Online Modeling of Real-World Time Series
Through Evolving AR Models

Ahmad Kalhor
School of Electrical and Computer Engineering,
University of Tehran and Institute for International Energy Studies, Tehran, Iran
akalhor@alumni.ut.ac.ir

Hossein Iranmanesh
Assistant Prof., Department of Industrial Engineering,
University of Tehran and Institute for International Energy Studies, Tehran, Iran
hiranmanesh@ut.ac.ir

Majid Abdollahzade
Department of Mechanical Engineering,
K. N. T. University of technology and Institute for International Energy Studies, Tehran, Iran
m.abdollahzade@gmail.com

Abstract—There is a growing demand to model and predict real-world time series such as natural or financial time series. In this paper, we propose an online modeling approach for real-world time-series through evolving Auto Regressive (eAR) models. At first, it is supposed that a real-world time series can be modeled as a summation of some basic signals. Then, it is shown adaptive AR models which can evolve in the number of lags are suitable models to generate such time series. To evolve an adaptive AR model, it switches to a pre-learned adaptive AR model which has a lag less or more than main model. The capability of the proposed eAR model in detection proper number of lags and prediction is shown through an illustrative example and a real-world application to the prediction of monthly time series of U.S. coal consumption.

Keywords- online modeling; real-world time series; evolving AR models; prediction; coal consumption;

I. INTRODUCTION

Modeling and prediction of real-world time series such as time series of natural phenomena and financial problems is a demanding task. During the last decades, many predictive models from linear classic prediction formulas up to different nonlinear models have been suggested for time series. Classical linear models like AR, ARMA and ARIMA have superiority in straightforward optimization and performing noise and parameter analyzing, [1]-[3]. However, to detect nonlinear nature of data and to represent both temporal and spatial variability, linear models are not enough. Hence, many nonlinear models such as Artificial Neural Networks (ANN) [4]-[7], support vector machines [8]-[9], matrix decomposition techniques [10]-[11], immune systems [12] and fuzzy systems [13]-[15] as well as hybrid techniques of linear and nonlinear models, [16]-[17], have been suggested.

Since temporal behaviors of real-world time series are affected by many known and unknown exogenous inputs, they have time-varying structures. Hence, online modeling approaches which evolve the structure of model seem to be suitable solutions.

Among various nonlinear models, the evolving methods for neuro-fuzzy models –due to their interpretable, flexible and trainable structures– have been developed vastly during last decade. Two particularly influential works in this area of research are [18] and [19]. Kasabov proposes an adaptive online learning algorithm as a dynamic evolving neural–fuzzy inference system (DENFIS) in [18]. Angelov and Filev introduce an online identification approach for the TS model in [19], where evolving clustering method along with a concept of potential is used to define the antecedent parts of the rules. This approach has been modified and extended in [20] and [21]. In [22], to evolve a specific form of TS Fuzzy Model, Lughofer suggests to use a modified version of vector quantization for new rule generation. In [23], an agile nonlinear model called “Adaptive Habitually Linear and Transiently Nonlinear Model” (AHLTNM) is introduced to follow uncertain and fast-varying processes.

Although the above mentioned evolving models can follow the spatial variation in the structure, they cannot follow the structure when the proper number of lags of time series changes in input. Actually, due to impacts of the number and dynamic influence of exogenous inputs, the required number of lags of a time series in a proper modeling may change.

In this paper we introduce an evolving model which can detect the required number of lags, in an interval. Since evolving the structure of nonlinear models both in spatial variability and number of lags is difficult, we propose to evolve adaptive linear AR models in number of lags: evolving AR models. It is discussed that if time series be represented as a summation of some basic signals, evolving AR models are capable to follow their behavioral modes.
The rest of the paper is organized as follows: Section II introduces evolving AR models. In Section III a learning algorithm of evolving AR model is proposed. In Section IV through an illustrative example and a case study (U.S. monthly coal consumption) the capability of evolving AR models in detection of required number of lags and prediction is shown. The paper is concluded in Section V.

II. EVOLVING AR MODELS

In this section, we introduce a type of adaptive AR model which can evolve in number of lags; we call it evolving AR (eAR) models. It is shown such model is suitable model for real-world time series.

