Evolutionary Method for Real-Word Times Series Prediction

Tiago A. E. Ferreira
Center for Informatics
Federal University of Pernambuco
Av. Prof. Luiz Freire, s/n, CDU
50732-970, Recife - PE - Brazil
E-Mail: taef@cin.ufpe.br

Germano C. Vasoncelos
Center for Informatics
Federal University of Pernambuco
Av. Prof. Luiz Freire, s/n, CDU
50732-970, Recife - PE - Brazil
E-Mail: gcv@cin.ufpe.br

Paulo J. L. Adeodato
Center for Informatics
Federal University of Pernambuco
Av. Prof. Luiz Freire, s/n, CDU
50732-970, Recife - PE - Brazil
E-Mail: pjla@cin.ufpe.br

Abstract

This paper presents the Time-delay Added Evolutionary Forecasting (TAEF) method for time series prediction which performs an evolutionary search of the minimum necessary number of dimensions embedded in the problem for determining the characteristic phase space of the phenomenon generating the time series. The method proposed consists of an intelligent hybrid model composed of an artificial neural network (ANN) combined with a modified genetic algorithm (GA). Initially, the TAEF method finds the most fitted predictor model for representing the series and then performs a behavioral statistical test in order to adjust time phase distortions that may appear in the representation of some series.

1 Introduction

More recently, new promising approaches based on artificial neural networks (ANNs) have been proposed for the nonlinear modeling of time series [9]. However, in order to define a solution to a given problem, ANNs require the setting up of a series of system parameters.

In this work, a systematic procedure based on an hybrid intelligent system approach is proposed for the automatic search of the important system parameters that solve time series prediction problems. The adopted method consists of a combination of a standard ANN architecture with a modified genetic algorithm (GA) [2] which efficiently searches and defines 1. the minimum number of (and the specific) temporal lags, based on F. Takens theorem [8], 2. the best ANN structure in terms of the number of processing units, 3. the most fitted training algorithm that boosts the prediction performance, and 4. a behavioral statistical test carried out at the prediction model output to fix relative phase distortions in the series representation.

2 Time Series Problems

The aim when applying prediction techniques to a given temporal series is to identify certain regular patterns present in historical data. In this context, a crucial factor for a good forecasting performance is the correct choice of the time lags considered for representing the series. Such relationship structures among historical data constitute a d-dimensional phase space, where d is the minimum dimension capable of representing such relationship. The work of F. Takens [8] has proved that if d is sufficiently large, such built phase space has the same dynamics and is homeomorphic to the phase space which generated the time series.

The big problem in reconstructing the original state space is naturally the correct choice of the variable d, or, more specifically, the correct choice of the important time lags.

3 The TAEF Method

The method proposed in this work — Time-delay Added Evolutionary Forecasting (TAEF) method — tries to reconstruct the phase space of a given time series by carrying out a search for the minimum dimensionality necessary to reproduce the phenomenon generator of the times series. The proposed procedure is a intelligent hybrid system based on an ANN architecture (multilayer perceptron network - MLP) trained with a modified GA [2] which not only searches for a number of the ANN parameters but also for the adequate embedded dimension represented in the lags.

The scheme describing the proposed algorithm is based on the iterative definition of the three main elements: 1. the underlying information necessary to predict the series (the minimum number of lags); 2. the structure of the model capable of representing such underlying information for the purpose of prediction (the number of units in the ANN structure); and 3. the appropriate algorithm for training the model.

Following this principle, the important parameters defined by the algorithm are: 1. **The number of time lags to represent the series:** initially, a maximum number of lags (*MaxLags*) is defined by the user and a GA can choose any number of lags in the interval [1, *MaxLags*] for each individual of the population; 2. **The number of units in the ANN hidden layer:** the maximum number of hidden layer units (*NHiddenmax*) is determined by the user and the GA chooses, for each candidate individual, the number of units in the hidden layer (in the interval [1, *NHiddenmax*]); 3. **The training algorithm for the ANN:** RPROP [6], Levenberg-Marquardt [5], Scaled Conjugate Gradient [4], One Step Secant Conjugate Gradient [1] are candidates for the best algorithm for training the ANN.

