Temporal self-organized meta-learning for predicting chaotic time series

Rigoberto Fonseca[1], Pilar Gómez-Gil[1]

[1] National Institute of Astrophysics, Optics and Electronics, Tonantzintla, Mexico
rfonseca@inaoep.mx, pgomez@acm.org

Abstract. To predict future values in chaotic systems is difficult but indispensable in several real applications. Over the last years, some authors have been focusing on a meta-learning process of how to combine models to improve prediction accuracy. This research proposal addresses the meta-learning problem of how to combine models using different parts of the prediction horizon. Our aim is to improve the long-term prediction achieved by the current state of the art. We propose to split the prediction horizon in three parts: short, medium and long term prediction horizons. Next in each horizon, we can extract knowledge about what model has the best performance. Thus, we can improve the long-term prediction using different models in each prediction horizon. However, the search space increases and poses nontrivial difficulties because the models could be combined in many ineffective ways. To avoid that, we propose the use of auto-organization. In this paper, we present some preliminary results of our first idea, combining models in different prediction horizons.

Keywords: meta-learning, time series prediction, chaotic time series, self-organization.

1 Introduction

Chaotic time series are cataloged as unpredictable, due its high sensibility to initial conditions [1]. Despite of that, many applications deal with chaotic systems and require a reasonable estimation of future values. For this reason, many domains are looking for an improvement of the accuracy obtained by current prediction models, for example in financial applications, load forecasting or wind speed [2]. Nevertheless, the problem of predicting multi-step-ahead, based on data captured from the chaotic system, is still an open problem [2]. Several works have tackled this problem mainly using statistical models and models based on computational intelligence. Available forecasting algorithms can be roughly divided into a few groups. Examples of simple algorithms are moving average and single exponential smoothing. Complex systems, commonly used by statisticians, are based on ARIMA models. Examples of models based on computational intelligence include neural networks and support vector machines. In addition, models have been used stand-alone or as a combination of several strategies [3].

© J. A. Olvera-López et al. (Eds.)
Special Issue: Advances in Pattern Recognition
Research in Computing Science 61, 2013, pp. 13-22
Paper Received 20-02-2013 and Accepted 22-04-2013

In an effort to find the best predictors, Crone et al. [4] analyzed the results obtained by different models competing in the international forecasting tournament NN3 [5]. From that analysis, they concluded three important ideas: combinations of models obtained the best results; some models have better performance than other models depending on the number of steps to predict, that is, the size of the prediction horizon; data features determine the relative performance of different models.

A very important problem when using a combination strategy is to decide what models must be combined and how to combine them. The process used by human experts starts with inspecting the data. Next, the models are selected and adjusted according to their experience. High time and money costs of expert's analysis motivate finding automatic approaches. In the last years, several works have been published related to this issue. For example, Lemke and Gabris [6] presented an interesting work using meta-learning. Meta-learning automatically induces a meta-model from a meta-training set, (data about training data). Given a new prediction task, this meta-model is able to return the best model or combination of models chosen from a model set [7]. The authors extracted features from around 222 time series. This collected data is the meta-data used to train an expert system. The authors outperformed individual methods and combinations of all methods involved in their experiments. Other researchers have obtained good results using self-organization for building combinations of classifiers on no time-dependent domains, for example [8] [9]. Inspired by these successes, we want to investigate how predictor models can cooperate in a self-organized way. Besides, we want to include the use of different prediction horizons in the meta-learning process.

This paper is organized as follows: Section 2 describes the involved problem, including main concepts associated with this research, research questions, objectives and main contributions; section 3 describes the proposed methodology to achieve the objectives. As a starting point, we empirically evaluated if a combination of models in different prediction horizons could improve the prediction accuracy. We call this strategy "temporal combination," described in section 4. Section 5 presents an experiment comparing temporal combination with the most used combination strategy, known as average of predictions. Finally, section 6 presents some conclusions.

2 Problem Statement

2.1 Main concepts

For a time-series prediction system, a sequence of n elements sampled from the past forms a training series; the sequence of m values to predict is known as a prediction horizon. The first future value to be estimated is represented by x_{n+1}. If the estimation of this value is calculated using d past values, a model F that returns a future value may be described as:

$$x_{n+1} = F(x_n, x_{n-1}, \dots, x_{n-d}) \tag{1}$$

A critical factor in predicting time series is to determine the value of d. Chaos theory contains some interesting ideas for finding suitable values for this regard.