At first, it is supposed a time series could be modeled as summation of some basic signals. Here, a typical basic signal is defined as follows:

\[ f(t) = rt^hp^i \quad t = 1, 2, \ldots \]  

where \( r, p \in \mathbb{C} \) and \( h \in \{0, 1\} \). Choosing a suitable weighted summation of above defined basic signals, a large set of behavioral modes can be made: exponential, oscillatory, decaying oscillatory, time multiplied exponential, ramp and parabola. Actually, the above form of signal definition is a mathematical representation of terminologies used usually by experts when they explain about behavioral modes of real-world time series such as price, temperature and etc.

It is supposed a time series, in an interval, could be modeled as um of \( n \) basic signals as follows:

\[ y_t = \sum_{k=1}^{n} r_k t^{h_k} (p_k)^i \quad t = 1, 2, \ldots \]  

where \( r_k, p_k \in \mathbb{C} \) and \( h_k \in \{0, 1\} \). It can be shown easily an AR model can generate the considered form of signal in (2). For this purpose, the Z-transform of the equation (2) is computed as follows:

\[ Y(z^-1) = \sum_{k=1}^{n} F_k(z^-1) \]

\[ F_k(z^-1) = \begin{cases} r_k / (1 - p_k z^-1) & h_k = 0 \\ r_k p_k z^-1 / (1 - p_k z^-1) & h_k = 1 \end{cases} \]  

where \( p_k, r_k \in \mathbb{C} \) denote pole and residue of \( k \)th basic signal as well as the order of the pole \( p_k \) can be shown with \( h_k + 1 \). Since the time series have real numbers, for each non-real pair of \((r_k, p_k)\) in a basic signal its conjugated pair \((\overline{r_k}, \overline{p_k})\) must be existed in another basic signal. The Z-transfer equation of (3) can be rewritten as a rational function of \( z^-1 \) with real coefficients:

\[ Y(z^-1) = \frac{b_0 + b_1 z^-1 + \ldots + b_m z^{-m}}{1 + a_1 z^-1 + \ldots + a_n z^{-n}}, \quad m < n \]  

By applying the inverse Z-transform operation to equation (4) and with respect to time shifting property, the following linear difference equation or AR model is obtained.

\[ y_t = -a_1 y_{t-1} - \ldots - a_n y_{t-n} + b_0 \delta(t) + b_1 \delta(t-1) + \ldots + b_m \delta(t-m) \]  

where the unit impulse signals in right part of the equation (5) provide initial conditions for the time series, however, they can be ignored after some initial sample times.

Although the represented signal in (2) includes various behavioral modes of a time series, they are not suitable for many real-world time series whose dynamic order or behavior modes change adaptively. To supply required complexity to generate such time series, it is allowed the number of basic signals, \( n \), and their structural parameters in (2) change adaptively. To model such time series, we propose to use adaptive AR model whose number of lags, \( n_t \), can change through the time:

\[ y_t = -a_1 y_{t-1} - a_2 y_{t-2} - \ldots - a_n y_{t-n} \]  

III. THE LEARNING ALGORITHM OF AN EAR MODEL

In this section, we introduce a learning algorithm to identify eAR models. For this purpose, two following issues must be addressed:

1. Suitable number of lags in an eAR models
2. Evolving method of an AR model to upper or lower number of lags

To determine the suitable number of lags for an eAR model, we use a spectrum analyzing manner. To this end, we extract the poles and corresponding residues of the AR model. Next, some poles which have low absolute residues are ignored; the number of remaining poles shows the proper number of lags which AR model should have.
To provide an evolving procedure for AR models, we utilize a parallel learning strategy. For this mean, besides the main model, two neighboring adaptive AR models with one less lag and one more lag (as inputs) are identified. If it is revealed the number of existing lags in main model is insufficient, it can evolve by switching to the neighboring model with one more lag (N-up model). In contrast, if it is revealed the number of existing lags in main model is extra; it can switch to the neighboring model with one less lag (N-down model). Fig. 1 shows the diagram of the above mentioned strategy.

Fig. 1. The evolving strategy of an adaptive AR model.

