# Forecasting Time Series by Means of Evolutionary Algorithms

Cristóbal Luque del Arco-Calderón,
Pedro Isasi Viñuela, and Julio César Hernández Castro

Universidad Carlos III de Madrid, C/Butarque 15, E-28911 Leganés, Spain
{cluque,isasi,jcesar}@inf.uc3m.es
http://www.uc3m.es

**Abstract.** The time series forecast is a very complex problem, consisting in predicting the behaviour of a data series with only the information of the previous sequence. There is many physical and artificial phenomenon that can be described by time series. The prediction of such phenomenon could be very complex. For instance, in the case of tide forecast, unusually high tides, or sea surges, result from a combination of chaotic climatic elements in conjunction with the more normal, periodic, tidal systems associated with a particular area. Too much variables influence the behaviour of the water level. Our problem is not only to find prediction rules, we also need to discard the noise and select the representative data. Our objective is to generate a set of prediction rules. There are many methods tying to achieve good predictions. In most of the cases this methods look for general rules that are able to predict the whole series. The problem is that usually the time series has local behaviours that don't allow a good level of prediction when using general rules. In this work we present a method for finding local rules able to predict only some zones of the series but achieving better level prediction. This method is based on the evolution of set of rules genetically codified, and following the Michigan approach. For evaluating the proposal, two different domains have been used: an artificial domain widely use in the bibliography (Mackey-Glass series) and a time series corresponding to a natural phenomenon, the water level in Venice Lagoon.

## 1 Introduction

Time series consists on a data sequence of measures along a time period:

$$y_1, y_2, ...., y_D$$

where the sub-index represents each unit of time. The goal is to predict the values of the series for $i' > D$. In other words, we use the set $\{y_1, \ldots, y_D\}$ to predict $y_{D+\tau}$, where $\tau$ is a non negative integer, which receives the name of prediction horizon. In time series related to real phenomenon we have and additional handicap: the measures may have noise. For example, an unusual hard wind will produce unusual measures. A good model needs to detect which elements in the data set can generate knowledge and refuse those that are noise. In this work we developed a model, based on genetic algorithms to search for good rules to detect local behaviours in a time series that allow to improve the prediction level in that area.

This results are applied to predict two examples: The Mackey-Glass series [11][14] as example of artificial series (without noise) and a example of real series extracted from the measures of the water level in the Venice Lagoon.

This problem had been usually approached by means of neural networks. These approaches have mainly used Radial Bases Function Neural Networks [11], with an algorithm that allows to change the net configuration, where neurons are added as needed, during the learning process. In [14], the algorithm combines the growth criterion of the resource-allocating network (RAN) of Platt [11], with a pruning strategy based on the relative contribution of each hidden unit to the overall network output. The resulting network leads toward a minimal topology for the RBFNN. Both papers, [14] and [11], show the results of applying that approach to the Mackey-Glass series. In [15] we can find a time series analysis using nonlinear dynamic systems theory and multilayer neural networks models. This strategy is applied to the time sequence of water level data, recorded from Venice Lagoon during the years 1980-1994. Recent works [5, 13], use a learning method that automatically selects the more appropriated training patterns to the new sample to be predicted. This training method follows a lazy learning strategy, in the sense that it builds approximations centered around the novel sample. Galvan et al. [5] applies his method for the Mackey-Glass, and for the Venice Lagoon time series. Following the Packard's work to predict dynamical systems [10], [8], [9], and using genetic algorithms [4] to generate predictions rules on a time series, we have applied some advanced genetic algorithms' techniques to attain better results. In the experiments explained in this paper, the constants $D$ and $\tau$ use the values $D = 24$, and $\tau = 1, 4, 12, 24, 28, 48, 72, 96$. In other words, we use the value of water level along 24 hours to predict the water level 1,4,12... etc. hours later.

## 2 Genetic Encoding of Rules for Time Series Forecasting

As example of artificial series we selected the Mackey-Glass Series, due to the extensive bibliography about it [11][14]. The chaotic time series known as Mackey-Glass series is defined by the following time-delay ordinary differential equation:

$$\frac{ds(t)}{dt} = -bs(t) + a\frac{s(t-\lambda)}{1+s(t-\lambda)^{10}}$$

with $a = 0.2$, $b = 0.1$ and $\lambda = 17$.