The sequence $\{x_{n+1}, x_{n+2}, \dots, x_{n+m}\}$ represents a prediction horizon greater than one, known as multi-step ahead prediction. There are two forms to archive this sequence; one is estimating the complete horizon in a single iteration. A second strategy, known as iterative prediction [4] and used in this research, consists of estimating one value each time, using the previous predicted value for calculating the next prediction.

For many prediction applications, the best results have been achieved combining different models [5]. Diversity among the members of a set of models is deemed to be a key issue in models combination [10]. There are many strategies for combining models. The simple average of predictions is one of the most used, due to its simplicity and good accuracy. In this strategy, each prediction element is the average of all model's estimations. Let C be the prediction horizon, obtained as the average of predictions of k models. Then the elements of C are:

$$c_i = \frac{1}{k}\sum_{j=1}^{k} x_i^j \tag{2}$$

where x_i^j is the i-prediction, $0 \le i \le m$, obtained from the jth-model.

In general, the accuracy of a model comes from comparing their estimation $\hat{x}$ with the corresponding real values over the prediction horizon. There are several metrics for this error estimation, being the most used the mean square error (MSE) and the symmetric mean absolute percentage error (SMAPE) [11], defined as:

$$MSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{x}_i - x_i)^2} \tag{3}$$

$$SMAPE = \frac{1}{m}\sum_{i=1}^{m}\frac{|\hat{x}_i - x_i|}{\frac{1}{2}(\hat{x}_i + x_i)}100 \tag{4}$$

Meta-learning has become an important tool for designing prediction applications. Castiello and Fenalli [7] state the following definition of meta-learning:

Let A be a set of learning algorithms and T a set of tasks. Let $a_A(t)$ be the best algorithm in A applicable to a specific task t, for each $t \in T$, and $c(t)$ a characterization of the chosen task t. Then a meta-learning process is an automatic mechanism that starting from the meta-data set:

$$\{\langle c(t), a_A(t)\rangle : t \in T\} \tag{5}$$

induces a meta-model which is able to predict, for a new task, the best model in A. Consequently, the construction of meta-data set is a crucial part in the process of meta-learning. The selected features should cluster the time series correctly. That is, to group the most similar and separates the most different.

A system is self-organized if it acquires a temporary or functional spatial structure without specific interference from outside [12]. A good sample of the flexibility and success of self-organization are the Self Organizing Maps (SOM), proposed by Teuvo Khonen [13]. They are a kind of neural network with unsupervised learning. The training of the SOM uses competitive learning, which starts looking for what is the neuron most similar to the example shown, that is, the winner. Then it uses a collaborative strategy for updating weights of neurons in the winner's neighborhood.

2.2 Research questions

We propose to search for answers to the following questions:

1. How can we automatically find the right methods to combine and the right way to combine them, in order to improve multi-step ahead prediction in a chaotic time series?
2. How can we extract and exploit the knowledge of the models that work best in different prediction horizons?
3. How can self-organization of prediction methods improve the prediction of a combination of prediction methods?

2.3 Objectives

With the aim of answering the research questions, we have the following main objective in this research: *to develop a meta-learning algorithm capable of building, in a self-organized way, combinations of models considering different prediction horizons on chaotic time series.*

The commitment of this research is to obtain a better prediction accuracy than the models presented in the state of the art. We will compare our method mainly with the work of Lemke and Gabris [6] for their good results. Our results are expected to be a statistically significant improvement in prediction accuracy.

To achieve our general objective, we have the following specific objectives:

1. Define general guidelines for combining models in different prediction horizons, in order to improve the multi-step prediction performance.
2. Develop a meta-learning method considering different prediction horizons, to train an expert system builder of combinations of models.
3. Develop a strategy for self-organizing models, promoting collaboration among them during the meta-learning process.

The expected contributions of this research are:

1. A new strategy to combine prediction models, considering different prediction horizons.
2. A time series meta-data builder, able to find the best model in a search space previously defined.
3. A meta-learning method for training an expert system combining models.
4. A self-organized method for meta-learning in the context of predicting chaotic time series.

3 Proposed methodology

Based in the KDD process [14], we defined the main steps for achieving each of the objectives proposed in this research, which are detailed next. Tasks contributing to the development of the method are validated before declaring the task as ended. The complete method will be validated by comparing it with Lemke and Gabris work [6] and other state of the art works.

