Applied Soft Computing 32 (2015) 481–493

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Time series forecasting with a neuro-fuzzy modeling scheme

Hung-Wen Peng, Shen-Fu Wu, Chia-Ching Wei, Shie-Jue Lee*

Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 804, Taiwan

ARTICLE INFO

Article history:
Received 30 April 2014
Received in revised form 13 January 2015
Accepted 31 March 2015
Available online 16 April 2015

Keywords:
Time series prediction
Neuro-fuzzy modeling
TSK fuzzy rules
Learning algorithm
Training patterns

ABSTRACT

Time series forecasting concerns the prediction of future values based on the observations previously taken at equally spaced time points. Statistical methods have been extensively applied in the forecasting community for the past decades. Recently, machine learning techniques have drawn attention and useful forecasting systems based on these techniques have been developed. In this paper, we propose an approach based on neuro-fuzzy modeling for time series prediction. Given a predicting sequence, the local context of the sequence is located in the series of the observed data. Proper lags of relevant variables are selected and training patterns are extracted. Based on the extracted training patterns, a set of TSK fuzzy rules are constructed and the parameters involved in the rules are refined by a hybrid learning algorithm. The refined fuzzy rules are then used for prediction. Our approach has several advantages. It can produce adaptive forecasting models. It works for univariate and multivariate prediction. It also works for one-step as well as multi-step prediction. Several experiments are conducted to demonstrate the effectiveness of the proposed approach.

1. Introduction

Time series prediction is concerned about the forecasting of future values based on a time series of the previously observed data. It has played an important role in the decision making process in a variety of fields such as finance, power supply, and medical care. One example is to use the previously collected data to predict the stock exchange indices or the closing stock prices [1–9]. Another example is to predict the electricity demand to avoid producing extra electric power [10–13]. If forecasting is done for one time step ahead into the future, it is called single-step or one-step prediction. Forecasting can also be done for two or more time steps ahead. In this case, it is called multi-step prediction [14–17].

Two approaches have been adopted for constructing time series forecasting models. The global modeling approach constructs a model which is independent of the target to be forecasted. For time series prediction, the conditions of the environment may vary as time goes on. A global model is not adaptive and thus accuracy suffers. A local model constructed by the local modeling approach [19,18] is dependent on the target to be forecasted and therefore is adaptive. Local models are usually characterized by using a small number of the neighbors in the proximity of the predicting sequence. Another issue in time series prediction is to determine the lags to be involved in the model. The lags have a big influence on the forecasting accuracy. For example, in steel making engineering [20,21], the furnace temperature will change after two to eight hours from the time when the materials are applied into the furnace. This indicates there is a time lag of two to eight hours for the temperature change. Furthermore, the lags may vary as time goes on and adjusting them is required [22]. Two strategies, direct [23] and iterative [24], have been traditionally adopted for constructing multi-step time series forecasting models. The difference between them lies on the incorporation of the forecasts of previous steps in the prediction of the current step. These strategies have their respective pros and cons. Due to accumulated errors, an iterative forecasting model may suffer from low prediction accuracy. On the other hand, a direct forecasting model can only acquire the estimated value of the specified step. However, all the estimated values up to the specified steps can be acquired by applying an iterative model. It should be noted that several research efforts have been paid to some new modeling strategies for multi-step time series forecasting. For example, a smart and adaptive modeling strategy was proposed in [17], which employs a PSO based heuristic to create flexible divides with varying sizes of prediction horizons under the multiple-input several multiple-output (MISMO) strategy.

Recently, machine learning techniques have drawn attention and useful forecasting systems based on these techniques have been developed [25]. The multilayer perceptron, often simply called neural network, is a popular network architecture in use for time series prediction [26–31]. Neural network encounters the local

* Corresponding author. Tel.: +886 75252000x4141; fax: +886 75254199.
E-mail address: leesj@mail.ee.nsysu.edu.tw (S.-J. Lee).

http://dx.doi.org/10.1016/j.asoc.2015.03.059
1568-4946/

minimum problem during the learning process and the number of nodes in the hidden layer is difficult to decide. Also, it is hard to provide a comprehensible expression for humans to understand or investigate. The k-nearest neighbor regression method is a non-parametric method which bases its prediction on the k nearest neighbors of the target to be forecasted [18,32]. However, it lacks the capability of adaptation and the distance measure adopted may affect the prediction performance. Fuzzy theory is incorporated for prediction in stock market [2,3,5,6,33,7,8,34]. However, membership functions need to be determined which is often a challenging task. Also, no learning is offered by fuzzy theory. Support vector regression provides high accuracy in time series prediction [35,12,9,13,16]. It is free from local minimum. However, the computational burden is heavy due to solving quadratic programming problems. Also, choosing kernels and hyperparameters may not be an easy task. To overcome these difficulties, the least squares form of support vector regression was used [18] and a multiple-kernel framework was adopted [36]. Neuro-fuzzy modeling is a hybrid approach, which takes advantage of both fuzzy theory and neural network learning techniques, for modeling complex relationships between inputs and outputs [5,37,38,32]. However, deriving an appropriate set of fuzzy rules automatically and refining the associated parameters efficiently are two of the challenges involved.

In this paper, we propose an approach based on neuro-fuzzy modeling for time series prediction. Given a predicting sequence, the local context of the sequence is located in a series of the observed data. A distance measure taking the trend of the data into account is adopted. Proper lags of relevant variables for prediction are selected and training patterns are extracted. Based on the extracted training patterns, a set of TSK fuzzy rules are constructed automatically by an incremental clustering algorithm. The parameters involved in the fuzzy rules are then refined by an efficient hybrid learning algorithm which incorporates a least squares estimator and the gradient descent method. The refined fuzzy rules constitute the predicting model and can then be used for prediction. Both direct and iterative forecasting models are developed. Our approach has several advantages. It can produce adaptive forecasting models. It works for univariate and multivariate prediction. It also works for one-step as well as multi-step prediction. The neuro-fuzzy modeling scheme is adopted since it incorporates the ideas of fuzzy theory and neural networks, offering good properties such as non-linear learning capability, quick convergence, and high accuracy. Furthermore, the rules obtained are comprehensible to human beings.