The algorithm starts with the user defining a minimum initial fitness value (*MinFit*) which should be reached by at least one individual of the population in a given GA round. The fitness function is defined as $Fitness = \frac{1}{1+MSE}$, where *MSE* is the Mean Squared Error of the ANN and will be formally defined in the next section.

In each GA round, a population of *M* individuals is generated, each of them being represented by a chromosome (here, $M = 10$). Each individual is in fact a three-layer ANN where the first layer is defined by the number of time lags, the second layer is composed of a number of hidden processing units (sigmoidal units) and the third layer is composed by one processing unit (prediction horizon of one step ahead).

The stopping criteria for each one of the individual are the number of epochs ($NEpochs$), the increase in the validation error (Gl) and the decrease in the training error (Pt).

The best repetition with the smallest validation error is chosen to represent the best individual. Following this procedure, the GA evolves towards a good fitness solution (which may not be the best solution yet), according to the stopping criteria: number of generations created ($NGen$) and fitness evolutions of the best individual ($BestFit$).

After this point, when the GA reaches a solution, the algorithm checks if the fitness of the best individual paired or overcame the initial value specified for the variable *MinFit* (minimum fitness). If this is not the case, the value of *MaxLags* (maximum number of lags) is increased by the unit and the GA procedure is repeated to search for a better solution. The objective here is to increase the possible number of lags in the lag set until a solution of minimum fitness is reached.

However, if the fitness reached was satisfactory, then the algorithm checks the number of lags chosen for the best individual, places this value as *MaxLags*, sets *MinFit* with the fitness value reached by this individual, and repeats the whole GA procedure. In this case, the fitness achieved by the best individual was better than the fitness previously set and, therefore, the model can possibly generate a solution of higher accuracy with the lags of the best individual (and with the *MinFit* reached by the best individual as the new target). If, however, the new value of *MinFit* is not reached in the next round, *MaxLags* gets again the same value defined for it just before the round that found the best individual, increased by the unit (the maximum number of lags is increased by one). The idea here is that if the time lags found in the best individual were not capable of producing a higher fitness than the one previously found this may be because some important lag (or lags) was discarded. The state space for the lag search is then increased by one to allow a wider search for the definition of the lag set. This procedure goes on until the stop condition is reached. After that, the TAEF method chooses the best model found among all the candidates.

In order to conclude the definition of the method a last aspect had to be considered. During the development and test of the method, a peculiar prediction behavior was observed in the prediction model. The predictions of financial series were

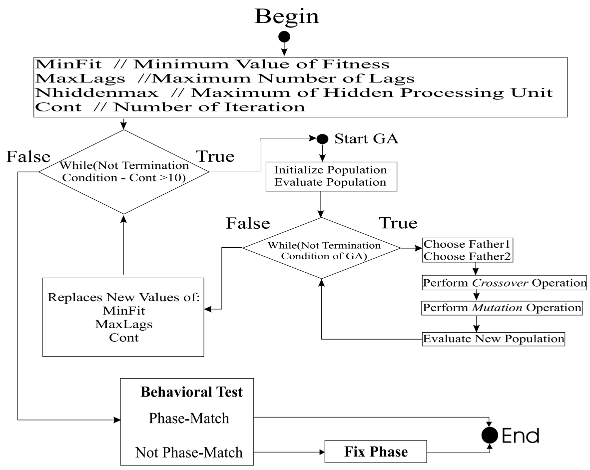

Figure 1: Algorithm for the TAEF method

always presented with a one step shift (delayed) with respect to the original data ("out-of-phase" matching). An interesting point to observe is that this one step delay behavior with respect to the actual series is similar to a random walk like model. Since it is a common sense in economics that financial times series behave like random walks, as a first approximation [3], it is not strange that predictor models generated for them show this one step time delay distortion [7].