Here, some notes about the learning of eAR models are explained:

Note1- The linear parameters in an adaptive AR model can be updated by recursive least square model with a forgetting factor \( \xi \):

\[
\theta_{t+1} = \theta_t + P_t z_t e_t
\]

\[
P_t = \frac{1}{\xi} (P_{t-1} - P_{t-1} z_{t-1}^T z_t P_{t-1}) \quad 0 < \xi \leq 1
\]

\[
z_t = [y_{t-1} \ y_{t-2} \ldots \ y_{t-n}] \quad \theta_t = [a_1 \ a_2 \ldots \ a_n]
\]

\[
e_t = (s_t - y_t)
\]

where \( \theta_t \in \mathbb{R}^n \), \( P_t \in \mathbb{R}^{n \times n} \), and \( s_t \) denote linear parameters, covariance matrix and the original value of the time series at time \( t \), respectively. To use Equation (7), the linear parameters and covariance matrix must be initialized. To provide initial exploration power for RLS algorithm, \( P_1 = g I_n \) is chosen, where \( g \) a large positive value is. Initial value of linear parameters can be chosen at random or set to zero, \( \theta_1 = 0 \). The appropriate value for \( \xi \) depends on the case study. Smaller forgetting factors are more suitable for cases with higher changing rates and vice versa. Here, in this paper, we use \( \xi = 0.95 \) to update linear parameters of adaptive AR models.

Note2- To decide that an extracted pole at \( t : p_k \) with residue \( r_k \) is significant or insignificant, we use an adaptive threshold \( \bar{r} \) which is defined as a ratio of maximum absolute residue.

\[
\bar{r} = \eta \max (r_k) \quad k = 1, 2, \ldots, n \quad 0 < \eta < 1 \quad (8)
\]

If \( |r_k| \leq \bar{r} \) then the \( k \)th pole is insignificant. Here, \( \eta \) acts as a separation coefficient. Larger values of \( \eta \) provide a smaller group of significant poles (basic signals) as well as lags for an AR model. In this paper we have chosen \( \eta = 0.01 \) to separate insignificant poles (lags).

Note3- To have a confidence about the accuracy of extracted poles and residues we consider the output error of the main AR model: \( e_t \). Computing \( E_{av} \) as mean of all former observed absolute errors and defining \( 0 < \lambda \leq 1 \) as a confidence coefficient, if \( |e_t| \leq \lambda E_{av} \) the accuracy of extracted poles and residues is satisfied. Smaller values of confidence coefficient increase the confidence and vice versa. In this paper we have chosen \( \lambda = 0.3 \). Also, to provide more confidence in an evolving procedure, after each evolving procedure, for a while which is considered \( 5n_t \) samples, no other evolving procedure is done.

Note4- At begin of evolving in the number of lags, based on an incremental learning method, the initial number of lags in the main model is chosen low: \( n_t = 2 \).

Now, the learning algorithm of the eAR model is presented in Table 1.

### Table 1. The learning algorithm of the eAR model

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step1.</strong></td>
<td>Define main AR model at ( t=1 ): let ( n_t = 2 ), ( \theta_1 = 0_{n_t \times 1} ) and ( P_{11} = 1001 ). Also, initialize this coefficients: ( \eta = 0.01 ), ( \xi = 0.95 ) and ( \lambda = 0.3 ) and then put ( E_{av} = 0 ) and ( g = 0 ).</td>
</tr>
<tr>
<td><strong>Step2.</strong></td>
<td>Define neighboring AR models, N-up (N-down): let ( n_t = n_t + 1 (n_t = n_t - 1) ), ( \theta_{n_t} = 0_{n_t \times 1} ) (( \theta_{n_t} = 0_{n_t-1 \times 1} ) and ( \theta_{n_t} = 0_{n_t-1 \times 1} ) ( P_{n_t+1} = 1001 ) ( P_{n_t-1} = 1001 )).</td>
</tr>
<tr>
<td><strong>Step3.</strong></td>
<td>Update the mean absolute value of errors: respecting to ( s_t ) as the value of time series at ( t ), update parameters of main AR model and...</td>
</tr>
</tbody>
</table>
neighboring models through (7) then compute $E_n = (e_{t+1}^n + iE_{n+1}^n)/(t+1)$ and let $q = q + 1$.

**Step4.** Check the considered confidence condition in Note3: if $|e_t| < \lambda E_n$ and $q > 5n$, go to next step, otherwise let $t = t + 1$ return to Step3.