The rest of this paper is organized as follows. Section 2 describes the problem to be solved. Section 3 gives a brief description of the adopted neuro-fuzzy modeling technique. Our proposed approach is outlined in Section 4. Section 5 describes the process of extracting training patterns from a series of the observed data. Section 6 describes the process of deriving a forecasting model from the training patterns. A small example is given in Section 7. Experimental results are presented in Section 8. Finally, concluding remarks are given in Section 9.

2. Time series forecasting problem

Consider a series of real-valued observations [39]:

$$\mathbf{X}_0, Y_0, \mathbf{X}_1, Y_1, \ldots, \mathbf{X}_t, Y_t \tag{1}$$

taken at equally spaced time points t_0, $t_0 + \Delta t$, $t_0 + 2\Delta t$, ... for some process P, where Y_i denotes the value of the output variable (or dependent variable) observed at the time point $t_0 + i\Delta t$ and $\mathbf{X}_i$ denotes the values of m additional variables (or independent variables), $m \geq 0$, observed at the time point $t_0 + i\Delta t$. Time series prediction is to estimate the value of Y at some future time $t+s$, i.e., Y_{t+s}, by

$$\hat{Y}_{t+s} = G(\mathbf{X}_{t-q}, Y_{t-q}, \ldots, \mathbf{X}_{t-1}, Y_{t-1}, \mathbf{X}_t, Y_t) \tag{2}$$

where $s \geq 1$ is called the horizon of prediction, G is the predicting function or model, Y_{t-i} is the ith lag of Y_t, $\mathbf{X}_{t-i}$ is the ith lag of $\mathbf{X}_t$, and q is the lag-span of the prediction. For $s = 1$, it is called one-step prediction. For $s > 1$, it is called multi-step prediction. Also, if $m = 0$, it is univariate prediction; otherwise, it is multivariate prediction. For convenience,

$$\mathbf{Q} = \langle \mathbf{X}_{t-q}, Y_{t-q}, \ldots, \mathbf{X}_{t-1}, Y_{t-1}, \mathbf{X}_t, Y_t \rangle \tag{3}$$

is called the predicting sequence for predicting Y_{t+s}.

The prediction of Y_{t+s} can be regarded as a function approximation task. Two strategies are usually adopted to construct forecasting models [23,24]:

- Direct. Train on $\mathbf{X}_{t-q}$, Y_{t-q}, ..., $\mathbf{X}_{t-1}$, Y_{t-1}, $\mathbf{X}_t$, Y_t to predict Y_{t+s} directly, for any $s \geq 1$.
- Iterative. Train to predict Y_{t+1} only, but iterate to get Y_{t+s} for any $s > 1$.

These strategies work identically for the case of $s = 1$. However, for the case of $s > 1$, the iterative strategy cannot work for multivariate prediction since $\mathbf{X}_{t+1}, \ldots, \mathbf{X}_{t+s-1}$ are not available to obtain $\hat{Y}_{t+2}, \hat{Y}_{t+3}, \ldots, \hat{Y}_{t+s}$ successively for this case.

3. Neuro-fuzzy modeling

Neuro-fuzzy modeling concerns deriving a model for an unknown system based on a set of training input–output data observed for the system. The derived model consists of a set of fuzzy rules which, given a certain input, can be used to predict the system output through fuzzy inference. In this paper, we adopt the neuro-fuzzy modeling technique proposed in [40]. Two stages, creation of fuzzy rules and refinement of fuzzy rules, are involved in the modeling process. In the first stage, a set of TSK fuzzy rules are generated from given training patterns. In the second stage, the parameters involved in the fuzzy rules are refined by a hybrid learning algorithm. The refined fuzzy rules constitute the predicting model for the unknown system and can then be used to predict the output of the system.

Suppose the system we'd like to model has n input variables, denoted as $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$, and one output variable, y. A brief description of the neuro-fuzzy modeling technique is given below.

3.1. Creation of fuzzy rules

Assume that we are given a set $\mathcal{T}$ of N training patterns $(\mathbf{p}_v, q_v)$, $1 \leq v \leq N$, where $\mathbf{p}_v = \langle p_{1v}, p_{2v}, \ldots, p_{nv} \rangle$ and q_v denote the n input values and the desired output value, respectively, associated with the vth pattern. By applying an incremental clustering algorithm [38], the patterns in $\mathcal{T}$ are grouped into J clusters $C_1, C_2, \ldots, C_J$. Each training pattern is assigned to only one cluster. Each cluster C_j, $1 \leq j \leq J$, is characterized by $G_j(\mathbf{x})$ and h_j, where $G_j(\mathbf{x})$ is a distribution with mean $\mathbf{m}_j = \langle m_{1j}, m_{2j}, \ldots, m_{nj} \rangle$ and deviation $\boldsymbol{\sigma}_j = \langle \sigma_{1j}, \sigma_{2j}, \ldots, \sigma_{nj} \rangle$, and h_j is the height of C_j. Note that $\mathbf{m}_j$, $\boldsymbol{\sigma}_j$, and h_j of C_j are computed as follows:

- m_{ij} and σ_{ij} are the average and deviation, respectively, of the ith coordinate of the patterns contained in G_j, $1 \leq i \leq n$.
- h_j is the average of the desired output values of the patterns contained in G_j.