is a collection of observations made sequentially through time, for instance, the temperature at a particular location. In order to obtain an interesting result in time series forecasting we implement a new input representation of the data and also we generate a new learning technique which through probabilistic mechanism this learning could be applied to the interesting forecasting problem. The result indicate that using the methodology proposed in this article it is possible to obtain forecasting results with good enough accuracy.

1. Introduction

The time series (TS) are principal used when the phenomena are not calculated or measure by mathematical models but relying on observation or experiment. In other words, a time series is a sequence of values over the time of a system $x(t)$ which registers a sequence of experimental values [3][2]:

$$x(t_1), x(t_2), x(t_3), ..., x(t_n) \quad (1)$$

for some interval $t = n$ with $t_0 < t_1 < ... < t_n$ [1]. Many natural phenomena can be represented as time series such as temperature, electronical signals, economical data, social data among others.

In fact, for time series forecasting there exist many nonlinear techniques such as Artificial Neural Networks (ANNs) [6], Support Vector Machine (SVM) [7] among others. Those learning techniques has some incovenients, for example in ANNs we need to decide the optimal architecture and the optimal function to train the network [5]. In the other hand, in SVM it is necesary to decide the type of kernel function either linear or nonlinear [3]. It is important to notice that an incorrect choice of either in the input data representation or in the learning process should affect the final result. A complete review about this nonlinear techniques can be found at [4].

In this article, we present a new methodology to obtain the input data using the image information [10] and a new learning technique based on statistical procedures is also implemented. These methods are implemented to perform time series forecasting from the information obtained from the image axes of the time series.

2. Input data representation

This new input data representation is based on the principle coined "subgoals" [8] as follows: we divide the image of time series into small sections and this new divisions are called "boxes". That is because, is easier to learn the behaviour of small amplitude intervals of time series than trying to learn the behaviour of the whole time series. In order to do that we implement the definition of intervals.

An interval is a set that contains every real number between two defined numbers and may contain the two numbers themselves. i.e., if $y_1, y_2 \in \Re$ then the interval $[y_1, y_2]$ is defined as the following subset of $\Re$:

$$[y_1, y_2] = \{y \mid y_1 \leq y \leq y_2\} \quad (2)$$

where y_1 is the lower bound of the interval and y_2 is the upper bound of an interval $[y_1, y_2]$. In order to enclose points between the two dimensions (i.e. the x and y axis) it is necesary to use the next interval real vector definition.

An interval real vector ($[Box]$) is a subset of $\Re^n$ that can be defined as the cartesian product of n intervals $[y_1, y_2]$:

$$[Box] = [x_1, x_2] \times [y_{1i}, y_{2i}] \, \forall i = 1, 2, 3, ..., n \quad (3)$$

Then it is possible to make the next definition:

Definition 1: Given the following function $area_f = [Box] \rightarrow \mathbb{N} \cup \{0\} \cup \{\infty\}$ it is possible to define it as:

$$area_f \, [Box] = card \left([y_1, y_2] \cap \{f(x) \mid x \in [x_1, x_2]\} \right)$$

where $card\,(*)$ means the cardinality of the set $(*)$. This area function indicates the nunmber of points contained in one box (see Fig.)

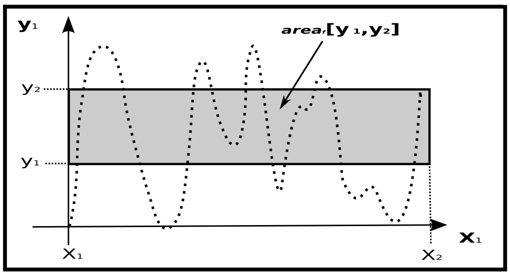

Figure 1. The box is defined by the upper and lower image limits y_1 and y_2, and the upper and lower time limits x_1 and x_2 respectivelly. It is also observe that the $area_f$ within this box is equal to 86 points.