1. Create a target data set: select a set of chaotic time series and a set of prediction models. Based on the state of the art we have selected the following: statistics models (ARIMA, Random Walk and Exponential Smoothing) and computational intelligence models (Recurrent Neural Networks and Support Vector Machines). This step also includes:
   (a) Data cleaning and preprocessing: remove noise mainly outliers and approximate missing values.
   (b) Data reduction and projection:
      (i) define representative features of time series (e.g. standard deviation, trend, skewness and largest Lyapunov exponent [6]),
      (ii) for each model, define its parameters and possible values (e.g. number of delay neurons, number of neurons in the hidden layer and training algorithm),
      (iii) define a set of basic strategies for combining models (e.g. simple average, stacking with probability distribution [15], and rotation forest [16]).
   (c) Model evaluation: define metrics for evaluating models in the multi-step prediction task. The most common metrics for assessing prediction are MSE and SMAPE (see equations 3 and 4). In addition, run tests of statistical significance, as the null-hypothesis significance test.
   (d) Develop a time series meta-data builder, able to find the best model in a search space previously defined. An interesting option to select the best model can be Monte Carlo cross-validation [17]. Also, review other alternatives for selecting models.
2. Define a strategy for combining models in different prediction horizons.
   (a) Evaluate the existing strategies of combination of models. Decide which of them, if any, allows models to effectively combine and exploit the best performance of different algorithms in different prediction horizons.
   (b) Propose a combination strategy using different prediction horizons.
   (c) Evaluate the proposed strategy comparing it with the best strategies of combination of models found in the state-of-the-art work.
3. Design a meta-learning algorithm considering different prediction horizons, to train an expert system builder of combinations of models.
   (a) Find a meta-learning strategy able to build an expert system to define combinations of models.
   (b) Extend this meta-learning strategy to allow the combination of models in different prediction horizons. A possible combination could be using the short-term model outputs to enhance the initial conditions of a long-term model.

(c) Compare the performance accuracy of expert systems, trained by the two meta-learning strategies, both the original and the extended.

4. Develop a strategy for self-organizing models, to promote collaboration among them during the meta-learning process
   (a) Analyze the current strategies for self-organization literature, particularly those focused on building combinations of models. Include an analysis of negative correlation [18], aimed to seek a diversity of models.
   (b) Extend the meta-learning algorithm obtained from the third objective, adding the selected strategy of self-organization.
   (c) Compare the new meta-learning algorithm with that obtained by the third objective.
5. Develop a prediction system to exploit the ability of the expert system trained.

The following section shows the progress made so far, with respect to the first two points of the proposed methodology.

4 Temporal combination of models

A desired prediction horizon can be divided into three parts, each with the same number of elements, named short-term, medium-term and long-term. Our goal is to combine models with the best performance in short, medium and long term, as illustrated in figure 1. The prediction models are previously trained, and each model predicts the entire horizon. The prediction of the combination will consist of the prediction of the three models in their prediction horizons. The prediction horizons include the left bound but not the right bound, except the long-term prediction that includes the right boundary.

The selection of the best models in each forecast horizon requires some preprocessing. The original training set is divided into two series, one to train the models and other to evaluate the three prediction horizons. For each model, a SMAPE is calculated. The model selected for short-term horizon will be the one with the smallest value of short-term SMAPE averaged for all series. A similar procedure is followed to select models of the horizons of medium and long term.

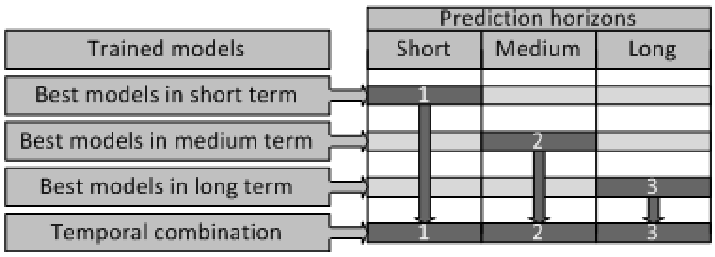

Fig. 1. The temporal combination of prediction models previously trained. The result is composed by the models predictions in their different prediction horizons.

Having defined the models of temporal combination, these are trained with the original training series. Each model produces the full horizon of prediction of size m; predictions are represented by $\{x^s\}$, $\{x^m\}$, and $\{x^l\}$ for selected models of short, medium and long term, respectively. We take a segment from each model prediction. The horizons short, medium and long terms are the same size b. Complete temporal prediction C_T is obtained by joining the three prediction horizons. The binding is expressed in equation 6.

$$C_T = \left\{ \begin{array}{c} \{x_i^s, n+1 \leq i < n+b\} \cup \\ \{x_i^m, n+b \leq i < n+2b\} \cup \\ \{x_i^l, n+2b \leq i < n+3b\} \end{array} \right\} \quad (6)$$

The prediction C_T is assessed by calculating SMAPE according to short, medium and long term prediction horizons.