For our model we want to generate rules that make predictions. For the examples in this paper we use a value $D = 5$. A rule is an assert as "if the series at time unit 1 is smaller than 100 and bigger than 50, at time unit 2 is smaller than 90 and bigger than 40, at time unit 3 is smaller than 5 and bigger than -10,

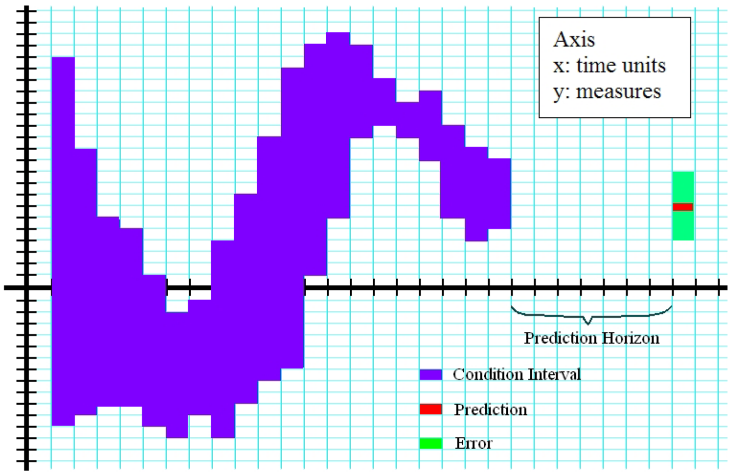

**Fig. 1.** Graphical representation of a rule.

at time unit 5 is smaller than 100 and bigger than 1, then the measure at time unit 5+$\tau$ will be 33 with an error of 5". It could be expressed as

$$\text{IF } (50 < x_1 < 100)\ AND\ (40 < x_2 < 90)\ AND\ (-10 < x_3 < 5)$$

$$AND\ (1 < x_5 < 100) \text{ THEN prediction} = 33 + / - 5$$

We can represent graphically a rule as in figure 1.

We encode this information in an individual as

$$(50, 100, 40, 90, -10, 5, dc, dc, 1, 100, 33, 5)$$

where $dc$ means "don't care". Now we apply the genetic algorithms' paradigm. Two individuals can generate an offspring. This offspring inherits each gene from one parent. A gene is a pair $(LL, UL)$, where $LL$ is the lower limit for a time instant, and $UL$ is the upper limit for the same instant. In other words, the offspring receives a gene from a parent with equal probability for each time instant. The offspring doesn't inherit parent's predictions and errors. We can see an example above:

Parent 1: $(50, 100, 40, 90, -10, 5, dc, dc, 1, 100, 33, 5)$
Parent 2: $(\mathbf{60}, \mathbf{90}, \mathbf{10}, \mathbf{20}, \mathbf{15}, \mathbf{30}, \mathbf{40}, \mathbf{45}, \mathbf{dc}, \mathbf{dc}, \mathbf{60}, \mathbf{8})$
Offspring: $(50, 100, \mathbf{10}, \mathbf{20}, -10, 5, \mathbf{40}, \mathbf{45}, \mathbf{dc}, \mathbf{dc}, ?, ?)$

Obviously, the offspring's "prediction" and "error" are not assigned (and appeared as "?" in the above representation). Once generated, an offspring may suffer mutation of some gene. At this point we need to divide the data set in two subsets: a training set and a test set, as we do with neural networks. In Neural Networks we use the training set to train the net, and we use the test set to verify that the training has been right. We do something similar in our model: we use the training set to see how good our individual is and after the process, we use the test set to verify the training. Let $C$ an individual, at first we calculate the prediction and the error of $C$ using the training data: we look for five consecutive values of time in the series at the point $i$, $(x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4})$ that fits the conditions of the individual:

$$(50 < x_i < 100)\ AND\ (10 < x_{i+1} < 20)$$

$$AND\ (-10 < x_{i+2} < 5)\ AND\ (40 < x_{i+3} < 45)$$

and we assign $v_i = x_{i+4+\tau}$. If this five values fits the conditions of $C$, we say $C(i) = true$, otherwise $C(i) = false$. Then we make a multiple regression on the variables $(x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4})$ for all the points $i$ of the series. The prediction $p_i$ will be the regression function applied to five consecutive values if and only if $C(i) = true$. In other words, the prediction will be a function of the form

$$p_i(x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}) = a_0 x_i + a_1 x_{i+1} + a_2 x_{i+2} + a_3 x_{i+3} + a_4 x_{i+4}$$

where $a_j$ are constants, for $j = 1, ..., 4$. Then, for each point $i$ in the time series in which $(x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4})$ fits the conditions of the individual (i.e. $C(i) = true$), we will have a real value $v_i$ and a prediction $p_i$. So, the error we use is the maximum absolute error, $e$, of each prediction to the real value for all the points, $i$, that fit the conditions of the individual. In other words:

$$e = Max_i\{|p_i - v_i| \mid C(i)\}$$

In our model we look for individuals which can predict the maximum number of points with the minimum error possible. The fitness function we used was:

```
IF ((N_C>1) AND (e < VAR_MAX)) THEN
        fitness = (N_C*10) - e
ELSE
    fitness = f_min
```

where $C$ is the individual, and $N_C$ is the number of data points in the training data set satisfying the conditions of $C$ (in other words, $N_C = \sharp\{i|C(i)\}$). VAR_MAX is a constant that makes the fitness function punishes individuals with a variance greater than VAR_MAX. *f_min* is a minimal value assigned to the individual when the rule is not fitted.

## 3 Evolution of Simple Rules

We cannot use an standard genetic algorithm, because of the fact that the medium values of the series have more data points than the extreme points or unusual set of data (for example an unusual high tide in the Venice Lagoon) deletes the individuals which makes predictions for that values of data, and the population becomes dominated by individuals which predicts medium values of the series. We decided to use a Michigan's approach [2] using a Steady-State strategy. In the Michigan's approach, the solution to the problem is the total population instead of only one individual. In the Pittsburgh approach [12] [7] [3], the solution to the problem is the best individual of the population, which chromosome encode a set of rules. We decided to use a Michigan's approach because in a complex time series we can find a lot of rules, and for the Pittsburgh approach this produces very big individuals that consume lot of memory and make his fitness evaluation too slow. We apply the Michigan's approach selecting each generation only two parents by three rounds trial to generate only one offspring. Then we replace the nearest individual to the offspring in phenotypic distance. That is, we find the individual whose prediction is nearest to the offspring's prediction, and replace it by the offspring if and only if the offspring fitness is better than the individual's fitness. If this doesn't happen, there isn't any change this generation. That strategy generates a diverse population, in which each individual makes a prediction different to the others individuals, instead of genetic clones of the best individual, produced by the standard genetic algorithm method. Finally, after each execution of the model (75.000 generations), we store in a file, that we called "pool", the individuals which predicts more than 5 points in the training set (and not only one as we do in a standard genetic algorithm model), an execute again the process. After some executions we have a file with a set of individuals. Some individuals predicts the same points of the test set, so the final prediction is the mean of all predictions (we must remember that, possibly, not all the individuals could predict a point of the series).

The last step is to generate the initial population of individuals. We divide the prediction range (i.e., $(-50, 150)$ for the water level) in 40 intervals of 5 centimeters. We create a "void" individual for each interval. For example, for interval $(40, 45)$ we have the "void" individual:

$$(150, -50, 150, -50, 150, -50, 150, -50, 150, -50, ?, ?)$$

Clearly, the upper limit for this "void" individual is -50, and the lower limit is 150, so this individual's rules cannot be complied at any point of the time series. Then we search for all time unit $t$ in the training set the measures $m(t)$ such $m(t) \in (40, 45)$, and adjust the void individual for this interval by this way: we take the 5 measures in the interval of time $[t-4-\tau, t-\tau]$, and for all $n \in [1, 5]$ we adjust each hour of the void individual: if the upper limit for the $n$ hour of the individual is lower than the measure of the $t-\tau-5+n$ time unit, $m(t-\tau-5+n)$, we take the measure as upper limit; if the lower limit for the n hour of the individual is greater than the measure of the $t-\tau-5+n$ time unit, $m(t-\tau-5+n)$, we take the measure as lower limit. We repeat the process for each hour and for each interval to generate the initial population of 40 individuals.