3 Learning process

Firstly, it is necessary to define the term of learning we used in this article as follows [11]:

Definition 2: Learning is the process of a*cquire and acumulate knowledge about some task through experience, which allows to execute the same task better in the future than the last time.*

In order to learn the behaviour of a TS, firstly we divide the image of the TS into n boxes, for this purpose we generated a new mechanism as follows:

- Firstly, it is necessary to identify the corresponding box belonging to the original point of the TS i.e., $x(i)$, following the next condition: $if \; x(i) > y_j \; and \; x(i) \leq y_{j+1} \; then \; x(i) \in Box(j)$. For instance, see the Fig. 2a) where $x(6) = 0.68$ and 0.68>0.6 and 0.68≤0.8 $then \; x(6) \in Box(4)$.
- Then, we must determine the tendency of the points according to the following conditions:
  - $if \; x(i) < x(i-1) \; then \; band = 0$ (i.e., the points are decreasing).
  - $if \; x(i) > x(i-1) \; then \; band = 1$ (i.e., the points are rising).

  For example, if we observe again the Fig. 2a) the point $x(6) > x(5)$ this means that the point $x(6)$ is rising therefore it band is equal to 1.
- After, the point that track the dynamical behaviour of the observed point is calculated by a bounded random between the lower and upper bound of the image box, i.e., $trackPoint\,(i) = \mathrm{rand}(y_j, y_{j+1}]$ we can observe the Fig. 2b) where $x(6) \in Box\,(4)$ and the image interval of that box is equal to $(0.6, 0.8]$ then the track point is calculated by randomly between the image interval of the Box(4) which supossed is equal to 0.681. Then, in order to verify if this random point is closed to the observe point it is necesary to calculate the error between the original and calculated point.
- Finally, it is necessary to save the calculated points as well as its errors in order to learn from its experience. This information is saved into two matrices called Ua and Ue, the first one stores the tracking points and the second one stores its corresponding error. As we can observe from the Fig. 2c) there are two matrices where the rows correspond to the number of sequence points into a specific box (the columns). For example, the tracking point equals to 0.681 is stored in $Ua(1, 4)$ and its error in $Ue(1, 4)$.

4 Time Series Forecasting

One of the goals of forecasting techniques is to extract the biggest amount of information from the behavior of time series during the learning process and from this information predict the behavior of next points in the future [3].

On the basis of the learning process, we generate the matrices Ua and Ue but those matrices are not the only extracted information from the learning process. During the implementation of the learning process we could notice that when we use the boxes representation we can define and extract the following additional information:

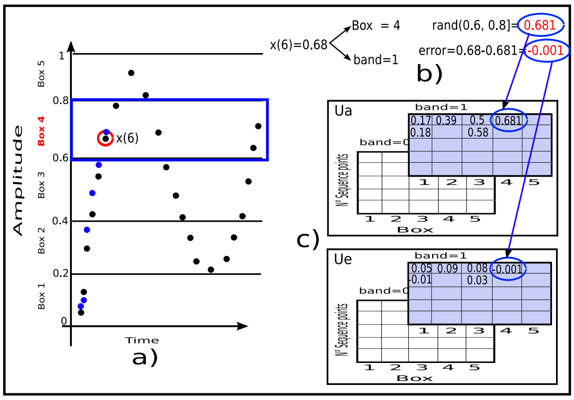

Figure 2. Example of the learning process. In a) is observed the identification of the observed point (black dots) to an specific box and its tendency. In b) is shown how to compute the tracking points (light dots) and its error. Finally, in c) is observed how to store the information from this learning process.

The **length** of points indicate the sequence of points that belong to an specific box and this following one of the next conditions:

- if the points are rising these should achieve that $trackPoint\,(1) < trackPoint\,(2) < ... < trackPoint\,(l)$
- if the points are decreasing these should achieve that $trackPoint\,(1) > trackPoint\,(2) > ... > trackPoint\,(l)$

If one set of points follows the above conditions then we say that these points have a length equals to l (see Fig. 3b).