5 Preliminary results

In this section, we show some preliminary results. In this experiment, a set of time series was modeled using several prediction models, all based on a NAR neural networks [19]. Then a temporal combination of models was built and compared with two other kinds of combinations.

5.1 Data description

For this first experiment time series were obtained from the NN3 prediction tournament which can be downloaded from: http://www.neural-forecasting-competition.com/NN3/datasets.htm. We used the available reduced set, which consists of 11 time series representing a homogeneous population of empirical business time series. Each training sequence contains between 116 and 126 items, while the prediction horizon is composed of 18 future values for all series. The set of values to predict is called the test set.

The set of models used in this experiment is composed of different non-linear auto-regressive neural networks (NAR) [19]. NAR is a recurrent dynamic network with feedback connections enclosing several layers. In this experiment, different models with the same base form are NARs trained with different parameters. Notice that, if a NAR is trained using different algorithms, their weight values will be different and consequently its performance may vary. For that reason, this experiment considers the training algorithm as a parameter.

The parameters used to generate the models are three: the number of delay neurons (1 to 5), the number of neurons in the hidden layer (1 to 5) and the training algorithm (12 in the neural networks toolbox of MATLAB). In total, there are 300 models with the same NAR form. The training algorithms used in this experiment are: 'trainbfg' (BFGS quasi-Newton backpropagation (BP)) 'trainbr' (Bayesian regulation BP), 'traincgb' (Conjugate gradient BP with Powell-Beale restarts), 'traincgf' (Conjugate gradient BP with Fletcher-Reeves updates), 'traincgp' (Conjugate gradient BP with Polak-Ribiére updates), 'traingd' (Gradient descent BP), 'traingda' (Gradient descent with adaptive learning rate BP), 'traingdm' (Gradient descent with momentum BP), 'traingdx' (Gradient descent with momentum and adaptive learning rate BP), 'trainlm' (Levenberg-Marquardt BP), 'trainoss' (One-step secant BP), 'trainrp' (Resilient BP).

5.2 Experiment setup and analysis of results

The aim of our first experiment is to test whether a temporary combination can perform better than the most commonly used combination in the state of the art. This last, described in section 2.1, is based on averaging the predictions of the all models, throughout the prediction horizon. We compare the results of this temporal combination with a combination made with the three models that best predicted the complete horizon. All models were trained using the same set of training series. Then each model predicted the entire horizon and SMAPE was calculated for each series. Next, we calculated the mean SMAPE of all the series and the top three models with minimum mean SMAPE are selected. Each combination models are trained with all the training set. The outputs of each combination of models are compared with the test set. The experiments were executed 10 times to remove the bias caused by the instability of neural networks. Next, we conducted an evaluation of statistical significance in predicting each series and all predictions. The estimated error is calculated using SMAPE both in the whole prediction horizon as in short, medium and long term horizons. The results are shown in Table 1. First column indicates the ID of the series, the second column shows the error of the combination based on average, the third column shows the error of the combination of the top three models and the fourth column is the error of the temporal combination.

Table 1. Comparison of the three combinations: average, “top three” and temporal in the series of prediction tournament NN3.

No.	Mean SMAPE of Average combination	Mean SMAPE of Top three combination	Mean SMAPE of Temporal combination
1	3.87	4.39	4.34
2	39.05	40.22	57.37
3	97.17	92.41	111.96
4	28.60	28.11	27.56
5	3.10	3.62	3.75
6	4.52	4.52	5.15
7	5.86	5.36	6.89
8	24.93	29.86	29.38
9	11.76	12.68	12.43
10	41.04	32.34	39.28
11	24.92	22.05	23.89
Mean	25.89 +/-26.21	25.05 +/-24.70	29.27 +/-30.75

The last row in the table shows the mean of SMAPE with its corresponding standard deviation. The best result was obtained by the combination of the top three models. However, the statistical significance test showed that the difference between the means of the models is not significant. Indeed, our combinations of models obtained a better performance for some series. The temporal combination obtained the best results in series number 1 and number 4.

Notice that the performance of a model depends on the involved time series. For some cases, the temporal combinations obtained the best result. We expect that increasing the diversity of base models will improve the results of the temporal combination, this according to [10]. On the other hand, the obtained results motivate us to explore different strategies for combining models. Finally, an improvement in the model selection criterion could reduce variability of results.