## 4 Results

In Table 1 we compare our results with the results obtained in [15] for the high tides prediction in the Venice Lagoon. The experiments have been done with a training data set of 45.000 measures, along 75.000 generations, and the predictions on a data set of 10.000 measures. Individuals use the measures of 24 hours to predict the water level a number of hours equal to the prediction horizon. The individuals in the pool is the number on individuals that we obtained after some executions. "Percentage of prediction" is the percentage of points in the test series which can be predicted by, at least, an individual in the pool. The rest of points cannot be predicted by anyone individual. The error in Table 1, column RMS and in [15] is the root mean square prediction error, where the error is

$$e = \frac{1}{2}(x - \bar{x})^2$$

**Table 1.** Comparative of predictions for the tides of the Venice Lagoon.

| Prediction Horizon | Individuals in the pool | Percentage of prediction | MAE | NRMS | RMS | Error in [15] |
|---|---|---|---|---|---|---|
| 1 | 10412 | 91,3% | 4,22 | 0,12 | 3,37 | 3,30 |
| 1 | 3475 | 97,2% | 5,54 | 0,15 | 4,30 | 3,30 |
| 4 | 3254 | 99,1% | 10,10 | 0,29 | 8,26 | 9,55 |
| 12 | 3227 | 98,0% | 10,35 | 0,30 | 8,46 | 11,38 |
| 24 | 3200 | 99,3% | 10,60 | 0,31 | 8,70 | 11,64 |
| 28 | 3038 | 98,8% | 13,93 | 0,41 | 11,62 | 15,74 |
| 48 | 3076 | 97,8% | 13,19 | 0,40 | 11,28 | - |
| 72 | 2870 | 99,7% | 16,99 | 0,51 | 14,45 | - |
| 96 | 2613 | 99,5% | 19,08 | 0,57 | 16,04 | - |

The results of the error show an improvement of the prediction level starting from a prediction horizon of 4, and show similar results for a prediction horizon of 1. In all the cases, it has been tried to maximize the percentage of the series that could be predicted by the method. With lower levels of series predicted, even better results, in terms of error, could be reached. It is interesting to remark that, even when the prediction horizon grows up, the number of rules are, more or less, similar,and the percentage of prediction does not decrease. The method seems, therefore, to be stable to the variations of prediction horizon. This property is very interesting, because it points out that the rules are adapted to special and local features of the series. Additionally, it can be seen that as the prediction horizon grows up, less rules are needed to predict even higher percentage of the series. All this without affect, significantly the error committed.

The error in column MAE is the mean of the absolute error. The NRMS error is the normalized RMS error. All errors are measured in centimeters. In Table 1 two experiments with prediction horizon 1 are showed. In the first experiment a value of 12 for the constant VAR_MAX is used. In the second one a value of

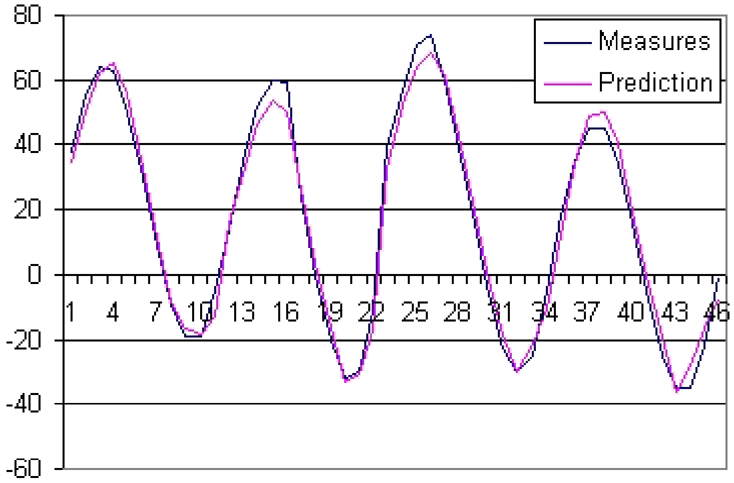

**Fig. 2.** Prediction of water level with horizon 1.

20 is used. A lower value causes the need of more individuals to increase the percentage of prediction, but decreases the mean error. A greater value for the constant VAR_MAX obtain a greater mean error but we need less individuals for a greater percentage of prediction.