The **Jumps** refers to a characteristic of some points that moves from one box to another box as long as these points do not change its tendency, i.e. the Jumps measure the leap that some points do when some point $x\,(i)$ is in one box and the next point $x\,(i+1)$ just jumps another box without to change its tendency (see Fig. 3c).

The information obtained from the above definitions is stored into two matrices called $longs$ and $jumps$ respectivelly (see Fig. 4a) and b)) and from these matrices it is possible to extract the following information:

- the length of some points during one box and in the same visit
- how many times some boxes are being visited
- the next box the points go after visiting some box

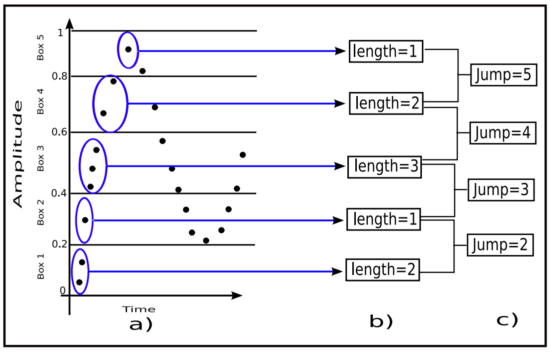

Figure 3. Ilustration of the definitions of length and jumps. In a) we observe that the firts points are rising therefore in b) is shown the length belonging to those points and in c) we observe the jumps the points do.

If we analize the matrices $longs$ and $jumps$ we can notice that if we merge these two matrices then it is posible to multiply the information. This new matrix is called $longJump$ defined as:

$$longJump := longJump_{i,j,k,l} \tag{4}$$

$\forall 1 \leq i \leq m, 1 \leq j \leq n, 1 \leq k \leq o, 1 \leq l \leq p$, where m is the length of some points during a visit, n is the visited box, i.e., the jump, o is the actual box and p is the sense of the points. The values that the matrix $longJump\,(i,j,k,l)$ contain are the frequency of points which has the length i, jump to the box j from the box k and those have the same tendency l. To illustrate the merge between the matrices $longs$ and $jumps$ observe the Fig. 4.

The matrix $longJump$ store the frequency of the length and jumps at the end of the learning process. For instance, $longJump\,(7,2,1,1) = 1$ means that the points have length equals to 7 these points jump to the box 2 when they where in box 1 when they have rising tendency and the frequency of that pattern is equal to one and that means that during the learning process only once that pattern is repeated.

When the learning process is finished, the last learned points determine the following information:

- the last tracking point
- the last box this point visited
- the tendency of that point and
- the its length.

In order to forecast the next points of the TS first is necessary to estimate the next box which this future points

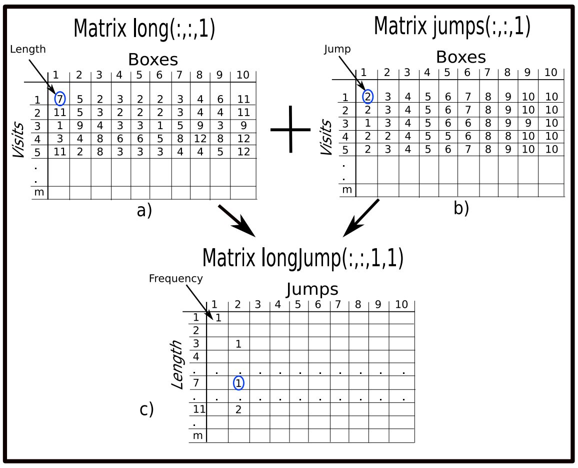

Figure 4. Example of the merge of the matrices $longs$ and $jumps$. In a) is observed the matrix $longs$, in b) is observed the $jumps$ matrix and finally in c) we can observe the merge of both matrices.

are going to jump and after that we must also estimate the length of this new points.