6 Conclusions

In this paper, we present the initial ideas for the creation of a new algorithm to predict chaotic time series using strategies taken from self-organization, meta-learning and combination of models. From a first experiment, we obtained empirical evidence of the viability of our proposal. This experiment compared the proposed temporal combination with the combination of models based on average strategy, which is the most commonly used in the state of art; also we compared with the combination of the best three models. Since the combination of models in different prediction horizons outperformed the other two strategies, we conclude that it is feasible to design automatic methods able to create temporal combinations. Nevertheless, it is necessary to explore other strategies for combining models, selecting models and extend the set of base models.

Acknowledgements

R. Fonseca thanks the National Council of Science and Technology (CONACYT), México, for a scholarship granted to him, No. 234540. This research has been partially supported by CONACYT, project grant No. CB-2010-155250

References

1. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. Cambridge University Press (2003)
2. De-Gooijer, J., Hyndman, R.: 25 years of time series forecasting. International Journal of Forecasting 22(3), 443-473 (2006) Twenty five years of forecasting.
3. Makridakis, S., Hibon, M.: The M3-Competition: results, conclusions and implications. International Journal of Forecasting 16(4), 451-476 (2000) The M3- Competition.
4. Crone, S., Hibon, M., Nikolopoulos, K.: Advances in forecasting with neural networks?

Empirical evidence from the NN3 competition on time series prediction. International Journal of Forecasting 27(3), 635-660 (2011)

5. Crone, S., Nikolopoulos, K., Hibon, M.: Automatic Modelling and Forecasting with Artificial Neural Networks– A forecasting competition evaluation. Final Report for the IIF/SAS Grant 2005/6, International Institute of Forecasters (April 2008)
6. Lemke, C., Gabrys, B.: Meta-learning for time series forecasting and forecast combination. Neurocomputing 73(10-12), 2006-2016 (2010)
7. Castiello, C., Fanelli, A.: Computational Intelligence for Meta-Learning: A Promising Avenue of Research. In Jankowski, N., Duch, W., Grabczewski, K., eds. : Meta-Learning in Computational Intelligence 358. Springer Berlin Heidelberg (2011) 157-177
8. García-Pedrajas, N., Hervás-Martínez, C., Ortiz-Boyer, D.: Cooperative Coevolution of Artificial Neural Network Ensembles for Pattern Classification. IEEE Transactions on Evolutionary Computation, 271-302 (2005)
9. Kordik, P., Cerny, J.: Self-organization of Supervised Models. In Jankowski, N., Duch, W., Grabczewski, K., eds. : Meta-Learning in Computational Intelligence 358. Springer Berlin Heidelberg (2011) 179-223
10. Kuncheva, L., Whitaker, C.: Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. Machine Learning 51, 181-207 (2003)
11. Armstrong, J.: Long-range forecasting from crystall ball to computer 2nd edn. John Wiley & Sons (1985)
12. Haken, H.: Information and Self-Organization: A Macroscopic Approach to Complex Systems 3rd edn. 40. Springer, Stuttgart, Germany (2006)
13. Haykin, S.: Neural Networks A Comprehensive Foundation Second Edition edn. Pearson Prentice Hall (1999)
14. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. AI Magazine 17(3), 37-54 (1996)
15. Dzeroski, S., Zenko, B.: Is Combining Classifiers with Stacking Better than Selecting the Best One? Machine Learning 54, 255-273 (2004)
16. Rodríguez, J., Kuncheva, L., Alonso, C.: Rotation Forest: A New Classifier Ensemble Method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(10), 1619-1630 (2006)
17. Song Xu, Q., Zeng Liang, Y., Ping Du, Y.: Monte Carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration. Chemometrics 18(2), 112-120 (2004)
18. Islam, M. M., Yao, X., Murase, K.: A constructive algorithm for training cooperative neural network ensembles. Neural Networks, IEEE Transactions on 14(4), 820-834 (july 2003)
19. Ardalani-Farsa, M., Zolfaghari, S.: Chaotic time series prediction with residual analysis method using hybrid Elman-NARX neural networks. Neurocomputing 73(13-15), 2540-2553 (2010)
20. Crone, S., Hibon, M., Nikolopoulos, K.: Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. International Journal of Forecasting 27(3), 635-660 (2011) Special Section 1: Forecasting with Artificial Neural Networks and Computational Intelligence Special Section 2: Tourism Forecasting.