**Step5.** Evolving the AR model: Extract all poles and residues of main AR model and N-up neighboring model [24].

5.1 For the main AR model, consider the pole which have smallest absolute residue (assume $k$'th pole). Respecting to given explanation in Note2, if $|r_k^*| \leq \tilde{r}$ then switch the main model to N-down AR model, let $q = 0$, $t = t + 1$ and return to Step 2.

5.2 For the N-up neighboring model, consider the pole which have smallest absolute residue (assume $k$'th pole). Respecting to given explanation in Note2, if $|r_k^*| > \tilde{r}$ then switch the main model to N-up AR model, let $q = 0$, $t = t + 1$ and return to Step 2.

IV. CASE STUDIES

In this section, we apply the learning algorithm of eAR model to a handmade time series as illustrative example and a case study: monthly time series of U.S. coal consumption. Through illustrative example we demonstrate the capability of the proposed learning algorithm in proper identification of an eAR models. Also, we demonstrate the capability of eAR models in prediction through the case study.

A. An Illustrative example

Here, we generate a hand-made time series by using three different summations of basic signals as follow:

$$s^i = 10 + \left( \cos(\pi/3) - 0.4 \sin(\pi/3) \right)$$

$$s^2 = s^1 + \left( e^{0.04t} \right) \left( 0.6 \cos(1.28t) - 1.67 \sin(1.28t) \right)$$

$$s^3 = s^2 + \left( 2 \cos(1.48t) - 0.399 \sin(1.48t) \right)$$

$S = \left\{ s_1^{100}, s_2^{100}, s_3^{100}, s_1^{100}, s_2^{100}, s_3^{100} \right\}$

Fig. 2 show the generated signal of this time series. As it is seen in Fig. 2 and (9), the time series are made from three different combination modes but in five parts.

![Fig. 2. The Handmade time series with different parts and different number of lags](image-url)

Now, we apply the presented learning algorithm in Table 1 to the generated time series. Fig. 3 shows the plot of estimated and original number of lags.

![Fig. 3. The diagram of original and estimated number of lags for the time series](image-url)

As it is seen in Fig. 3, the main adaptive AR model has evolved properly in number of lags just a while after the original number of lags changes. To present more details about the identified eAR model, Table 2 includes the extracted and the original poles of eAR model at different sample times.

<table>
<thead>
<tr>
<th>Sample times</th>
<th>Original and estimated poles</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>Original $[1 \ 0.5 \pm 0.866\ j]$</td>
</tr>
<tr>
<td></td>
<td>Extracted $[1 \ 0.4996 \pm 0.8658\ j]$</td>
</tr>
<tr>
<td>600</td>
<td>Original $[1 \ 0.5 \pm 0.866\ j \ 0.7 + 0.7741\ j]$</td>
</tr>
<tr>
<td></td>
<td>Extracted $[1 \ 0.499 \pm 0.8659\ j \ 0.691 + 0.71\ j]$</td>
</tr>
<tr>
<td>1170</td>
<td>Original $[1 \ 0.5 \pm 0.866\ j \ 0.7 \pm 0.7141\ j \ 0.095 \pm 0.995\ j]$</td>
</tr>
<tr>
<td></td>
<td>Extracted $[1 \ 0.4954 \pm 0.862\ j \ 0.686 \pm 0.7097\ j \ 0.0949 \pm 0.9949\ j]$</td>
</tr>
<tr>
<td>1790</td>
<td>Original $[1 \ 0.5 \pm 0.866 \ 0.7 + 0.7741\ j]$</td>
</tr>
<tr>
<td></td>
<td>Extracted $[1 \ 0.4984 \pm 0.862\ j \ 0.6799 + 0.7088\ j]$</td>
</tr>
<tr>
<td>2160</td>
<td>Original $[1 \ 0.095 \pm 0.995\ j]$</td>
</tr>
<tr>
<td></td>
<td>Extracted $[1 \ 0.4998 \pm 0.8659\ j]$</td>
</tr>
</tbody>
</table>

As it is understood from Table 2, the extracted poles are near to the original ones in the generated time series.

B. Monthly time series of U.S. coal consumption

Predicting the energy time series is a demanding task. Here, the time series of U.S. monthly coal consumption is considered as a case study. Coal, due to its relatively low cost and abundance, is used for generating about half of the electricity consumed in the U.S. It is the largest domestically-produced source of energy [25]. Hence having a accurate estimate for coal consumption is very important.
The data used in this case study for U.S. coal consumption prediction includes monthly values of the consumption of coal from Jan. 1973 up to April 2011 [25]. Fig. 4 shows the signal of the time series.

