Expert Systems with Applications 42 (2015) 341–354

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Employing local modeling in machine learning based methods for time-series prediction

Shin-Fu Wu, Shie-Jue Lee *

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

ARTICLE INFO

Article history:
Available online 7 August 2014

Keywords:
Time series prediction
Machine learning
Local modeling
Nearest neighbors
Mutual information

ABSTRACT

Time series prediction has been widely used in a variety of applications in science, engineering, finance, etc. There are two different modeling options for constructing forecasting models in time series prediction. Global modeling constructs a model which is independent from user queries. On the contrary, local modeling constructs a local model for each different query from the user. In this paper, we propose a local modeling strategy and investigate the effectiveness of incorporating local modeling with three popular machine learning based forecasting methods, Neural Network (NN), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Least Squares Support Vector Machine (LS-SVM), for time series prediction. Given a series of historical data, a local context of the user query is located and an appropriate number of lags are selected. Then forecasting models are constructed by applying NN, ANFIS, and LS-SVM, respectively. A number of experiments are conducted and the results show that local modeling can enhance the estimation performance of a forecasting method for time series prediction.

1. Introduction

Time series is defined as a sequence of observations from a system and usually the time intervals between any two adjacent observations are identical. The purpose of time series prediction is to estimate the future data tendency with available historical information (Bisqaard & Kulahci, 2011; Janacek, 2001; Commandeur & Koopman, 2007; Fan & Yao, 2005). Therefore, it plays an important role in decision making in various kinds of application such as finance, healthcare, weather, electricity demand, industrial monitoring, etc. If the prediction is made to estimate one time interval ahead into the future, it is called one-step or single-step forecasting (Gooijer & Hyndman, 2006; Wei, 2005). For rapidly changing data such as stock market exchange, one-step prediction is more favorable to show its short-term tendency. If we want to estimate more than one time interval ahead, a multi-step prediction must be done.

Statistical models have been investigated in the past decades by researchers for time series prediction. The widely adopted autoregressive-moving-average (ARMA) model was proposed by Box and Jenkins (2008). This statistical model consists of two polynomial parts, one for the auto-regression and the other for moving average. Many generalizations and modifications of this model have been proposed, including autoregressive integrated moving average (ARIMA), auto-regressive conditional heteroskedasticity (ARCH), etc. However, the Box–Jenkins models require the underlying time series to be stationary in order to make a successful prediction. Other statistical methods, e.g., spectral analysis, Markov process, and Kalman filter, are based on the probability theory and prior knowledge of the underlying time series is required.

Recently, machine learning provides another perspective to the time series problem (Alpaydm, 2004; Štěpnička, Cortez, Donate, & Štěpničková, 2013). Because of the learning process, machine learning based methods can extract some of the interesting underlying characteristics of the system without human intervention. Neural networks have been widely used in time series prediction (Adhikari & Agrawal, 2011; Hagan, Demuth, & Beale, 1995; Khashei & Bijari, 2010). A common practice adopts feed-forward networks which take input from the given series of data by employing a sliding window. Neuro-fuzzy techniques (Lee & Ouyang, 2003) combine the advantages of both fuzzy logic (Bajestani & Zare, 2011; Chen & Tanuwijaya, 2011; Hwang, Chen, & Lee, 1998; Zadeh, 1965) and neural networks (Hagan et al., 1995) to solve the non-linearity and uncertainty of the problem. Adaptive Neuro-Fuzzy Inference System (ANFIS) proposed by Jang (1991, 1993), Jang, Sun, and Mizutani (1997) is one of the most influential neuro-fuzzy systems. It is a kind of Takagi–Sugeno fuzzy-rule based neural network with a hybrid learning algorithm. Least Squares Support Vector Machines (LS-SVM), proposed by

* Corresponding author. Tel.: +886 7 5252000x4141.
E-mail address: leesj@mail.ee.nsysu.edu.tw (S.-J. Lee).

http://dx.doi.org/10.1016/j.eswa.2014.07.032
0957-4174/

Suykens and Vandewalle (1999), Suykens, Gestel, Brabanter, Moor, and Vandewalle (2002), are least squares versions of support vector machines (SVM) which are a class of kernel-based learning methods introduced for nonlinear function estimation. With LS-SVM, one finds the solution by solving a set of linear equations instead of a convex quadratic programming (QP) problem for classical SVMs (Huang & Shyu, 2010).

There are two different modeling options for constructing forecasting models in time series prediction. One is global which constructs a global model which is independent from user queries. The other is local which, instead, constructs a local model for each different query from the user (Martinez-Rego, Fontenla-Romero, & Alonso-Betanzos, 2011). In general, the global modeling option is simpler but is more vulnerable to outliers. On the other hand, the local modeling option is more complex but can be more adaptive. Local modeling is usually characterized by using a small number of neighboring subsequences in proximity of the user query. An issue related to local modeling is to select the lags involved in the model building process and the forecasting of the target (Bag, 2007; Kaneko, Matsuzaki, Ito, Oogai, & Uchida, 2010). Some lags selection approaches apply t-statistics of estimated coefficients or F-statistics of groups of coefficients to measure statistical significance (Hastie, Tibshirani, & Friedman, 2008). Mutual information was adopted in lags selection for electric load forecasting (Božić, Stojanović, Stajić, & Floranović, 2013).