Firstly, in order to estimate the following box, we extract from the matrix $longJump$ the vector that follows the information of the last tracking point. For instance, observe the Fig. 5a) where the last point visited the box=2 the length of this point is equal to 4 and it has a decrearising tendency. Then using that information we extract the corresponding vector from the $longJump$ matrix, i.e., vector=$longJump\,(4,:,2,0)$ see Fig. 5b) as we can observe the columns of this new vector contain information of the box we want to estimate and inside this vector is the frequency of the points that jumped in the past to that box and as we observe from this vector extracted we realize that there are two future boxes to choose either the box 1 or the box 2. Therefore, in order to decide the next box we procede to compute a probabilistic table of each future box. For example, see Fig. 5b) and observe the rows of this table of the probabilities where row(1)= box(1) and row(2)=box(2) and from this probabilistic table it is posible to choose the next box by executing a random value between 0 and 1 and suposse that this value is equal to 0.123 and observe that this value correspond to box=1.

Secondly, it is necesary to determine the tendency of the next points when they jump to this new box as follows:

- it is important to notice that the tendency of this future points vary according to the nex box compute above, because if this next box is the same that the last visited box, then its tendency change.
- On the other hand, if the last tracking point has a rising tendency and the next estimated box is bigger than the last box visited then the tendency of the next points remains the same and viceversa. For instance, supposed that the last visited box is equal to 2 and those points has decreasing tendency and the next box is equal to 1 then the tendency of future points remains decreasing.
- When the last tracking points has a rising point and the next estimated box is smaller than the last visited box then the tendency of the future points change because they start to decrease and viceversa. For instante, supposed that the last visited box is equal to 3 and the points have rising tendency and also supposed that the next estimated box is equal to 2 then the tendency of the future points is now decreasing.

Once we estimate the next box it is necessary to estimate the length of the next points in the following terms:

1. From the matrix $longJump$ we extract a submatrix which has the information of the last tendency of the last tracking points and the next box calculated above.
2. The columns from the new matrix are summed and we get a new vector in which each row represent the next length of the future points.
3. As in association with the estimation of the next box the procedure used to estimate the next length of points is very alike. This means that we generate a table of probabilities about the frequent lenghts and then randomly choose one, also each lenght has its weight (frequency) (see Fig. 5c).

Finally, after the estimation of the next box, the next tendency and the next length it is necesary to determine the values of those next points by designing an indexing mechanism of the matrix computed during the learning process called Ua. This indexing process is as follows:

- Identify the right Ua matrix by the tendency estimated above then, from this Ua matrix is necesary to find the estimated box which correspond to a column of that Ua matrix. Once, the column of this matrix is identify then the estimated length indicate how many points is necesary to recover from the Ua matrix and this points are ploted as the forecasting points (see Fig. 5 d) y e)).
- Finally, it is necesary verify the performance of this forecasting approach between the original and the predicted point and the measure that we implemented is the root mean square error (RMSE):

$$\sqrt{\frac{\sum_{i=1}^{n}(x_o - x_p)^2}{n}} \tag{5}$$

where x_o means the original points from the time series, x_p means the forecasted points from the time series and n means the total number of the both time series [2]. This procedure is repeated until the forecasting points are equal to the total of points we want to predict.