![Time series of coal consumption](image)

**Fig.4.** The monthly time series of coal consumption from 3 Jan. 1973 up to April 2011.

The last 12 (one year) values of the time series are considered for prediction purpose and the rest part of time series is used in training phase. At first, we apply the learning algorithm of eAR to the training part of time series. Fig. 5 shows the diagram of the detected number of lags during the sample times.

![Number of lags versus sample times](image)

**Fig.5.** The diagram of estimated number of lags for the time series

As it is seen in Fig. 5, the number of lags in eAR model has increased up to 9 lags. In other words, the number of significant modes estimated by eAR model changes through the time and it is less than 10 modes. Table 3 presents AR model, the extracted poles and their residues.

### Table 3. The AR model, extracted poles, their residues and the estimated signal function at end of training phase.

<table>
<thead>
<tr>
<th>AR model</th>
<th>N.of rules</th>
<th>RMSE</th>
<th>Learning Time* (sec)</th>
<th>Parameters setting**</th>
</tr>
</thead>
<tbody>
<tr>
<td>eAR</td>
<td>1</td>
<td>6294</td>
<td>2.58</td>
<td></td>
</tr>
<tr>
<td>FLEXFIS</td>
<td>9</td>
<td>8764</td>
<td>3.20</td>
<td>fac=0.3</td>
</tr>
<tr>
<td>DENFIS</td>
<td>18</td>
<td>9037</td>
<td>4.1</td>
<td>Distance-Thre.=0.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>epochs=30</td>
</tr>
<tr>
<td>ANFIS</td>
<td>12</td>
<td>10469</td>
<td>13.63</td>
<td>Range of inf.=0.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLP</td>
<td>10</td>
<td>10093</td>
<td>8.21</td>
<td>Learn. rate=0.003</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>epochs=100</td>
</tr>
</tbody>
</table>

* All the execution times are given for a Dell XPS M1530 2.2GHz Core 2 Duo notebook with 4GB RAM.

**All other parameters are default values suggested in the original algorithms.

As it is seen, the computed criterion: Root Mean Square Error (RMSE) for eAR is considerably lower than other approaches. Moreover, due to low number of parameters in eAR model, the computed learning time for this model is lower than other identified models. The results, demonstrate the superiority of the proposed eAR model in Prediction.

### Table 4 the prediction result of coal consumption time series

<table>
<thead>
<tr>
<th>Method</th>
<th>N.of rules</th>
<th>RMSE</th>
<th>Learning Time* (sec)</th>
<th>Parameters setting**</th>
</tr>
</thead>
<tbody>
<tr>
<td>eAR</td>
<td>1</td>
<td>6294</td>
<td>2.58</td>
<td></td>
</tr>
<tr>
<td>FLEXFIS</td>
<td>9</td>
<td>8764</td>
<td>3.20</td>
<td>fac=0.3</td>
</tr>
<tr>
<td>DENFIS</td>
<td>18</td>
<td>9037</td>
<td>4.1</td>
<td>Distance-Thre.=0.05</td>
</tr>
<tr>
<td>ANFIS</td>
<td>12</td>
<td>10469</td>
<td>13.63</td>
<td>Range of inf.=0.5</td>
</tr>
<tr>
<td>MLP</td>
<td>10</td>
<td>10093</td>
<td>8.21</td>
<td>Learn. rate=0.003</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>epochs=100</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, evolving AR model was introduced for online modeling of real-world time series. It was supposed that each time series can be represented as a summation of basic signals in each interval. Then, it was shown that adaptive AR models which can evolve in the number of lags were suitable to generate such time series. A learning algorithm of evolving AR model was introduced. According to this algorithm, the main adaptive AR model could evolve by switching to neighboring models with one lag less or one lag more. The proposed learning algorithm was applied to a handmade time series and a
real-world case study of monthly time series of the U.S. coal consumption. It was demonstrated that evolving AR model could detect the proper number of lags and the accuracy of extracted poles was favorable. Also, the prediction capability of the proposed evolving AR model in comparison to other known approaches was shown in both case studies.

REFERENCES