In this paper, we propose a local modeling strategy and investigate the effectiveness of incorporating local modeling with three popular machine learning based forecasting methods, Neural Network (NN) (Hagan & Menhaj, 1994), Adaptive Neuro-Fuzzy Inference System (ANFIS) (Jang et al., 1997), and Least Squares Support Vector Machine (LS-SVM) (Suykens et al., 2002), for time series prediction. Given a series of historical data, several steps are involved. Firstly, a local context of the user query is found by locating the query's k nearest neighbors based on a hybrid distance measure. Secondly, an appropriate number of lags are selected by applying mutual information to measure their relevance to the target to be predicted. A desired set of training patterns is then extracted from the given historical data. Finally, the training patterns are fed to NN, ANFIS, and LS-SVM, respectively, and forecasting models are derived. A number of experiments are conducted and the results show that local modeling can enhance the estimation performance of a forecasting method for time series prediction.

The rest of this paper is organized as follows. Section 2 gives the definition of the time series problem. NN, ANFIS, and LS-SVM are briefly introduced in Section 3. Our local modeling approach is described in Section 4. An illustrating example is given in Section 5. Experimental results from running several real-world data sets are presented in Section 6. Some issues are discussed in Section 7. Finally, concluding remarks are given in Section 8.

2. Time series prediction

Given a sequence of real-valued observations (Wei, 2005):

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

taken at equally spaced time points $t_0, t_0 + \triangle t, t_0 + 2\triangle t, \ldots$ for some process P, where Y_i denotes the value of the output variable (or dependent variable) observed at the time point $t_0 + i\triangle t$ and $\mathbf{X}_i$ denotes the values of m additional variables (or independent variables), $m \geqslant 0$, observed at the time point $t_0 + i\triangle t$, 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 $s \geqslant 1$ 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. If $s = 1$, it predicts just one time step ahead into the future and is called one-step prediction. For $s > 1$, it is usually called multi-step prediction. Also, if $m = 0$, it is univariate prediction; otherwise, it is multivariate prediction. For convenience, the subsequence

$$\mathbf{Q} = [\mathbf{X}_{t-q} \quad Y_{t-q} \quad \ldots \quad \mathbf{X}_{t-1} \quad Y_{t-1} \quad \mathbf{X}_t \quad Y_t] \tag{3}$$

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

3. Machine learning based forecasting methods

The three forecasting methods NN, ANFIS, and LS-SVM we are going to work with is briefly introduced. Details can be found in (Hagan & Menhaj, 1994; Jang et al., 1997; Suykens et al., 2002). These are classic machine learning algorithms that we would like to investigate the effectiveness of our proposed local modeling strategy. Any other pattern-based machine learning algorithm can also be employed. Let the system we would like to model have n input variables $x_1, x_2, \ldots, x_n$, denoted as $\mathbf{x} = [x_1 \quad x_2 \quad \ldots \quad x_n]$, and one output variable y. Assume that a set $\mathcal{T}$ of ℓ training patterns $(\mathbf{p}_1, q_1), (\mathbf{p}_2, q_2), \ldots, (\mathbf{p}_\ell, q_\ell)$ is given, where $\mathbf{p}_v = [p_{v,1} \quad p_{v,2} \quad \cdots \quad p_{v,n}]$ and q_v denote the n input values and the desired output value, respectively, associated with the vth pattern, $1 \leqslant v \leqslant \ell$.

3.1. Neural network (NN)

The architecture of the neural network we use is shown in Fig. 1. It consists of three layers: input layer, hidden layer, and output layer. There are n nodes in the input layer, h nodes in the hidden layer, and one node in the output layer. The weight between node i of the input layer and node j of the hidden layer is denoted by $w_{j,i}^{(1)}$, $1 \leqslant j \leqslant h$, $1 \leqslant i \leqslant n$. The weight between the output node and node j of the hidden layer is denoted by $w_j^{(2)}$, $1 \leqslant j \leqslant h$. The bias of node j in the hidden layer is denoted by $b_j^{(1)}$, $1 \leqslant j \leqslant h$, and the bias of the output node is denoted by $b^{(2)}$. For n given inputs $p_1, p_2, \ldots, p_n$, the network provides the following output:

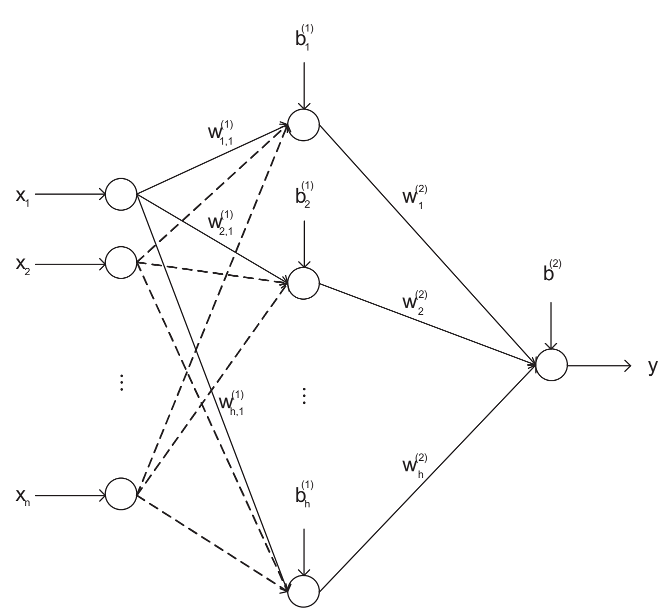

Fig. 1. Architecture of neural network.