Engineering Applications of Artificial Intelligence 44 (2015) 101–110

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Kalman filter-based method for Online Sequential Extreme Learning Machine for regression problems

Jarley Palmeira Nobrega *, Adriano L.I. Oliveira

[a] Centro de Informática, Universidade Federal de Pernambuco (UFPE), Cidade Universitária, 50740-560 Recife, PE, Brazil

ARTICLE INFO

Article history:
Received 12 December 2014
Received in revised form
13 April 2015
Accepted 19 May 2015
Available online 6 June 2015

Keywords:
Online sequential learning
Extreme learning machine
Online Sequential Extreme Learning Machine
Kalman filter regression
Multicollinearity

ABSTRACT

In this paper, a new sequential learning algorithm is constructed by combining the Online Sequential Extreme Learning Machine (OS-ELM) and Kalman filter regression. The Kalman Online Sequential Extreme Learning Machine (KOSELM) handles the problem of multicollinearity of the OS-ELM, which can generate poor predictions and unstable models. The KOSELM learns the training data one-by-one or chunk-by-chunk by adjusting the variance of the output weights through the Kalman filter. The performance of the proposed algorithm has been validated on benchmark regression datasets, and the results show that KOSELM can achieve a higher learning accuracy than OS-ELM and its related extensions. A statistical validation for the differences of the accuracy for all algorithms is performed, and the results confirm that KOSELM has better stability than ReOS-ELM, TOSELM and LS-IELM.

1. Introduction

There has been growing interest in the development of machine learning methods for intelligently recognizing complex patterns even when all training data is not available. Over the past few decades, a large number of batch learning algorithms have been discussed and studied thoroughly, including the classical learning methods based on Backpropagation (BP) (Ruhmelhart et al., 1986) and Support Vector Machine (SVM) (Drucker et al., 1997). These methods can approximate complex nonlinear functions from the input samples without knowing the internal structure of the training data. However, it is known that the classical methods require a long time for tuning the parameters, and the learning process can be inefficient when the amount of training data is very large. Retraining the model using all of the data with the batch learning algorithms is a time consuming task which is unacceptable for many real-world applications that require fast learning speed and high accuracy. To overcome the weakness of traditional neural networks, Extreme Learning Machine (ELM) was introduced by Huang et al. (2006) to this end. One advantage of ELM over traditional neural networks is that it is not necessary to adjust parameters iteratively. Thus, ELM has a much faster training process with better performance in some cases when compared with traditional neural networks (Huang et al., 2015). This has been a motivation for the application of ELM to many practical problems, for instance, image classification (Luo and Zhang, 2014), multi-step-ahead time series prediction (Grigorievskiy et al., 2014) and fault section identification in transmission lines (Malathi et al., 2011).

In order to decrease the training time for batch algorithms, new classes of sequential learning methods were introduced (Platt, 1991; Kadirkamanathan and Niranjan, 1993; Yingwei et al., 1998; Huang et al., 2005). In some applications, these methods are preferred over batch algorithms as they do not need to retrain the model whenever a new training sample is received. In Liang et al. (2006), a sequential version of ELM has been developed to deal with the problem where training samples are received one-by-one or chunk-by-chunk. The Online Sequential Extreme Learning Machine (OS-ELM) is based on a single layer feedforward neural network (SLFN), in which the input weights and hidden layer biases are chosen randomly and the output weights are determined by the pseudo-inverse of the hidden layer matrix. The output weight estimation of ELM can be expressed as an ordinary least squares (OLS) solution. However, the OLS estimator can generate poor predictions in the presence of multicollinearity due to their large variance which decreases the model stability (Foucart, 1999).

To overcome this problem, some related work should be mentioned. In Huynh and Won (2011), the ReOS-ELM is

* Corresponding author.
E-mail addresses: jpn@cin.ufpe.br (J.P. Nobrega), alio@cin.ufpe.br (A.L.I. Oliveira).