In Table 2 the results of a cross-validation are shown. The total set of data has 50.000 measures of the level water.

**Table 2.** Cross-validation.

| Training Set | Prediction Set | Percentage of prediction | Error (MAE) | Error (RMS) |
|---|---|---|---|---|
| [0, 30.000] | [30.000, 50.000] | 99,1% | 5,67 | 4,33 |
| [10.000, 40.000] | [0, 10.000]∪[40.000, 50.000] | 98,9% | 5,78 | 4,45 |
| [20.000, 50.000] | [0, 20.000] | 98,7% | 5,79 | 4,45 |

In graphs 2 and 3 we can see how our model predicts the water level of for a prediction horizon of 1 and 12 hours.

In graph 4 we can see how our model predicts the water level of Venice in a case of abnormally high tides for a prediction horizon 1.

In table 3 we have compared of our algorithm with the results in [11] for the Makey-Glass series with prediction horizon of 85, and the results in [14].

## 5 Conclusions

Our model uses some very interesting tricks to better face the time series forecasting problem, even though the measures could contains noise. Our model not

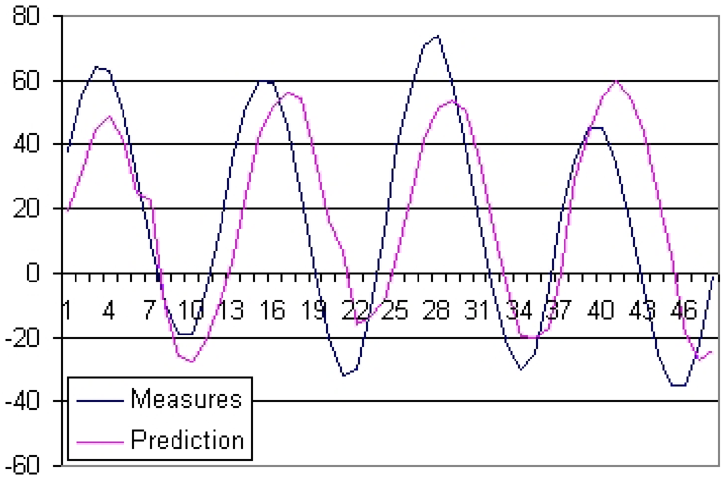

**Fig. 3.** Prediction of water level with horizon 12.

**Table 3.** Mackey-Glass series comparative.

| Prediction Horizon | Individuals in the pool | Percentage of prediction | Error | Error in [14] | Error in [11] |
|---|---|---|---|---|---|
| 50 | 3416 | 78,9 % | 0,025 | 0,040 | - |
| 85 | 2582 | 78,2 % | 0,046 | - | 0,050 |

only gives us predictions, it can also tell as when an abnormal measure is coming. For example, when the percentage of prediction is around a 99% and there isn't any individual to predict for a measure sequence, our model tells us that an abnormal behavior of the series is approaching. Additionally, the model is able to make much better predictions in situations than, being normal, are very unusual. This is due to the fact that the model do not try to generalize all the series, by the opposite it construct small set of rules that better adapt to all situations in a local way.

By increasing the percentage of series predicted, the method performs as any traditional method of generalization, that is, taken into account the whole series to make predictions. In the case that a reduction of the mean of the global error was needed, our approach could be also useful increasing the number of rules allowed. If, by the opposite, it is important to better predict exceptional situations, even though the global error is worse, the system could be adjusted to do so.

Another important feature of this approach, is that the method is able to find the regions of the series, where the behaviour is far from being generalizable. When the series contains regions that have some special particularities, that are different one each other, the method, not only finds this regions, but it constructs particular rules for better predict this special regions.