After the best model is chosen when training is finished, an statistical test is employed to check if the network representation has reached an "in-phase" (without a one step shift) or "out-of-phase" matching. This is conducted by comparing the outputs of the prediction model with the actual series making use of the validation data. If this test (for example the t-test) accepts the "in-phase" matching hypothesis, the elected model is ready for practical use. Otherwise, the method carries out a new procedure to adjust the relative phase between the prediction and the actual time series. The validation patterns are presented to the ANN and the output of these patterns are re-arranged to create new inputs that are both presented to the ANN and set as the output (prediction) target. The approximation results for both the "in-phase" and "out-of-phase" models are measured and the best model (smaller MSE error) is elected as the final model. Figure 1 depicts the complete algorithm for the TAEF model construction.

4 Performance Evaluation

Most of the works found in the literature of time series prediction frequently employ only one performance criterion for model evaluation. Most of the times, the measure used is the MSE (mean squared error),

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(target_j \quad output_j)^2 \tag{1}$$

where *N* is the number of patterns, $target_j$ is the desired output for pattern *j* and $output_j$ is the predicted value for pattern *j*.

A second relevant measure is the $MAPE$ (Mean Absolute Percentage Error), given by

$$MAPE = \frac{100}{N} \sum_{j=1}^{N} \left| \frac{target_j \quad output_j}{X_j} \right| \tag{2}$$

where N, $target_j$, and $ouytput_j$ are the same MSE parameters, and X_j is the time series at point j.

A third performance measure is the U of Theil Statistics, or NMSE (Normalized Mean Squared Error), which is given by

$$Theil = \frac{\sum_{j=1}^{N} (target_j \quad output_j)^2}{\sum_{j=1}^{N} (target_j \quad target_{j+1})^2} \tag{3}$$

which associates the model performance with a random walk model. If the U of Theil Statistics is equal to 1, the predictor has the same performance of a random walk model. If the U of Theil Statistics is greater than 1, then the predictor has a worse performance than a Random Walk model, and if the U of Theil Statistics is less than 1, the predictor is better than a random walk model.

Another relevant evaluation measure considers the calculation of the correctness of Prediction of Change in Direction, or $POCID$ for short,

$$POCID = 100 \frac{\sum_{j=1}^{N} D_j}{N} \tag{4}$$

where,

$$D_j = \begin{cases} 1 & if (target_j - target_{j-1})(output_j - output_{j-1}) > 0, \\ 0 & otherwise. \end{cases} \tag{5}$$

The two last evaluation measures that include the freedom degrees, penalizing the models with additional parameters, are the Akaike (AIC) and Bayesian (BIC) information criteria. The AIC and BIC are approximated by

$$AIC = N \ln(MSE) + 2p \tag{6}$$

$$BIC = N \ln(MSE) + p + N \ln(p) \tag{7}$$

where N is the number of time series points, MSE is the Mean Squared Error and p is the number of freedom degrees.

5 Experimental Results

Two times series were used as a test bed for evaluation of the method proposed. All series were normalized to lie within the interval [0,1] and divided in three sets: training set (50% of the points), validation set (25% of the points) and test set (25% of the points). The AG parameters are the same for the two series, with a mutation probability of 10%, and crossover and mutation operations, as those reported in Leung et al [2]. For all the experiments carried out, the following system parameters were employed: initialization parameters — $MinFit = 0.99$, $MaxLags = 4$ and $NHiddenmax = 20$; stopping conditions for the GA — $NGen = 1000$ and $BestFit \leq 0.0001$; Stopping conditions for each individual — $NEpochs = 1000$, $Gl \leq 5\%$ and $Pt \leq 10^{-6}$.