http://dx.doi.org/10.1016/j.engappai.2015.05.010
0952-1976/

proposed with the use of a bi-objective optimization to obtain the small norm of the output weights. In this work, the problem of multicollinearity is controlled by applying a regularization parameter to update the weights. However, this method has some drawbacks. First, there is an additional parameter to adjust, the regularization constant, which makes the output of the algorithm dependent on its correct estimation. In addition, the optimization of the output weights is also dependent on the size of the sequential chunk data. In Gu et al. (2014), the stability and the convergence of the model are improved by the use of an adaptive scheme to update the output weights. This method, namely TOSELM, calculates the mean and variance of the sequential training data and adjusts the output weights by using an exponential function of the distribution of the data. In spite of the results reported in this work there has been a decrease indicated in the variance. This approach increases the computational complexity of the sequential learning algorithm since an additional iterative method is applied during the regular cycle of learning. Moreover, the stopping condition for the weight adjustment requires an additional parameter for the tolerance of the convergence, and this parameter must be tuned before the sequential training phase. Two similar approaches have been recently presented in Ye et al. (2013) and Guo et al. (2014). In the first work, the input and output weights are changing over time for both training and testing phases. The proposed OS-ELM-TV does not directly handled the problem of multicollinearity, but it reduces the variance of the estimations by using an output basis function to calculate a linear combination of the weights. This work present a fast training step and a stable model, despite the fact that the limitation of this approach is the correct choice of the output basis function, such as Fourier or Legendre functions, which demands an additional optimization step. The second approach, namely LS-IELM, has also an additional parameter to adjust, which is defined as a cost constant. This approach is similar to the ReOS-ELM, since the cost constant can be considered as a regularization parameter for the least squares solution of OS-ELM. In the LS-IELM, the problem of multicollinearity is not controlled directly since it updates the output weights by using the same recursive procedure of OS-ELM. A limitation of this method is the requirement of the optimization of the cost constant using a portion of the training data.

The limitations of the cited related work indicate that the problem of multicollinearity should be addressed without the optimization of additional parameters since they increase the computational complexity of the algorithm. It is known in the literature that the adjustment of the variance for an OLS based model can be modeled as a state space problem (Watson, 1983). In this context, a Kalman filter regression model could be applied to adjust the output weights of the OS-ELM with no additional parameters to be optimized. Since the Kalman filter applies a recursive least squares (RLS) solution in a similar way as the OS-ELM, an adjustment coefficient could be updated for each cycle of the sequential learning step. At the end of this process, the output weights are adjusted to avoid the effects of multicollinearity in the variance of the estimations.

The main motivation behind this paper is to investigate the limitations of OS-ELM and the related extensions with respect to the problem of multicollinearity. Based on a state space model, this paper proposes an improvement of OS-ELM, namely Kalman Online Sequential Extreme Learning Machine (KOSELM), in which the Kalman filter regression is applied to update the estimates of the output weights by adjusting its variance. In KOSELM, we use the capacity of the Kalman filter to handle the multicollinearity and adjust the variance of the estimated state. We perform a filtering procedure for the estimated output weight matrix during the update cycle. Next, the estimation is adjusted through the regression coefficient provided by the filtering procedure.

The main contributions of this paper are enumerated as follows:

1. The proposed KOSELM algorithm to the sequential learning problem in both one-by-one and chunk-by-chunk modes.
2. A weight update scheme using the Kalman filter regression for variance adjustment of the output weights.
3. An empirical comparison between the KOSELM and the related work for benchmark regression problems. Additionally, a statistical validation was performed for the differences of the accuracy of the studied algorithms.

This paper is organized as follows. Section 2 gives a brief overview of ELM, OS-ELM and Kalman filter. In Section 3, the proposed method for variance adjustment is presented. In Section 4, the description of the sample data, the experimental settings and the statistical validation of KOSELM are presented. Section 5 concludes the paper.

2. Fundamentals

In this section, the concepts of the batch version of ELM, the OS-ELM and Kalman filter regression are briefly reviewed.

2.1. Extreme Learning Machine

Unlike the traditional learning algorithms for neural networks, the main characteristic of ELM is learning without iterative training as proposed by Huang et al. (2006). This makes the learning process faster when compared to the traditional algorithms for the training of neural networks. Let the training set be $\{(\mathbf{x}_i, \mathbf{t}_i)\}, \mathbf{x}_i \in \mathbb{R}^n, \mathbf{t}_i \in \mathbb{R}^m, i = 1, \ldots, N$, where $\mathbf{x}_i$ is an $n \times 1$ input vector and $\mathbf{t}_i$ is a $m \times 1$ target vector. The training process is briefly described as follows.

Step 1: Randomly assign values to the inputs weights and hidden neuron biases, a_i and b_i.

Step 2: The output weights are analytically determined through the generalized inverse operation of the hidden layer matrices (Huang et al., 2006), according to the following equation:

$$\sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) = \mathbf{t}_j, \quad j = 1, \ldots, N \tag{1}$$

where a_i are the input weights, b_i are the hidden layer biases, $\boldsymbol{\beta}_i$ is the output weight that connects the ith hidden node and the output node, and G is the activation function. L is the number of hidden neurons. N is the number of distinct input or output data. This is equivalent to $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$, where

$$\mathbf{H} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_L) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L}, \tag{2}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \tag{3}$$