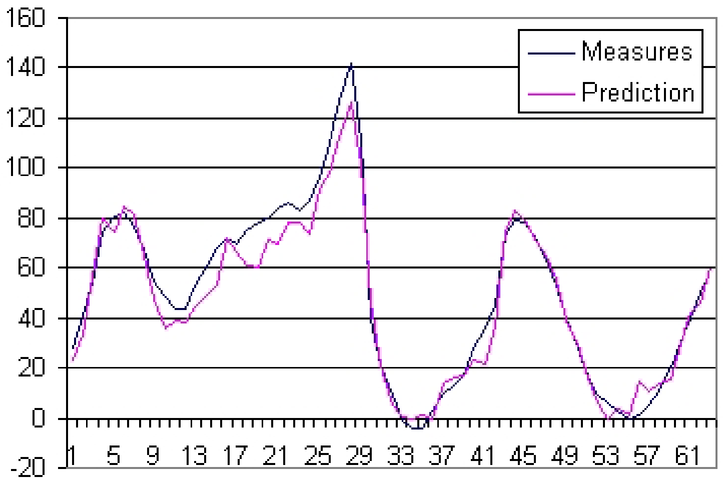

**Fig. 4.** Prediction of a high tide with horizon 1.

The method is also easily generalizable to others domains different from times series. In particular all domains of inductive learning, where many training examples could be obtained, are susceptible of applying our approach. We are now in the way of modifying the representation schemata of the rules, to makes it more complex and more complete.

## Acknowledgements

Investigation supported by the Spanish Ministry of Science and Technology through the TRACER project under contract TIC2002-04498-C05-4

## References

1. T. Bäck, H.P. Schwefel.: Evolutionary Algorithms: Some Very Old Strategies for Optimization and Adaptation. In Perret-Gallix (1992), pp. 247-254.
2. L.B. Booker, D.E. Goldberg, J.H. Holland: Classifier Systems and Genetic Algorithms. Artificial Intelligence No 40 (1989), pp. 235-282.
3. K.A. De Jong, W.M. Spears, F.D. Gordon: Usign Genetic Algorithms for Concept Learning. Machine Learning 13 (1993), pp. 198-228.
4. D.B. Fogel: An introduction to simulated evolutionary optimization. IEEE transactions on neural networks, vol 5, n 1, jan 1994.
5. I.M. Galván, P. Isasi, R. Aler, J.M. Valls: A selective learning method to improve the generalization of multilayer feedforward neural networks. International Journal of Neural Systems, Vol 11, No 2 (2001), pp. 167-177.
6. J.H. Holland: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975).

7. C.Z Janikow: A Knowledge Intensive Genetic Algorithm for Supervised Learning. Machine Learning 13 (1993), pp. 189-228.
8. T.P. Meyer, N. H. Packard: Local Forecasting of High-Dimensional Chaotic Dynamics, Nonlinear modeling and forecasting. 1990; editors, Martin Casdagli, Stephen Eubank, pp. 249-263.
9. M. Mitchell: An introduction to Genetic Algorithms, Cambridge, MA: MIT Press (1996), pp. 55-65.
10. N. H. Packard: A genetic learning algorithm for the analysis of complex data. complex systems 4, no 5 (1990), pp. 543-572.
11. J. Platt: A Resource-Allocating Network for Function Interpolation. Neural Computation, 3 (1991), pp. 213-225.
12. S.F. Smith: A Learning System Based on Genetic Adaptative Algorithms. Ph.D. Thesis, University of Pittsburgh (1980).
13. J. Valls : Selección Diferenciada del Conjunto de Entrenamiento en Redes de Neuronas mediante Aprendizaje Retardado. Ph.D. Thesis, Universidad Carlos III de Madrid (2004).
14. L. Yingwei, N. Sundararajan, P. Saratchandran. A Sequential Learning Scheme for Function Aproximation using Minimal Radial Basis Function Neural Networks. Neural Computation, 9 (1997), pp. 461-478.
15. J.M. Zaldívar, E. Gutiérrez, I.M. Galván, F. Strozzi, A. Tomasin: Forecasting high waters an Venice Lagoon using chaotic time series analysis and nonlinear neural network. Journal of Hydroinformatics 02.1 (2000), pp. 61-84.