5.1 Down Jones Series

The Dow Jones Industrial Average Index (DJIA) series corresponds to daily observations from 1st January 1998 to 26th of August 2003 (1420 points) of the DJIA index.

The hybrid model proposed automatically chose the lags 2, 4, 8, 6, 9 and 10 as the relevant lags for the series representation, defined 10 processing units for the hidden layer of the ANN, selected the algorithm Levenberg-Marquardt as the most fitted for the ANN training and classified the model as "out-of-phase" matching. Table 1 shows the results with all the performance measures for both cases: "out-of-phase" matching and if the prediction model had been chosen as "in-phase" matching. Of particular interest to this financial series are the measures shown by the statistics U of Theil (0.03), denoting a far better result than a random walk (see Section 4), and by the POCID which shows a 97% series approximation.

Table 1: Experimental Results for the DJIA Series

	In-phase Matching	Out-of-phase Matching
MSE	$8.4183 \cdot 10^{-4}$	$2.6841 \cdot 10^{-5}$
MAPE	1.15 %	0.20%
U of Theil	1.0006	0.0318
POCID	47.58%	97.14%
AIC	-2206.1	-3408.5
BIC	-1510.6	-2713.4

Figure 2 shows a comparative graph of the actual DJIA (solid lines) and the prediction generated by the TAEF method (dash lines) for the last 100 points of the test set, for both cases of prediction model classification ("in-phase" matching and "out-of-phase" matching). It is seen that the "out-of-phase" matching model chosen by the method was the correct choice.

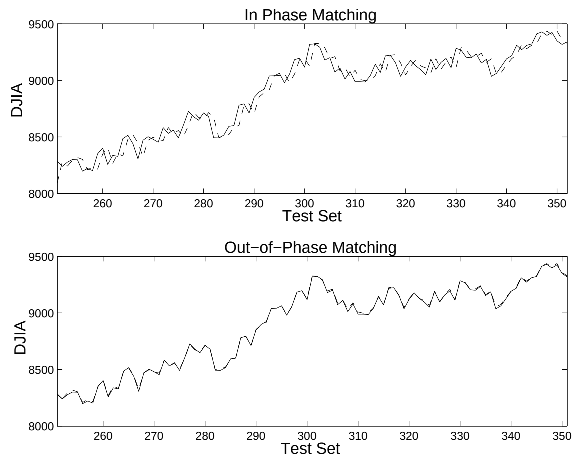

Figure 2: Prediction results for the DJIA series (test set): actual values (solid lines) and predicted values (dashed lines).

5.2 Nasdaq Series

The Nasdaq series (National Association of Securities Dealers Automated Quotation) corresponds to daily observations

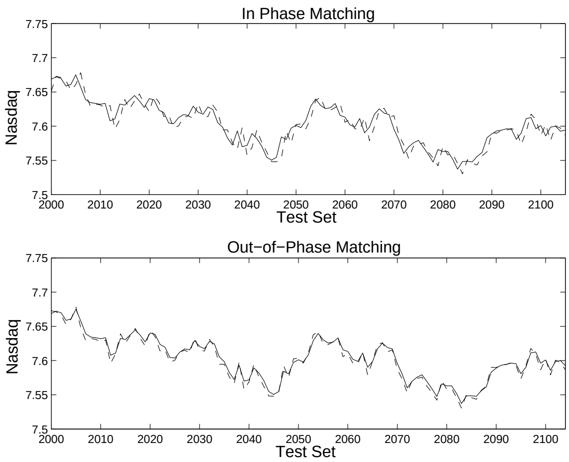

Figure 3: Prediction results for the Nasdaq series (test set): actual values (solid lines) and predicted values (dashed lines).

from 2nd February 1971 to 18th of June 2004 of the Nasdaq index (8428 points).