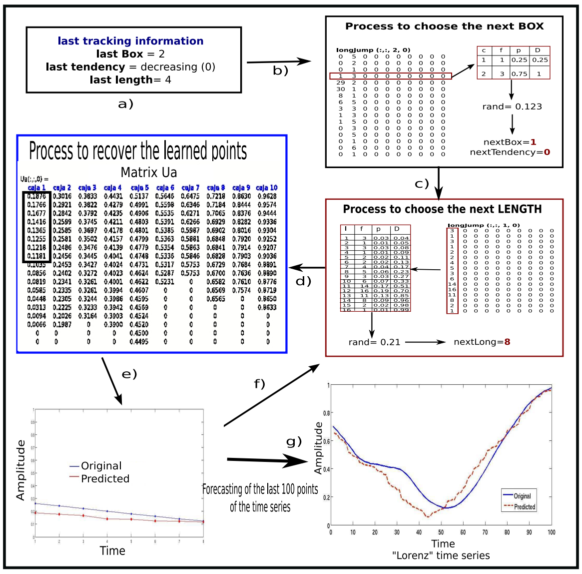

Figure 5. a) shown the last learning (tracking) information extracted from the TS, in b) is shown the procedure to estimate the next box, in c) is possible to observe the process to estimate the next length, in d) is shown the procedure to recovery the estimated points and finally, in e) this new points are ploted and this process is repeated (g) until the number of estimated points are equal to the number of point we want to estimate (f).

5 Experimental Results

In this article we presented a new approach to obtain new information from the image of the time series and this new information could be applied to the problem of forecasting.

In order to verify the performance of this new forecasting approach we prove this technique with time series from different behaviours such as: periodical, quasiperidodic, chaotic and stochastic systems [3]. To all of them we predict the next 100 points and we calculate the RMSE. We are going to show some of the obtained results[1].

Firstly, in Figure 6 the time series ploted shows a quasiperiodic behavior which means that the time evolution could be decompose into different sections which appear to be periodic themself but at different times. The prediction as we can see is very accuracy with an error $RMSE = 0.0012$

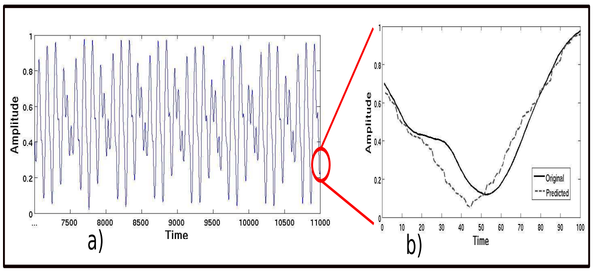

Figure 6. Results for a quasiperiodic time series. In a) is observed the plot of the whole time series and in b) is shown the 100 points predicted.

Then, in Figure 7 it is shown the lorenz time series which has a chaotic behavior that means that it describes the behavior of certain nonlinear dynamical systems that under certain conditions exhibit dynamics that are sensitive to initial conditions (popularly referred to as the butterfly effect). As a result of this sensitivity, the behavior of chaotic systems appears to be random, because of an exponential growth of errors in the initial conditions. Despite of its behavior the results indicate that this new forecasting approach it is possible to be predicted and the RMSE obtained is equal to 0.0082.

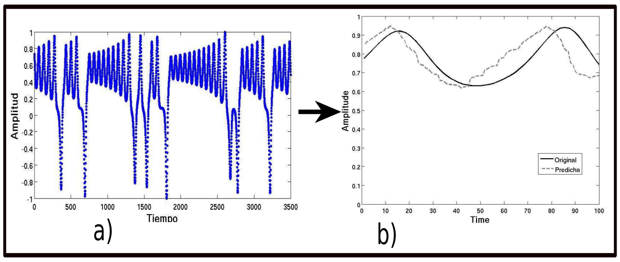

Figure 7. Results from the lorenz time series. In a) we observe the whole time series and in b) we observe the 100 predicted points.

Finally, in Figure 8 we observe the results obtained from the White Noise time series wich has a stochastic behav-

[1]The software used in our simulations was Matlab 7.0®and the specification of the hardware we used is a CPU Pentium 4 with 3.0 GHz and 512MB of RAM.

ior. A stochastic process is one whose behavior is non-deterministic in that a state does not fully determine its next state. Therefore, in this figure we observe that the forecasting is accuracy due its behavior with a RMSE equal to 0.1554.