For the prediction of the Nasdaq series, the TAEF method identified the lags 3, 4, 6 and 8 as the relevant to the problem, defined 11 processing units in the hidden layer of the network, elected the Levenberg-Marquardt algorithm as the most fitted for the ANN training and classified the model as "out-of-phase" matching. Table 2 shows the results with all the performance measures for both cases: "out-of-phase" matching and if the prediction model had been chosen as "in-phase" matching. Of particular interest to this financial series are the measures shown by the statistics U of Theil (0.17), denoting a far better result than a random walk (see Section 4), and by the POCID which shows a 89.6% series approximation. Figure 3 shows a comparative graph of the actual Nasdaq series (solid lines) and the prediction generated by the TAEF method (dash lines) for the last 100 points of the test set, for both cases of prediction hypotheses ("in-phase" matching and "out-of-phase" matching). It is clear that the "out-of-phase" matching model chosen by the method was the correct choice.

Table 2: Experimental Results for the Nasdaq Series

	In-phase Matching	Out-of-phase Matching
MSE	$2.1449 \cdot 10^{-5}$	$3.2374 \cdot 10^{-6}$
MAPE	0.20 %	0.08%
U of Theil	1.1441	0.1720
POCID	52.71%	89.63%
AIC	-22342.4	-26310.1
BIC	-21391.2	-25358.9

6 Conclusions

This paper has presented an intelligent hybrid system approach, the Time-delay Added Evolutionary Forecasting (TAEF) method, which consists of an artificial neural network combined with a modified genetic algorithm and a behavior test of phase match hypotheses carried out at the model's output for the solution of time series forecasting problems.

The experimental results using a set of consistent performance measures with six different metrics showed that this system can boost the performance of time series prediction. The experimental validation of the method was carried out on two complex and relevant time series.

With the introduction of the behavior test for identifying whether the prediction model is "in-phase" or "out-of-phase" with the series to be forecasted, the TAEF method was able to classify if a given time series tends or not to a Random Walk like model, thus adjusting the model if necessary. Such adjustment is conducted on the model constructed without the use of any additional training phase nor the use of any additional training data (the same original validation data is employed). Only one additional epoch is used for presenting the original validation data and deciding which of the models generated (in-phase or out-of-phase) produces the best series approximation.

A systematic study is yet necessary to determine any possible limitations of the method when dealing with other types of components found in other different real world time series such as trends, seasonality, impulses, steps, and other nonlinearities. Taking that into account, other time series with those components are being collected to carry out a broader investigation.

References

[1] R. Battiti. One step secant conjugate gradient. *Neural Computation*, 4:141–166, 1992.

[2] F. H. F. Leung, H. K. Lam, S. H. Ling, and P. K. S. Tam. Tuning of the structure and parametrs of the neural network using an improved genetic algorithm. *IEEE Transaction on Neural Networks*, 14(1):79–88, January 2003.

[3] Terence C. Mills. *The Econometric Modelling of Financial Time Series*. Cambridge University Press, Cambridge, 2003.

[4] M. F. Moller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533, 1993.

[5] J. J. Mor. The levenberg-marquardt algorithm: Implementation and theory. In *Proceedings of Springer-Verlag in Numerical Analysis (Lecture Notes in Mathematics)*, pages 105–116, 1977.

[6] M. Reidmiller and H. Braun. A direct adaptive method for faster backpropagation learng: The rprop algorithm. In *Proceedings of the IEEE Int. Conf. on Neural Networks (ICNN)*, pages 586–591, San Francisco, 1993.

[7] Renate Sitte and Joaquin Sitte. Neural networks approach to the random walk dilemma of financial time series. *Applied Intelligence*, 16(3):163–171, May 2002.

[8] Floris Takens. Detecting strange attractor in turbulence. In A. Dold and B. Eckmann, editors, *Dynamical Systems and Turbulence*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381, New York, 1980. Springer-Verlag.

[9] G. Zhang, B. Eddy Patuwo, and M. Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14:35–62, 1998.