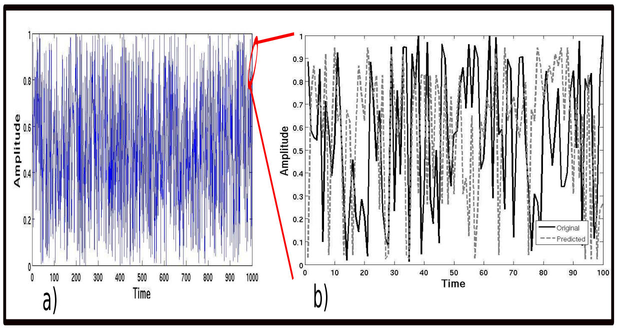

Figure 8. Results of the White Noise time series. In a) we observe the whole time series and in b) we observe the predicted points.

All the result indicates that it is possible to predict some points from the time series from different behaviors obtaining in all the cases a low error between the original and the predicted points from the time series with the RMSE.

6 Conclusions

We have presented in this paper a new approach to forecast time series through the information from its image. This new approach can predict points from time series with different behaviors from the easiest time series (periodical) to a more complex behavior (chaotic).

The errors obtained with our technique demonstrated that the predicted points from the time series has good accuracy. However, this error measure only indicates the predicted amplitude of the time series but not the tendency of these predicted points which as we can observe from the results the predicted points preserve the original tendency and this is an interesting characteristic of this new approach.

The results that we presented in this paper are from the first stage of the whole investigation in which we only used the local information to predict some points. To future work we are going to use also global information in order to get better results than this first stage.

Using the information obtained from the training process we can also use that information to expand these results to complex networks. It is important to keep in mind that any natural phenomena can be able to be represented as a time series, and then it is possible to apply this approach in several phenomena.

References

[1] Ramírez-Amaro, K., Figueroa-Nazuno, J.G.: *Empleo de la Técnica Mapa Recurrente en el Análisis de los Índices de Teleconexión Atmosféricos*. México, D.F. Centro de Investigación en Computación, I.P.N. (2006). Reporte Técnico. ISBN: 970-36-0330-0.

[2] Kantz, H., Schreiber, T.: *Nonlinear Time Series Analysis*. Cambridge : Cambridge University Press (1997).

[3] Bautista-Thompson, E.F.: *Medición de la Predictibilidad de las Series de Tiempo: un Estudio Experimental*. México, D.F. Centro de Investigación en Computación. (2004). Ph. D. En Ciencias de la Computación.

[4] De Gooijer J. G, Hyndman R.J.: *25 Years of Time Series Forecasting*. Elsevier, International Journal of Forecasting 22 pp. 443-473, 2006.

[5] Landassuri-Moreno V. M.: *Predicción de Series de Tiempo con Descomposición Empírica en Modos, Algoritmo Genético y Redes Neuronales Artificiales*. Tesis de Maestría, Centro de Investigación en Computación del Instituto Politécnico Nacional. 2006.

[6] McCulloch, W., Pitts, W.: *A Logical Calculus of the Ideas Immanent in Nervous Activity*. Bulletin of Mathematical Biophysics, 7:115 - 133. 1943.

[7] Vapnik V.: *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.

[8] Newell A., Shaw J.C., Simon H.A.: *A Variety of Intelligent Learning*. eds. C Marshall, Yovits and Scott Cameron. London: Pergamon: in a general problem solver in Self-organizing Systems, pp. 153-189, 1960.

[9] Ramírez-Amaro K., Chimal-Eguía J.C.: *New Approach to Time Series Tracking*. 27TH International Symposium on Forecasting, New York City, USA 24-27, June 2007.

[10] Ramírez-Amaro K. : Técnica de Aprendizaje de Series de Tiempo estacionarias a partir de la información de su imagen. México, D.F. Centro de Investigación en Computación (2007). Maestría en Ciencias de la Computación (to be published).

[11] Ramírez-Amaro K., Chimal-Eguía J.C.: Una Nueva Téecnica para Aprender el Comportamiento de Series de Tiempo. 6ta. Conferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI'07), Orlando, Florida, EE.UU (2007).